

# PROPERTY TESTING, PCPS AND CSPA

BY AMEY BHANGALE

A dissertation submitted to the  
Graduate School—New Brunswick  
Rutgers, The State University of New Jersey  
in partial fulfillment of the requirements  
for the degree of  
Doctor of Philosophy  
Graduate Program in Computer Science

Written under the direction of  
Swastik Kopparty  
and approved by

---

---

---

---

New Brunswick, New Jersey

May, 2017

## ABSTRACT OF THE DISSERTATION

### Property Testing, PCPs and CSPs

by Amey Bhangale

Dissertation Director: Swastik Kopparty

Many optimization problems can be modeled as constraint satisfaction problems (CSPs). Hence understanding the complexity of solving or approximating CSPs is a fundamental problem in computer science. The famous PCP (probabilistically checkable proof) Theorem states that certain CSPs are hard to approximate within a constant factor. In the language of proof verification, the theorem implies that a proof of a mathematical statement can be written in a specific format such that it allows an sublinear time verification of the proof. Thus, property testing procedures are central to PCPs, and in fact the proof of the PCP theorem involves many interesting property testing algorithms.

Some of the highlights of this dissertation include the following results:

1. *Low degree testing* is one of the important components in the proof of the PCP theorem and *Dictatorship testing* is central in proving hardness of approximation results. This thesis presents a Cube vs Cube low degree test which has significantly better parameters than the previously known tests. We also improve on the soundness of  $k$ -bit dictatorship test with perfect completeness.
2. In the area of inapproximability, this thesis offers a complete characterization of approximating the *covering number* of a CSP, assuming a covering variant of

UNIQUE GAMES CONJECTURE. We also prove tight inapproximability results for BI-COVERING problem.

3. This thesis studies CSPs from a multi-objective point of view. We give almost optimal approximation algorithms for multi-objective MAX-CSP (simultaneous CSPs), and also prove inapproximability results.

## Acknowledgements

This is the most difficult part to write as words of gratitude will not be enough to express my feelings. Nevertheless, I will make an effort to thank people who helped me achieve this.

First and foremost, I would like to sincerely thank my advisor Swastik Kopparty. It is always fun to work with him. In all these years, every meeting with Swastik was inspirational and a never ending learning experience for me. I am thankful to him for his patience with me. I could not have imagined having a better advisor for my Ph.D.

I would like to thank my mentor and guide Rajiv Gandhi for introducing me to the topic of my research, inspiring me throughout my career and helping me during my Ph.D journey. Things would have been completely different had I not met Rajiv Sir.

I have benefited from multiple research visits throughout my Ph.D. I would like to thank Irit Dinur, Prahladh Harsha, Subhash Khot, Rishi Saket to whom I visited during the summers. I would like to thank all the participants of the theory reading group at Rutgers and a big thanks to Mike Saks for organizing it.

I have been very fortunate to have collaborated with immensely talented and inspiring individuals throughout my journey and would like to thank each one of them: Irit Dinur, Rajiv Gandhi, Mohammad Hajiaghahi, Prahladh Harsha, Rohit Khandekar, Subhash Khot, Swastik Kopparty, Guy Kortsarz, Inbal Livni, Sushant Sachdeva, Devanathan Thiruvengkatachari, Girish Varma.

I would like to thank Abdul, Aditya, Abhishek, Ben, Mrinal for all the wonderful times spent in New Brunswick. I would like to thank all my friends from India for their support and encouragement: Ajey, Ajinkya, Chetna, Dhanashree, Mandar, Mayur, Pravin, Pooja, Sangeetha, Sanket, Sonali & Vinay.

Finally and most importantly, I would like to thank my family for their continuous support and love. My heartfelt thank you to my Mom and Dad for helping in whatever way they could during this challenging period.

## Dedication

To my Parents

# Table of Contents

<b>Abstract</b> . . . . .	ii
<b>Acknowledgements</b> . . . . .	iv
<b>Dedication</b> . . . . .	v
<b>1. Introduction</b> . . . . .	1
1.1. Constraint Satisfaction Problems . . . . .	2
1.2. Probabilistically Checkable Proofs . . . . .	3
1.3. Unique Games Conjecture . . . . .	7
1.4. Brief Summary of Results . . . . .	8
1.5. Organization . . . . .	11
<b>2. Preliminaries</b> . . . . .	13
2.1. Analysis of Boolean Functions . . . . .	13
2.2. Correlated Spaces . . . . .	15
2.3. Hypercontractivity . . . . .	17
2.4. Invariance Principle . . . . .	18
2.5. Expanders & Extractors . . . . .	22
<b>3. Covering CSP</b> . . . . .	23
3.1. Introduction . . . . .	23
3.2. Preliminaries . . . . .	30
3.3. UG Hardness of Covering . . . . .	32
3.4. NP-Hardness of Covering . . . . .	37
3.5. Improvement to covering hardness of 4-LIN . . . . .	46

<b>4. Parallel Repetition</b>	50
4.1. Introduction	50
4.2. Preliminaries	57
4.3. Sub-games on large rectangles	60
4.4. Fortifiers are necessary	61
4.5. Robustness from fortifiers	63
4.6. An explicit extractor that does not provide robustness	65
4.7. Lower bounds on degree of fortifiers	67
<b>5. Inapproximability of Bi-Covering</b>	69
5.1. Introduction	69
5.2. Results	70
5.3. Preliminaries	72
5.4. Dictatorship Test	74
5.5. $(2 - \varepsilon)$ - inapproximability	77
<b>6. Low Degree Test</b>	82
6.1. Introduction	82
6.2. Preliminaries and Notations	88
6.3. Proof of the Main Theorem	91
6.4. Comparing between different tests and their agreement parameter	104
6.5. Spectral properties of Certain Inclusion Graphs	110
6.6. Spectral Expansion Properties Proofs	118
6.7. Rubinfeld-Sudan Characterization	123
<b>7. <math>k</math>-bit Dictatorship Test</b>	127
7.1. Introduction	127
7.2. Proof Overview	132
7.3. Invariance Principle	135
7.4. Query efficient Dictatorship Test	136

7.5. Analysis of the Dictatorship Test . . . . .	140
<b>8. Simultaneous Optimization . . . . .</b>	<b>153</b>
8.1. Introduction . . . . .	153
8.2. Notation for the Main Algorithms . . . . .	167
8.3. Simultaneous MAX-2-AND . . . . .	168
8.4. Simultaneous MAX- $w$ -CSP $_q$ . . . . .	184
8.5. Simultaneous MAX- $w$ -SAT . . . . .	202
8.6. Hardness Results for Large $k$ . . . . .	215
8.7. Algorithm for Unweighted MAX-CUT . . . . .	223
8.8. SDP for Simultaneous Instances . . . . .	227
<b>9. Simultaneous Max-Cut . . . . .</b>	<b>233</b>
9.1. Introduction . . . . .	233
9.2. Overview of the algorithm . . . . .	235
9.3. Preliminaries . . . . .	239
9.4. Algorithm for simultaneous weighted MAX-CUT . . . . .	240
9.5. Deferred Proofs . . . . .	263
<b>References . . . . .</b>	<b>269</b>

# Chapter 1

## Introduction

The theory of NP-completeness deals with solving a problem exactly in the worst case. Many optimization problems are NP-hard which means these problems can be solved efficiently unless  $P = NP$ . In order to get around this barrier, one can hope that the instances we actually need to solve in practice are not the worst case instances or one can hope to design a sub-optimal algorithm. The later means even if we cannot find the exact solution efficiently, we can hope to find an approximate solution efficiently. In this dissertation, we are interested in designing approximation algorithms as well as understanding the limitations of such approximation algorithms. It turns out that different optimization problems behave differently with respect to admitting approximation algorithms.

For a maximization problem, a  $c$  approximation algorithm has a guarantee that for every instance  $I$  the algorithm guarantees to find a solution of cost at least  $1/c$  times the optimal value. Some problems admit PTAS which means that there is an efficient  $(1 + \epsilon)$ -approximation algorithm for every  $\epsilon > 0$  where as for other problems we only know approximation algorithm for some constant  $c > 1$ . In the extreme case, there are problems where the approximation guarantee degrades with the size of an instance. Given such erratic behavior of a wide class of optimization problems, it is natural to study this behavior from a systematic theory point of view.

In order to prove inapproximability of certain optimization problem  $\mathcal{P}$ , one way is to reduce a canonical NP-complete problem like SAT to a gap version of the maximization problem. A  $(c, s)$ - gap version of the problem  $\mathcal{P}$  is a promise problem where the task is to distinguish between the cases when  $OPT(I) \geq c$  vs  $OPT(I) \leq s$ , where  $c \geq s$ . Suppose

there is a reduction from SAT to  $(c, s)$  gap version of problem  $\mathcal{P}$  then this implies that  $\mathcal{P}$  is hard to approximate within a factor of  $c/s$  unless SAT can be efficiently solvable.

## 1.1 Constraint Satisfaction Problems

Constraint satisfaction Problems (CSP) are the most basic class of problems in NP. It consists of a set of variables  $X = \{x_1, x_2, \dots, x_n\}$  taking values from some domain  $\Sigma$  and a set of constraints  $\mathcal{C} = \{C_1, C_2, \dots, C_m\}$  on the set of variables. The constraints  $C_i$ 's are typically local in the sense that it only involves small set of variables, mostly constant independent of  $n$ . The task is to decide if there exists an assignment  $f : X \rightarrow \Sigma$  which satisfies all the constraints from  $\mathcal{C}$ . MAX-CSP is an optimization variant of CSP where one wants to find an assignment  $f$  which satisfies as many constraint as possible.

We give a couple of examples of CSP which we study.

- MAX-3-SAT: In MAX-3-SAT the constraints are of the type  $l_i \vee l_j \vee l_k$  where  $l_i, l_j, l_k$  are the literals.
- MAX-CUT: Given an undirected graph  $G(V, E)$  find a partition  $(U, \bar{U})$  of  $V$  such that it maximizes the fraction of edges with one endpoint in  $U$  and the other in  $\bar{U}$ . Although modeled as a graph optimization problem, it is easy to see MAX-CUT falls inside a class of MAX-CSP where the constraints are of the type  $x_i \oplus x_j$ .

The theory of approximation algorithms for constraint satisfaction problems is a very central and well developed part of modern theoretical computer science. Its study has involved fundamental theorems, ideas, and problems such as the PCP theorem, linear and semidefinite programming, randomized rounding, the Unique Games Conjecture, and deep connections between them [AS98, ALM<sup>+</sup>98, GW95, Kho02a, Rag08, RS09].

We study CSPs from multi-objective point of view which is a significant part of this dissertation.

## 1.2 Probabilistically Checkable Proofs

The NP-completeness theory of [Coo71, Lev73, Kar72] gives a way of reducing a generic language  $L$  in NP to 3-SAT with two guarantees: If the given instance  $x$  is in the language then the reduced 3-SAT instance  $\phi_x$  is satisfiable and if it is not in the language then  $\phi_x$  is not satisfiable. Thus, if  $P \neq NP$ , there is no efficient algorithm to decide if a given 3-SAT instance is satisfiable or not. This is useful since we now know one natural problem which is hard to solve exactly and can be used to show hardness of other optimization problems.

Although, this helps in settling the complexity of exactly solving certain optimization problems, it turns out that in order to get a better inapproximability result we need a robust characterization of the class NP which is provided by Probabilistically Checkable Proofs [AS98, ALM<sup>+</sup>98]. Using the PCP Theorem, the above reduction can be made stronger in the following sense: when  $x \in L$  then  $\phi_x$  is satisfiable whereas if  $x \notin L$  then no assignment satisfies more than  $1 - \varepsilon$  fraction of the clauses in  $\phi_x$  for some  $\varepsilon > 0$ . This immediately rules out a PTAS for MAX-3-SAT.

The modern study of inapproximability soon followed after the discovery of the PCP theorem. In fact, Fiege *et al.* [FGL<sup>+</sup>96] showed that a robust characterization of NP can be used to show inapproximability of Independent Set problem before the PCP theorem was proved. After the PCP Theorem was proved, many inapproximability results were proved for different optimization problems.

One of the important component in proving the PCP theorem is so called Low Degree Test. This dissertation offers a cube vs cube low degree test with arguably much simpler proof than the known proofs of low degree tests.

### 1.2.1 Label Cover

One of the important developments in proving the inapproximability results is formulating a problem called Label Cover, also known as 2-Prover-1-Round Game, and the use of Long code test as a gadget in the reduction. An instance  $G = (U, V, E, [L], [R], \{\pi_e\}_{e \in E})$  of the LABEL-COVER constraint satisfaction problem consists of a bi-regular bipartite

graph  $(U, V, E)$ , two sets of alphabets  $[L]$  and  $[R]$  and a projection map  $\pi_e : [R] \rightarrow [L]$  for every edge  $e \in E$ . Given a labeling  $\ell : U \rightarrow [L], \ell : V \rightarrow [R]$ , an edge  $e = (u, v)$  is said to be satisfied by  $\ell$  if  $\pi_e(\ell(v)) = \ell(u)$ .

The PCP Theorem is equivalent to the following inapproximability of LABEL-COVER.

**Theorem 1.2.1.** *There exists a constant  $c < 1$  such that given an instance of LABEL-COVER, it is NP-hard to distinguish between two cases*

- *There exists an assignment satisfying all the edges.*
- *No assignment satisfies more than  $c$  fraction of constraints.*

We can think of a LABEL-COVER instance  $G = (U, V, E, L, R, \{\pi_e\}_{e \in E})$  as a 2-Prover-1-Round Game where prover 1 assigns labels from  $[L]$  to  $U$  and prover 2 assigns labels from  $[R]$  to  $V$ . The verifier picks an edge  $e(u, v) \in E$  uniformly at random and asks prover 1 a label for vertex  $u$  and prover 2 a label for vertex  $v$ . The verifier accepts if the labels returned by the two provers satisfy the constraint on  $e$ . The strategies of two provers corresponds to the labelings  $\ell_1 : U \rightarrow [L], \ell_2 : V \rightarrow [R]$ . The value of the game is the maximum over all the provers strategies the probability that the verifier accepts. It is easy to see that if  $G$  is  $\delta$  satisfiable then the value of the game is at least  $\delta$  and vice versa. Thus, these two views are equivalent.

As a consequence of the PCP Theorem, one can characterize the class NP in the language of proof checking. Recall that NP is a class of languages for which if a given input is in the language then there is a short proof such that given the proof it can be verified in polynomial time. Apriori, the verifier might need to read the entire proof to make its decision. The proof checking viewpoint of the PCP Theorem states that there is a way of writing a short proof such that it can be verified by looking at only constant number of bits with an expense that the verifier makes an error with tiny probability. This way of interpreting the PCP Theorem has proven many applications.

### 1.2.2 Parallel Repetition Theorem

It turns out that approximating the value of a LABEL-COVER instance is much more harder than implied by the PCP theorem. This follows from the Parallel Repetition

theorem. Consider any 2-Prover 1-Round game. Suppose we repeat the game sequentially  $k$  times and let the provers win if they win every single round. It is easy to see that if there exists a strategy for the original game that makes verifier accept with probability 1 then the prover can make the verifier accept with probability 1 in the repeated game. What happens to the value of the repeated game if the value of original game is  $1 - \delta$ . In this case also it is trivial to see the the value of repeated game is  $(1 - \delta)^k$ : this follows because each round was independent of the previous rounds and the probability of winning a single round is at most  $(1 - \delta)$ . But the final repeated game is no longer a one-round game and because of the connection between the PCP Theorem and 2-Prover 1-Round game this sequentially repeated game is not useful.

How can we make the game one-round? This is where parallel repetition comes into picture. Suppose instead of asking questions sequentially, the verifier samples  $k$  questions from the original game and asks the two provers to reply to the  $k$  questions simultaneously. By doing this we are preserving the one-round nature of the game. In this case also, it is easy to see that if the value of original game is 1 then the value of parallel repeated game is also 1. It was not clear how fast the value of repeated game decays if the value of original game is less than 1. This is because the provers can correlated there strategies based on the sequence of  $k$  questions the receive. This question was answered by Raz [Raz98] and showed that the value does go down exponentially. This along with the PCP Theorem gives an improved hardness of approximating the value of label cover i.e for every constant  $\varepsilon > 0$ , even if there exists an assignment which satisfies all the edges, it is NP-hard to find an assignment which satisfies more than  $\varepsilon$  fraction of the edges.

**Theorem 1.2.2** (PCP Theorem + Raz's Parallel Repetition Theorem). *For every  $\varepsilon > 0$  there exists  $R, L \in \mathbb{N}^+$  such that given an instance  $G = (U, V, E, [L], [R], \{\pi_e\}_{e \in E})$  of LABEL-COVER, it is NP-hard to distinguish between two cases*

- *There exists an assignment satisfying all the edges.*
- *No assignment satisfies more than  $\varepsilon$  fraction of constraints.*

Another way of stating this theorem is  $(1, \varepsilon)$ -gap version of LABEL-COVER is NP-hard for every  $\varepsilon > 0$ .

### 1.2.3 Long Code Test

This inapproximability of LABEL-COVER is starting point of many inapproximability results that followed. Suppose one wants to reduce the  $(1, \varepsilon)$ -gap version of LABEL-COVER to a gap version of some maximization problem. The reduction involves replacing each vertex  $u \in U$  ( $v \in V$ ) of the LABEL-COVER instance by a boolean hypercube of dimension  $L$  ( $R$ ). The labels to the vertices of the LABEL-COVER instance is identified by the dimension of the boolean hypercube. Then one can *encode* a label of a vertex by defining a boolean function on the hypercube associated with the vertex. A very wasteful way of encoding a label is by defining a function  $f_u : \{0, 1\}^L \rightarrow \{0, 1\}$  as  $f_u(x_1, x_2, \dots, x_L) = x_i$  where  $i$  corresponds to the label of vertex  $u$ . Such an encoding is called a Long Code encoding and the class of such functions are called *dictator* functions. The reduction then involves using the vertices of boolean hypercubes as variables to the maximization (or minimization) problem.

If such a reduction to work, one needs a testing procedure to check if the encoded function is a dictator function or far from a dictator function. More specifically, if there is a testing procedure that accepts any dictator function with probability at least  $c$  and accepts any function far from a dictator function with probability at most  $s$ , then such procedures can be used to show the hardness of  $(c, s)$  gap version of the problem. Of course, the testing procedure and the number of queries decide the target hard problem. Dictatorship tests/long-code tests were introduced into hardness of approximation by the work of Bellare, Goldreich, and Sudan [BGS98].

Håstad in his influential work started the use of Fourier Analysis in analyzing such reductions starting from LABEL-COVER. In [Hås01], Håstad settled the inapproximability of many constraint satisfaction problems. More specifically, he showed that approximating MAX-3-SAT better than a trivial  $7/8$  approximation is NP-hard even if the instance is satisfiable. Note that getting  $7/8$  approximation for MAX-3-SAT is trivial because if one picks a random assignment, then the probability that a given

clause is satisfied is exactly  $7/8$ . Thus, in expectation a random assignment satisfies  $7/8$  fraction of clauses. He also showed optimal inapproximability of many other CSPs including MAX-3-LIN.

### 1.3 Unique Games Conjecture

As discussed above, the inapproximability of LABEL-COVER is used to prove many inapproximability results. Although, there was a lot of success in proving hardness results, the complexity of approximating many NP-hard problems remained elusive. For many optimization problems, including Vertex Cover, MAX-CUT etc. there was a large gap between the known approximation algorithms and inapproximability results.

Khot's insight [Kho02a] was to modify the type of constraints in the LABEL-COVER problem. The  $(1, \varepsilon)$ -gap version of LABEL-COVER implied by the PCP Theorem and Parallel Repetition Theorem has “many-to-one” constraints associated with every edge. This many-to-one requirement of the constraints created a problem in reducing the LABEL-COVER to a target problem where the constraints are between pair of variables as in the case of MAX-CUT, Vertex Cover etc. as well as for other optimization problems. Khot realized that if one makes the constraints of LABEL-COVER one-to-one by still preserving the gap then this is useful in proving many inapproximability results which were not known otherwise. LABEL-COVER instance with one-to-one constraints is called *Unique Game* because of the nature of the constraints - for any edge  $(u, v)$  of a label for vertex  $u$  is fixed then there is a *unique* label for  $v$  which satisfies the constraint. This one-to-one constraint has a drawback in the if the UNIQUE-GAMES instance is satisfiable then there is a simple polynomial time algorithm which can find the satisfying assignment. Thus, one cannot expect a hardness of  $(1, \varepsilon)$ -gap of UNIQUE-GAMES for any  $\varepsilon \in [0, 1]$ . Nevertheless, it is not clear how to find an optimal assignment if the instance is almost satisfiable. Khot conjectured that  $(1 - \varepsilon, \varepsilon)$ -gap problem of UNIQUE-GAMES is NP-hard.

**Conjecture 1.3.1** ([Kho02a]). (UNIQUE GAMES CONJECTURE) *For every  $\varepsilon > 0$  there*

exists  $L \in \mathbb{N}^+$  such that given an instance  $G = (U, V, E, [L], [L], \{\pi_e\}_{e \in E})$  of UNIQUE-GAMES, it is NP-hard to distinguish between two cases

- There exists an assignment satisfying at least  $(1 - \varepsilon)$  fraction of the constraints.
- No assignment satisfies more than  $\varepsilon$  fraction of the constraints.

In this dissertation, we explored implications of UNIQUE GAMES CONJECTURE or certain variant of it to give tight results for covering CSPs, Bi-covering and Max-Bi-Clique. These are explained in [Section 1.4](#).

## 1.4 Brief Summary of Results

In this section, we present the brief summer of contribution of this dissertation. It is mainly divided into *three* parts : PCPs, CSPs and Property testing.

### Part I : PCPs

**Parallel repetition theorem:** The parallel repetition theorem was first proven by Raz [\[Raz98\]](#). Many simpler proofs of the theorem came later. Moshkovitz gave a much simpler proof of the parallel repetition theorem restricted to certain instances which is enough for its application to hardness of approximation. Moshkovitz showed that the theorem follows quite easily if the original instance satisfies certain property, called *fortification*. Building upon the work of Moshkovitz [\[Mos14\]](#), we give a very simple combinatorial proof of the parallel repetition theorem in [\[BSVV15\]](#). With my co-authors in [\[BSVV15\]](#), we gave a construction of how to convert any instance into fortified one which is much easier to analyze. We crucially use the expansion of a graph to argue about the required fortification guarantee.

**Bi-covering:** Given an undirected connected graph  $G(V, E)$  Bi-covering is a problem of finding to sets  $A, B \subseteq V$  such that  $A \cup B = V$  and there are no edges between  $A \setminus B$  and  $B \setminus A$ . The objective is to minimize  $\max\{|A|, |B|\}$ . This problem has a trivial 2-approximation as  $A = B = V$  is always a feasible solution. In [\[BGH<sup>+</sup>16\]](#), we show that assuming the UGC where the constraint graph has some mild expansion,

2-approximation is the best possible in polynomial time. An important implication of this hardness result is that this implies inapproximability of a well-known problem called Max-Bi-Clique which is problem of finding largest  $k \times k$  Bi-Clique in an  $n \times n$  bipartite graph. The above hardness result implies that it is NP-hard to approximate Max-Bi-Clique within  $n^\delta$  for some  $\delta > 0$  under the same assumption.

## Part II : CSPs

**Simultaneous CSPs:** Multiobjective optimization is an area of optimizing over more than one objective function where all the objective functions share the same solution space. It has been used in many areas including engineering, data mining, machine learning etc. We initiated a study of the most fundamental multi-objective optimization where each objective function is a certain Max-CSP instance. More formally, we study the following problem : Given  $k$  instances of Max-CSP over variables  $x_1, x_2, \dots, x_n$ , find an assignment  $\mathbf{x}$  which maximizes  $\min_{i \in [k]} \text{val}(i, \mathbf{x})$  where  $\text{val}(i, \mathbf{x})$  is the fraction of the satisfied constraints in the  $i^{\text{th}}$  instance.

In a joint work with Kopparty and Sachdeva [BKS15], we explored this class of multiobjective optimization, which we call *simultaneous CSPs*. We gave a constant factor approximation algorithm for every CSP as long as the number of instances is bounded. We also show that if the number of instances is more than  $\text{poly}(\log n)$  then assuming ETH there is no constant factor approximation to simultaneous CSPs. In a follow-up work with Khot, Kopparty, Sachdeva and Thiruvengatachari [BKK<sup>+</sup>16], we improve the approximation ratio of simultaneous MAX-CUT problem from  $1/2$  to very close to  $\alpha_{GW}$  - an optimal approximation ratio for MAX-CUT problem assuming the UNIQUE GAMES CONJECTURE. This improved algorithm uses Sum-of-Squares hierarchies, a systematic way of tightening semi-definite program relaxation by adding a sequence of inequalities.

**Covering CSPs:** The *covering number* of an instance of a CSP is the minimum number of assignments needed such that for every constraint there is at least one assignment in the collection which satisfies the constraint. This problem is closely related to the

chromatic number of hypergraph. In [BHV15], we study the computational complexity of approximating the covering number. It was first studied by Dinur and Kol [DK13] where they showed that assuming a covering variant of the *Unique Games Conjecture* (UGC), for a small class of predicates, it is NP-hard to approximate the covering number within any constant. We completely settled the computational complexity of approximating the covering number of a given predicate. More specifically, we gave a simple and complete characterization of when the covering number is hard to approximate within any large constant under the same assumption. We also give partial characterization based on  $P \neq NP$  for a class of predicates by proving an invariance principle tailored to our setting.

### Part III : Property Testing

**Low degree Test:** One of the main components in proving the PCP theorem is so called *low degree test* - checking if a given function  $f : \mathbf{F}_q^m \rightarrow \mathbf{F}_q$  is a low degree function or far from it. To reduce the number of queries, a tester can have access to more information, for instance, the tester can request supposedly restriction of  $f$  to any line/plane in  $\mathbf{F}_q^m$ .

Towards getting a better PCPs in terms of the number of bits accessed vs the success guarantee, one needs a low degree test such that even if the test accepts with tiny probability, the function must be close to a low degree function. To this end, Arora-Sudan[AS98] analysed line-vs-line test which queries the lines table at two intersecting line, selected u.a.r. and checks the consistency at the intersection point. The analysis of line-vs-line test is much more algebraic and involved. Instead of analyzing line-vs-line test, Raz-Safra[RS96] gave an analogous plane-vs-plane test and gave a combinatorial proof of the low degree test. Although Raz-Safra proof is simpler, they need to do induction on the dimension of the ambient space and hence the argument gets slightly involved. In my work with Dinur and Livni [BDL17], we gave an analysis of cube-vs-cube test which not only simplifies the overall analysis but also improves the soundness parameter compared to all the previously known low degree tests. More specifically, we show that if the cube-vs-cube test accepts with probability  $\varepsilon = \Omega(d^4/\sqrt{q})$

then the function  $f$  must be  $\Omega(\varepsilon)$ -close to a degree  $d$  function.

**Dictatorship Test:** A Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  is called a dictator if it depends on exactly one variable. Given a Boolean function, dictatorship test is a randomized test which queries a function at a few locations and based on it decides whether a function is a dictator or far from it. Suppose we want the test to have *perfect completeness* meaning that the test should never make an error if a given function is a dictator. Furthermore, we restrict the tester to query only  $k$  locations. What can we say about the success guarantee if  $f$  is far from dictator. Since dictatorship test is prevalent in proving inapproximability results based on Label Cover hardness or Unique Games Conjecture, the central question is - if a given function is far from any dictator, how small we can make the error probability of the tester under these restrictions.

In a joint work with Khot and Thiruvengatachari [BKT16], we improve the previously known bound. We give a randomized test which queries  $f$  at  $k$  locations and has following two guarantees: If  $f$  is a dictator then the tester accepts with probability 1. If  $f$  is  $\varepsilon$ -far from dictator (for appropriate notion of farness) then the tester accepts with probability at most  $\frac{2k+1}{2^k} + O(\varepsilon)$ . The previous work required the queried bits to satisfy pairwise independence condition and hence couldn't reach to  $\frac{2k+1}{2^k}$  soundness. We design a test which lacks pairwise independence condition but still proves the required soundness guarantee.

## 1.5 Organization

We start with some preliminaries in Chapter 2. The first three chapters present results about inapproximability and Parallel Repetition theorem. We describe the result on Covering CSPs in Chapter 3. In Chapter 4, we prove the parallel repetition theorem for certain class of games. This is followed by Chapter 5 proving inapproximability of Bi-COVERING. The next two chapters are related to Property Testing. In Chapter 6, we prove Cube vs Cube Low degree test which then followed by Chapter 7 about improved dictatorship test. Finally, the last couple of chapters study simultaneous CSPs. In Chapter 8, we present a constant factor approximation algorithm for all simultaneous

CPSs. In [Chapter 9](#), we give an improved approximation algorithm for simultaneous MAX-CUT.

## Chapter 2

### Preliminaries

In this chapter, we describe some preliminaries.

#### 2.1 Analysis of Boolean Functions

For a function  $f : \{0, 1\}^n \rightarrow \mathbb{R}$ , the *Fourier decomposition* of  $f$  is given by

$$f(x) = \sum_{\alpha \in \{0, 1\}^n} \widehat{f}(\alpha) \chi_\alpha(x),$$

where  $\chi_\alpha(x) := (-1)^{\sum_{i=1}^n \alpha_i \cdot x_i}$  and  $\widehat{f}(\alpha) := \mathbf{E}_{x \in \{0, 1\}^n} f(x) \chi_\alpha(x)$ . We will use  $\alpha$ , also to denote the subset of  $[n]$  for which it is the characteristic vector. Let  $\|f\|_2 := \mathbf{E}_{x \in \{0, 1\}^n} [f(x)^2]^{1/2}$  and  $\|f\|_\infty := \max_{x \in \{0, 1\}^n} |f(x)|$ .

The *Efron-Stein decomposition* is a generalization of the Fourier decomposition to product distributions of arbitrary probability spaces.

**Definition 2.1.1.** *Let  $(\Omega, \mu)$  be a probability space and  $(\Omega^n, \mu^{\otimes n})$  be the corresponding product space. For a function  $f : \Omega^n \rightarrow \mathbb{R}$ , the Efron-Stein decomposition of  $f$  with respect to the product space is given by*

$$f(x_1, \dots, x_n) = \sum_{\beta \subseteq [n]} f_\beta(x),$$

where  $f_\beta$  depends only on  $x_i$  for  $i \in \beta$  and for all  $\beta' \not\supseteq \beta$ ,  $a \in \Omega^{\beta'}$ ,  $\mathbf{E}_{x \in \mu^{\otimes n}} [f_\beta(x) \mid x_{\beta'} = a] = 0$ .

**Claim 2.1.2** (Parseval's Theorem). *For any  $f : \{0, 1\}^n \rightarrow \mathbb{R}$ ,*

$$\langle f, f \rangle = \mathbf{E}_{x \in \{0, 1\}^n} [f(x)^2] = \sum_{\alpha \subseteq [n]} \widehat{f}(\alpha)^2,$$

*In particular, if  $f$  takes values in  $\{-1, 1\}$  then  $\sum_{\alpha \subseteq [n]} \widehat{f}(\alpha)^2 = 1$ .*

We define the influence of  $i$ th variable as follows:

**Definition 2.1.3.** For  $i \in [n]$ , the influence of the  $i$ th coordinate on  $f$  is defined as follows.

$$\text{Inf}_i[f] := \mathbf{E}_{x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n} \text{Var}_{x_i}[f(x_1, \dots, x_n)] = \sum_{\beta: i \in \beta} \|f_\beta\|_2^2.$$

For an integer  $d$ , the degree  $d$  influence is defined as

$$\text{Inf}_i^{\leq d}[f] := \sum_{\beta: i \in \beta, |\beta| \leq d} \|f_\beta\|_2^2.$$

When  $f$  is a function from Boolean hypercube to reals, then under the uniform distribution, the  $i$ th influence is equal to  $\sum_{\alpha: i \in \alpha} \hat{f}(\alpha)^2$ .

It is easy to see that for Boolean functions, the sum of all the degree  $d$  influences is at most  $d$ . We prove this fact.

**Claim 2.1.4.** For all  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ ,  $\sum_i \text{Inf}_i^{\leq d}[f] \leq d$ .

*Proof.*

$$\begin{aligned} \sum_i \text{Inf}_i^{\leq d}[f] &= \sum_i \sum_{\alpha \subseteq [n], |\alpha| \leq d, i \in \alpha} \hat{f}(\alpha)^2 \\ &= \sum_{\alpha \subseteq [n], |\alpha| \leq d} |\alpha| \hat{f}(\alpha)^2 \\ &\leq d \sum_{\alpha \subseteq [n], |\alpha| \leq d} \hat{f}(\alpha)^2 \leq d, \end{aligned}$$

where the last inequality uses Parseval's Identity and the fact that  $\langle f, f \rangle \leq 1$ .  $\blacksquare$

A dictator is a function which depends on one variable. Thus, the degree 1 influence of any dictator function is 1 for some  $i \in [n]$ . We call a function *far* from any dictator if for every  $i \in [n]$ , the degree  $d$  influence is very small for some small  $d$ . This motivates the following definition.

**Definition 2.1.5** ( $(d, \tau)$ -quasirandom function). A multilinear function  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  is said to be  $(d, \tau)$ -quasirandom if for every  $i \in [n]$  it holds that

$$\sum_{i \in \alpha \subseteq [n], |\alpha| \leq d} \hat{f}(\alpha)^2 \leq \tau$$

We recall the Bonami-Beckner operator on Boolean functions.

**Definition 2.1.6.** For  $\gamma \in [0, 1]$ , the Bonami-Beckner operator  $T_{1-\gamma}$  is a linear operator mapping functions  $f : \{0, 1\}^n \rightarrow \mathbb{R}$  to functions  $T_{1-\gamma}f : \{0, 1\}^n \rightarrow \mathbb{R}$  as  $T_{1-\gamma}f(x) = \mathbf{E}_y[f(y)]$  where  $y$  is sampled by setting  $y_i = x_i$  with probability  $1 - \gamma$  and  $y_i$  to be uniformly random bit with probability  $\gamma$  for each  $i \in [n]$  independently. Let us denote the distribution of  $y$  given  $x$  as  $\mathcal{N}_{1-\gamma}(x)$ .

We have the following relation between the Fourier decomposition of  $T_{1-\gamma}f$  and  $f$ .

**Fact 2.1.7.**  $T_{1-\gamma}f = \sum_{\alpha \subseteq [n]} (1 - \gamma)^{|\alpha|} \hat{f}(\alpha) \chi_\alpha$ .

*Proof.* For  $\alpha \subseteq [n]$ ,

$$\begin{aligned} \widehat{T_{1-\gamma}f}(\alpha) &= \mathbf{E}_x[T_{1-\gamma}f(x) \chi_\alpha(x)] \\ &= \mathbf{E}_x \left[ \mathbf{E}_{y \sim \mathcal{N}_{1-\gamma}(x)} [f(y)] \chi_\alpha(x) \right] \\ &= \mathbf{E}_{\substack{x, \\ y \sim \mathcal{N}_{1-\gamma}(x)}} \left[ \sum_{\beta \subseteq [n]} \hat{f}(\beta) \chi_\beta(y) \chi_\alpha(x) \right] \\ &= \sum_{\beta \subseteq [n]} \hat{f}(\beta) \mathbf{E}_{\substack{x, \\ y \sim \mathcal{N}_{1-\gamma}(x)}} [\chi_\beta(y) \chi_\alpha(x)] \end{aligned}$$

Now, we use the fact that the marginals on each of  $x_i$  and  $y_i$  is uniform in  $\{0, 1\}$ . This, if  $\alpha \neq \beta$  the expectation is *zero*. Therefore,

$$\begin{aligned} \widehat{T_{1-\gamma}f}(\alpha) &= \hat{f}(\alpha) \mathbf{E}_{\substack{x, \\ y \sim \mathcal{N}_{1-\gamma}(x)}} [\chi_\alpha(x + y)] \\ &= (1 - \gamma)^{|\alpha|} \cdot \hat{f}(\alpha), \end{aligned}$$

where the last equality follows from the fact that  $x_i + y_i = 0$  with probability  $1 - \gamma$  and is uniform with probability  $\gamma$ . ■

## 2.2 Correlated Spaces

We need a few definitions and lemmas related to correlated spaces defined by Mosser [Mos10]. Let  $\Omega_1 \times \Omega_2$  be two correlated spaces and  $\mu$  denotes the joint distribution.

Let  $\mu_1$  and  $\mu_2$  denote the marginal of  $\mu$  on space  $\Omega_1$  and  $\Omega_2$  respectively. The correlated space  $\rho(\Omega_1 \times \Omega_2; \mu)$  can be represented as a bipartite graph on  $(\Omega_1, \Omega_2)$  where  $x \in \Omega_1$  is connected to  $y \in \Omega_2$  iff  $\mu(x, y) > 0$ . We say that the correlated spaces is *connected* if this underlying graph is connected.

**Definition 2.2.1.** *Let  $(\Omega_1 \times \Omega_2, \mu)$  be a finite correlated space, the correlation between  $\Omega_1$  and  $\Omega_2$  with respect to  $\mu$  is defined as*

$$\rho(\Omega_1, \Omega_2; \mu) := \max_{\substack{f: \Omega_1 \rightarrow \mathbb{R}, \mathbf{E}[f]=0, \mathbf{E}[f^2] \leq 1 \\ g: \Omega_2 \rightarrow \mathbb{R}, \mathbf{E}[g]=0, \mathbf{E}[g^2] \leq 1}} \mathbf{E}_{(x,y) \sim \mu} [|f(x)g(y)|].$$

The following result (from [Mos10]) provides a way to upper bound correlation of a correlated spaces.

**Lemma 2.2.2.** *Let  $(\Omega_1 \times \Omega_2, \mu)$  be a finite correlated space such that the probability of the smallest atom in  $\Omega_1 \times \Omega_2$  is at least  $\alpha > 0$  and the correlated space is connected then*

$$\rho(\Omega_1, \Omega_2; \mu) \leq 1 - \alpha^2/2$$

**Definition 2.2.3** (Markov Operator). *Let  $(\Omega_1 \times \Omega_2, \mu)$  be a finite correlated space, the Markov operator, associated with this space, denoted by  $U$ , maps a function  $g : \Omega_2 \rightarrow \mathbb{R}$  to functions  $Ug : \Omega_1 \rightarrow \mathbb{R}$  by the following map:*

$$(Ug)(x) := \mathbf{E}_{(X,Y) \sim \mu} [g(Y) \mid X = x].$$

The following results (from [Mos10]) provide a way to upper bound correlation of a correlated spaces.

**Lemma 2.2.4** ([Mos10, Lemma 2.8]). *Let  $(\Omega_1 \times \Omega_2, \mu)$  be a finite correlated space. Let  $g : \Omega_2 \rightarrow \mathbb{R}$  be such that  $\mathbf{E}_{(x,y) \sim \mu} [g(y)] = 0$  and  $\mathbf{E}_{(x,y) \sim \mu} [g(y)^2] \leq 1$ . Then, among all functions  $f : \Omega_1 \rightarrow \mathbb{R}$  that satisfy  $\mathbf{E}_{(x,y) \sim \mu} [f(x)^2] \leq 1$ , the maximum value of  $|\mathbf{E}[f(x)g(y)]|$  is given as:*

$$|\mathbf{E}[f(x)g(y)]| = \sqrt{\mathbf{E}_{(x,y) \sim \mu} [(Ug(x))^2]}.$$

**Proposition 2.2.5** ([Mos10, Proposition 2.11]). *Let  $(\prod_{i=1}^n \Omega_i^{(1)} \times \prod_{i=1}^n \Omega_i^{(2)}, \prod_{i=1}^n \mu_i)$  be a product correlated spaces. Let  $g : \prod_{i=1}^n \Omega_i^{(2)} \rightarrow \mathbb{R}$  be a function and  $U$  be the Markov*

operator mapping functions from space  $\prod_{i=1}^n \Omega_i^{(2)}$  to the functions on space  $\prod_{i=1}^n \Omega_i^{(1)}$ . If  $g = \sum_{S \subseteq [n]} g_S$  and  $Ug = \sum_{S \subseteq [n]} (Ug)_S$  be the Efron-Stein decomposition of  $g$  and  $Ug$  respectively then,

$$(Ug)_S = U(g_S)$$

i.e. the Efron-Stein decomposition commutes with Markov operators.

**Proposition 2.2.6** ([Mos10, Proposition 2.12]). Assume the setting of [Proposition 2.2.5](#) and furthermore assume that  $\rho(\Omega_i^{(1)}, \Omega_i^{(2)}; \mu_i) \leq \rho$  for all  $i \in [n]$ , then for all  $g$  it holds that

$$\|U(g_S)\|_2 \leq \rho^{|S|} \|g_S\|_2.$$

## 2.3 Hypercontractivity

**Definition 2.3.1.** A random variable  $r$  is said to be  $(p, q, \eta)$ -hypercontractive if it satisfies

$$\|a + \eta r\|_q \leq \|a + r\|_p$$

for all  $a \in \mathbb{R}$ .

We note down the hypercontractive parameters for Rademacher random variable (uniform over  $\pm 1$ ) and standard gaussian random variable.

**Theorem 2.3.2** ([Wol07][Ole03]). Let  $X$  denote either a uniformly random  $\pm 1$  bit, a standard one-dimensional Gaussian. Then  $X$  is  $\left(2, q, \frac{1}{\sqrt{q-1}}\right)$ -hypercontractive.

The following proposition says that the higher norm of a low degree function w.r.t hypercontractive sequence of ensembles is bounded above by its second norm.

**Proposition 2.3.3** ([MOO05]). Let  $\mathbf{x}$  be a  $(2, q, \eta)$ -hypercontractive sequence of ensembles and  $Q$  be a multilinear polynomial of degree  $d$ . Then

$$\|Q(\mathbf{x})\|_q \leq \eta^{-d} \|Q(\mathbf{x})\|_2$$

## 2.4 Invariance Principle

Let  $(\Omega^k, \mu)$  be a probability space. Let  $S = \{x \in \Omega^k \mid \mu(x) > 0\}$ . We say that  $S \subseteq \Omega^k$  is *connected* if for every  $x, y \in S$ , there is a sequence of strings starting with  $x$  and ending with  $y$  such that every element in the sequence is in  $S$  and every two adjacent elements differ in exactly one coordinate.

**Theorem 2.4.1** ([Mos10, Proposition 6.4]). *Let  $(\Omega^k, \mu)$  be a probability space such that the support of the distribution  $\text{supp}(\mu) \subseteq \Omega^k$  is connected and the minimum probability of every atom in  $\text{supp}(\mu)$  is at least  $\alpha$  for some  $\alpha \in (0, \frac{1}{2}]$ . Then there exists continuous functions  $\bar{\Gamma} : (0, 1) \rightarrow (0, 1)$  and  $\underline{\Gamma} : (0, 1) \rightarrow (0, 1)$  such that the following holds: For every  $\varepsilon > 0$ , there exists  $\tau > 0$  and an integer  $d$  such that if a function  $f : \Omega^L \rightarrow [0, 1]$  satisfies*

$$\forall i \in [n], \text{Inf}_i^{\leq d}(f) \leq \tau$$

*then*

$$\underline{\Gamma}\left(\mathbf{E}_{\mu}[f]\right) - \varepsilon \leq \mathbf{E}_{(x_1, \dots, x_k) \sim \mu} \left[ \prod_{j=1}^k f(x_j) \right] \leq \bar{\Gamma}\left(\mathbf{E}_{\mu}[f]\right) + \varepsilon.$$

*There exists an absolute constant  $C$  such that one can take  $\tau = \varepsilon^{C \frac{\log(1/\alpha) \log(1/\varepsilon)}{\varepsilon \alpha^2}}$  and  $d = \log(1/\tau) \log(1/\alpha)$ .*

The following invariance principle for correlated spaces is an adaptation of similar invariance principles (c.f., [Wen13, Theorem 3.12], [GL15, Lemma A.1]) to our setting.

**Theorem 2.4.2** (Invariance Principle for correlated spaces). *Let  $(\Omega_1^k \times \Omega_2^k, \mu)$  be a correlated probability space such that the marginal of  $\mu$  on any pair of coordinates one each from  $\Omega_1$  and  $\Omega_2$  is a product distribution. Let  $\mu_1, \mu_2$  be the marginals of  $\mu$  on  $\Omega_1^k$  and  $\Omega_2^k$  respectively. Let  $X, Y$  be two random  $k \times L$  dimensional matrices chosen as follows: independently for every  $i \in [L]$ , the pair of columns  $(x^i, y^i) \in \Omega_1^k \times \Omega_2^k$  is chosen from  $\mu$ . Let  $x_i, y_i$  denote the  $i$ th rows of  $X$  and  $Y$  respectively. If  $F : \Omega_1^L \rightarrow [-1, +1]$  and  $G : \Omega_2^L \rightarrow [-1, +1]$  are functions such that*

$$\tau := \sqrt{\sum_{i \in [L]} \text{Inf}_i[F] \cdot \text{Inf}_i[G]} \quad \text{and} \quad \Gamma := \max \left\{ \sqrt{\sum_{i \in [L]} \text{Inf}_i[F]}, \sqrt{\sum_{i \in [L]} \text{Inf}_i[G]} \right\},$$

then

$$\left| \mathbf{E}_{(X,Y) \in \mu^{\otimes L}} \left[ \prod_{i \in [k]} F(x_i) G(y_i) \right] - \mathbf{E}_{X \in \mu_1^{\otimes L}} \left[ \prod_{i \in [k]} F(x_i) \right] \mathbf{E}_{Y \in \mu_2^{\otimes L}} \left[ \prod_{i \in [k]} G(y_i) \right] \right| \leq 2^{O(k)} \Gamma \tau. \quad (2.4.1)$$

*Proof.* We will prove the theorem by using the hybrid argument. For  $i \in [L+1]$ , let  $X^{(i)}, Y^{(i)}$  be distributed according to  $(\mu_1 \otimes \mu_2)^{\otimes i} \otimes \mu^{\otimes L-i}$ . Thus,  $(X^{(0)}, Y^{(0)}) = (X, Y)$  is distributed according to  $\mu^{\otimes L}$  while  $(X^{(L)}, Y^{(L)})$  is distributed according to  $(\mu_1 \otimes \mu_2)^{\otimes L}$ . For  $i \in [L]$ , define

$$\text{err}_i := \left| \mathbf{E}_{X^{(i)}, Y^{(i)}} \left[ \prod_{j=1}^k F(x_j^{(i)}) G(y_j^{(i)}) \right] - \mathbf{E}_{X^{(i+1)}, Y^{(i+1)}} \left[ \prod_{j=1}^k F(x_j^{(i+1)}) G(y_j^{(i+1)}) \right] \right|. \quad (2.4.2)$$

The left hand side of Equation (2.4.1) is upper bounded by  $\sum_{i \in [L]} \text{err}_i$ . Now for a fixed  $i$ , we will bound  $\text{err}_i$ . We use the Efron-Stein decomposition of  $F, G$  to split them into two parts: the part which depends on the  $i$ th input and the part independent of the  $i$ th input.

$$F = F_0 + F_1 \text{ where } F_0 := \sum_{\alpha: i \notin \alpha} F_\alpha \text{ and } F_1 := \sum_{\alpha: i \in \alpha} F_\alpha.$$

$$G = G_0 + G_1 \text{ where } G_0 := \sum_{\beta: i \notin \beta} G_\beta \text{ and } G_1 := \sum_{\beta: i \in \beta} G_\beta.$$

Note that  $\text{Inf}_i[F] = \|F_1\|_2^2$  and  $\text{Inf}_i[G] = \|G_1\|_2^2$ . Furthermore, the functions  $F_0$  and  $F_1$  are bounded since  $F_0(x) = \mathbf{E}_{x'}[F(x') | x'_{[L] \setminus i} = x_{[L] \setminus i}] \in [-1, +1]$  and  $F_1(x) = F(x) - F_0(x) \in [-2, +2]$ . For  $a \in \{0, 1\}^k$ , let  $F_a(X) := \prod_{j=1}^k F_{a_j}(x_j)$ . Similarly  $G_0, G_1$  are bounded and  $G_a$  defined analogously. Substituting these definitions in Equation (2.4.2) and expanding the products gives

$$\text{err}_i = \left| \sum_{a, b \in \{0, 1\}^k} \left( \mathbf{E}_{X^{(i)}, Y^{(i)}} [F_a(X^{(i)}) G_b(Y^{(i)})] - \mathbf{E}_{X^{(i+1)}, Y^{(i+1)}} [F_a(X^{(i+1)}) G_b(Y^{(i+1)})] \right) \right|.$$

Since both the distributions are identical on  $(\Omega_1^k)^{\otimes L}$  and  $(\Omega_2^k)^{\otimes L}$ , all terms with  $a = \bar{0}$  or  $b = \bar{0}$  are zero. Because  $\mu$  is uniform on any pair of coordinates on each from the  $\Omega_1$  and  $\Omega_2$  sides, terms with  $|a| = |b| = 1$  also evaluates to zero. Now consider the

remaining terms with  $|a|, |b| \geq 1, |a| + |b| > 2$ . Consider one such term where  $a_1, a_2 = 1$  and  $b_1 = 1$ . In this case, by Cauchy-Schwarz inequality we have that

$$\left| \mathbf{E}_{X^{(i-1)}, Y^{(i-1)}} \left[ F_a(X^{(i-1)}) G_b(Y^{(i-1)}) \right] \right| \leq \sqrt{\mathbf{E} F_1(x_1)^2 G_1(y_1)^2} \cdot \|F_1\|_2 \cdot \left\| \prod_{j>2} F_{a_j} \right\|_\infty \cdot \left\| \prod_{j>1} G_{b_j} \right\|_\infty.$$

From the facts that the marginal of  $\mu$  to any pair of coordinates one each from  $\Omega_1$  and  $\Omega_2$  sides are uniform,  $\text{Inf}_i[F] = \|F_1\|_2^2$  and  $|F_0(x)|, |F_1(x)|, |G_0(x)|, |G_1(x)|$  are all bounded by 2, the right side of above becomes

$$\sqrt{\mathbf{E} F_1(x_1)^2} \sqrt{\mathbf{E} G_1(y_1)^2} \cdot \|F_1\|_2 \cdot \left\| \prod_{j>2} F_{a_j} \right\|_\infty \cdot \left\| \prod_{j>1} G_{b_j} \right\|_\infty \leq \sqrt{\text{Inf}_i[F]^2 \text{Inf}_i[G]} \cdot 2^{2k}.$$

All the other terms corresponding to other  $(a, b)$  which are at most  $2^{2k}$  in number, are bounded analogously. Hence,

$$\begin{aligned} \sum_{i \in [L]} \text{err}_i &\leq 2^{4k} \sum_{i \in [L]} \left( \sqrt{\text{Inf}_i[F]^2 \text{Inf}_i[G]} + \sqrt{\text{Inf}_i[F] \text{Inf}_i[G]^2} \right) \\ &= 2^{4k} \sum_{i \in [L]} \sqrt{\text{Inf}_i[F] \text{Inf}_i[G]} \left( \sqrt{\text{Inf}_i[F]} + \sqrt{\text{Inf}_i[G]} \right). \end{aligned}$$

By applying the Cauchy-Schwarz inequality, followed by a triangle inequality, we obtain

$$\sum_{i \in [L]} \text{err}_i \leq 2^{4k} \sqrt{\sum_{i \in [L]} \text{Inf}_i[F] \text{Inf}_i[G]} \left( \sqrt{\sum_{i \in [L]} \text{Inf}_i[F]} + \sqrt{\sum_{i \in [L]} \text{Inf}_i[G]} \right).$$

Thus, proved. ■

Let  $\mathbb{F}_q$  be any finite field.

**Definition 2.4.3** (Symmetric Markov Operator). *Symmetric Markov operator on  $\mathbb{F}_q$  can be thought of as a random walk on an undirected graph with the vertex set  $\mathbb{F}_q$ . It can be represented as a  $q \times q$  matrix  $T$  where  $(i, j)$  th entry is the probability of moving to vertex  $j$  from  $i$ .*

**Definition 2.4.4.** *For a symmetric Markov operator  $T$ , let  $1 = \lambda_0 \geq \lambda_1 \geq \lambda_2 \dots \geq \lambda_{q-1}$  be the eigenvalues of  $T$  in a non-increasing order. The spectral radius of  $T$ , denoted by  $r(T)$ , is defined as:*

$$r(T) = \max\{|\lambda_1|, |\lambda_{q-1}|\}$$

For a Markov operator  $T$  the condition  $r(T) < 1$  is equivalent to saying that the induced regular graph (self-loop allowed) on  $\mathbb{F}_q$  is non-bipartite and connected.

For  $T$  as above, we also define a Markov operator  $T^{\otimes n}$  on  $[q]^n$  in a natural way i.e applying a Markov operator  $T^{\otimes n}$  to  $x \in [q]^n$  is same as applying the Markov operator  $T$  on each  $x_i$  independently. Note that if  $T$  is symmetric then  $T^{\otimes n}$  is also symmetric and  $r(T^{\otimes n}) = r(T)$ .

We will need the following Gaussian stability measure in our analysis:

**Definition 2.4.5.** Let  $\phi : \mathbb{R} \rightarrow [0, 1]$  be the cumulative distribution function of the standard Gaussian random variable. For a parameter  $\rho, \mu, \nu \in [0, 1]$ , we define the following two quantities:

$$\underline{\Gamma}_\rho(\mu, \nu) = \Pr[X \leq \phi^{-1}(\mu), Y \geq \phi^{-1}(1 - \nu)]$$

$$\overline{\Gamma}_\rho(\mu, \nu) = \Pr[X \leq \phi^{-1}(\mu), Y \leq \phi^{-1}(\nu)]$$

where  $X$  and  $Y$  are two standard Gaussian variables with covariance  $\rho$ .

We are now ready to state the invariance principle from [DMR09] that we need for our reduction in Chapter 7

**Theorem 2.4.6** ([DMR09]). Let  $T$  be a symmetric Markov operator on  $\mathbb{F}_q$  such that  $\rho = r(T) < 1$ . Then for any  $\tau > 0$  there exists  $\delta > 0$  and  $k \in \mathbb{N}$  such that if  $f, g : \mathbb{F}_q^n \rightarrow [0, 1]$  are two functions satisfying

$$\min(\mathbf{Inf}_i^{\leq k}(f), \mathbf{Inf}_i^{\leq k}(g)) \leq \delta$$

for all  $i \in [n]$ , then it holds that

$$\langle f, T^{\otimes n} g \rangle \geq \underline{\Gamma}_\rho(\mu, \nu) - \tau$$

where  $\mu = \mathbf{E}[f]$ ,  $\nu = \mathbf{E}[g]$ .

## 2.5 Expanders & Extractors

**Definition 2.5.1** (Expanders). *For a symmetric, stochastic matrix  $M$ , define*

$$\lambda(M) \stackrel{\text{def}}{=} \max_{\mathbf{v} \perp \mathbf{1}} \frac{\|M\mathbf{v}\|}{\|\mathbf{v}\|}$$

*A  $D$ -regular graph  $H = (X, E)$  is a  $\lambda$ -expander, if  $\lambda(H) \leq \lambda$ , where  $H$  is the normalized adjacency matrix of the graph  $H$ .*

*For a symmetric bipartite graph  $G = ((X, X), E)$ , we say  $G$  is a bipartite  $\lambda$ -expander if  $\lambda(H) \leq \lambda$  where  $H$  is the normalized biadjacency matrix of  $G$ .*

Henceforth, when we refer to a bipartite graph as being a  $\lambda$ -expander, we implicitly mean a *bipartite*  $\lambda$ -expander.

Any expander  $H = (X, E_H)$  can be transformed to a natural bipartite expander  $H'$  on  $X \times X$ , by including the edge  $(x, x')$  and  $(x', x)$  to  $H'$  for every  $(x, x') \in E_H$ . We shall abuse notation and call this graph  $H' = ((X, X), E_H)$  although each edge in  $H$  occurs “twice” in  $H'$ .

**Lemma 2.5.2** (Explicit expanders [BL06]). *For every  $D > 0$ , there exists a fully explicit family of graphs  $\{G_i\}$ , such that  $G_i$  is  $D$ -regular and  $\lambda(G_i) \leq D^{-1/2}(\log D)^{3/2}$ .*

**Definition 2.5.3** (Extractors). *A bipartite graph  $H = ((X, Y), E)$  is an  $(\delta, \varepsilon)$ -extractor if for every subset  $S \subseteq X$  such that  $|S| \geq \delta|X|$ , if  $\pi$  is the induced probability distribution on  $Y$  by taking a random element of  $S$  and a random neighbour, then*

$$|\pi - \mathbf{u}|_1 \leq \varepsilon.$$

**Lemma 2.5.4** (Explicit Extractors [RVW00]). *There exists explicit  $(\delta, \varepsilon)$ -extractors  $G = (X, Y, E)$  such that  $|X| = O(|Y|/\delta)$  and each vertex of  $X$  has degree  $D = O(\exp(\text{poly}(\log \log(1/\delta))) \cdot (1/\varepsilon^2))$ .*

## Chapter 3

### Covering CSP

#### 3.1 Introduction

One of the central (yet unresolved) questions in inapproximability is the problem of coloring a (hyper)graph with as few colors as possible. A (hyper)graph  $G = (V, E)$  is said to be  $k$ -colorable if there exists a coloring  $c : V \rightarrow [k] := \{0, 1, 2, \dots, k-1\}$  of the vertices such that no (hyper)edge of  $G$  is monochromatic. The chromatic number of a (hyper)graph, denoted by  $\chi(G)$ , is the smallest  $k$  such that  $G$  is  $k$ -colorable. It is known that computing  $\chi(G)$  to within a multiplicative factor of  $n^{1-\varepsilon}$  on an  $n$ -sized graph  $G$  for every  $\varepsilon \in (0, 1)$  is NP-hard. However, the complexity of the following problem is not yet completely understood: given a constant-colorable (hyper)graph, what is the minimum number of colors required to color the vertices of the graph efficiently such that every edge is non-monochromatic? The current best approximation algorithms for this problem require at least  $n^{\Omega(1)}$  colors while the hardness results are far from proving optimality of these approximation algorithms (see § 3.1.3 for a discussion on recent work in this area).

The notion of *covering complexity* was introduced by Guruswami, Håstad and Sudan [GHS02] and more formally by Dinur and Kol [DK13] to obtain a better understanding of the complexity of this problem. Let  $P$  be a predicate and  $\Phi$  an instance of a constraint satisfaction problem (CSP) over  $n$  variables, where each constraint in  $\Phi$  is a constraint of type  $P$  over the  $n$  variables and their negations. We will refer to such CSPs as  $P$ -CSPs. The *covering number* of  $\Phi$ , denoted by  $\nu(\Phi)$ , is the smallest number of assignments to the variables such that each constraint of  $\Phi$  is satisfied by at least one of the assignments, in which case we say that the set of assignments *covers* the instance  $\Phi$ . If  $c$  assignments cover the instance  $\Phi$ , we say that  $\Phi$  is  $c$ -coverable or

equivalently that the set of assignments form a  $c$ -covering for  $\Phi$ . The covering number is a generalization of the notion of chromatic number (to be more precise, the logarithm of the chromatic number) to all predicates in the following sense. Suppose  $P$  is the not-all-equal predicate **NAE** and the instance  $\Phi$  has no negations in any of its constraints, then the covering number  $\nu(\Phi)$  is exactly  $\lceil \log \chi(G_\Phi) \rceil$  where  $G_\Phi$  is the underlying constraint graph of the instance  $\Phi$ .

Cover- $P$  refers to the problem of finding the covering number of a given  $P$ -CSP instance. Finding the exact covering number for most interesting predicates  $P$  is NP-hard. We therefore study the problem of approximating the covering number. In particular, we would like to study the complexity of the following problem, denoted by **COVERING- $P$ -CSP**  $(c, s)$ , for some  $1 \leq c < s \in \mathbb{N}$ : “given a  $c$ -coverable  $P$ -CSP instance  $\Phi$ , find an  $s$ -covering for  $\Phi$ ”. Similar problems have been studied for the Max-CSP setting: “for  $0 < s < c \leq 1$ , “given a  $c$ -satisfiable  $P$ -CSP instance  $\Phi$ , find an  $s$ -satisfying assignment for  $\Phi$ ”. Max-CSPs and Cover-CSPs, as observed by Dinur and Kol [DK13], are very different problems. For instance, if  $P$  is an odd predicate, i.e, if for every assignment  $x$ , either  $x$  or its negation  $x + \bar{1}$  satisfies  $P$ , then any  $P$ -CSP instance  $\Phi$  has a trivial two covering, any assignment and its negation. Thus, **3-LIN** and **3-CNF**<sup>1</sup>, being odd predicates, are easy to cover though they are hard predicates in the Max-CSP setting. The main result of Dinur and Kol is that the **4-LIN** predicate, in contrast to the above, is hard to cover: for every constant  $t \geq 2$ , **COVERING-4-LIN-CSP**  $(2, t)$  is NP-hard. In fact, their arguments show that **COVERING-4-LIN-CSP**  $(2, \Omega(\log \log \log n))$  is quasi-NP-hard.

Having observed that odd predicate based CSPs are easy to cover, Dinur and Kol proceeded to ask the question “are all non-odd-predicate CSPs hard to cover?”. In a partial answer to this question, they showed that assuming a covering variant of the unique games conjecture **COVERING-UGC**( $c$ ), if a predicate  $P$  is not odd and there is a balanced pairwise independent distribution on its support, then for all constants  $k$ , **COVERING- $P$ -CSP**  $(2c, k)$  is NP-hard (here,  $c$  is a fixed constant that depends on the

---

<sup>1</sup>**3-LIN** :  $\{0, 1\}^3 \rightarrow \{0, 1\}$  refers to the 3-bit predicate defined by  $\mathbf{3-LIN}(x_1, x_2, x_3) := x_1 \oplus x_2 \oplus x_3$  while **3-CNF** :  $\{0, 1\}^3 \rightarrow \{0, 1\}$  refers to the 3-bit predicate defined by  $\mathbf{3-CNF}(x_1, x_2, x_3) := x_1 \vee x_2 \vee x_3$

covering variant of the unique games conjecture  $\text{COVERING-UGC}(c)$ ). See § 3.2 for the exact definition of the covering variant of the unique games conjecture.

### 3.1.1 Results

Our first result states that assuming the same covering variant of unique games conjecture

$\text{COVERING-UGC}(c)$  of Dinur and Kol [DK13], one can in fact show the covering hardness of *all* non-odd predicates  $P$  over *any* constant-sized alphabet  $[q]$ . The notion of odd predicate can be extended to any alphabet in the following natural way: a predicate  $P \subseteq [q]^k$  is odd if for all assignments  $x \in [q]^k$ , there exists  $a \in [q]$  such that the assignment  $x + \bar{a}$  satisfies  $P$ .

**Theorem 3.1.1** (Covering hardness of non-odd predicates). *Assuming  $\text{COVERING-UGC}(c)$ , for any constant-sized alphabet  $[q]$ , any constant  $k \in \mathbb{N}$  and any non-odd predicate  $P \subseteq [q]^k$ , for all constants  $t \in \mathbb{N}$ , the  $\text{COVERING-}P\text{-CSP}(2cq, t)$  problem is NP-hard.*

Since odd predicates  $P \subseteq [q]^k$  are trivially coverable with  $q$  assignments, the above theorem, gives a *full characterization of hard-to-cover predicates* over any constant sized alphabet (modulo the covering variant of the unique games conjecture): a predicate is hard to cover iff it is not odd.

We then ask if we can prove similar covering hardness results under more standard complexity assumptions (such as  $\text{NP} \neq \text{P}$  or the exponential-time hypothesis (ETH)). Though we are not able to prove that every non-odd predicate is hard under these assumptions, we give sufficient conditions on the predicate  $P$  for the corresponding approximate covering problem to be quasi-NP-hard. Recall that  $2k\text{-LIN} \subseteq \{0, 1\}^{2k}$  is the predicate corresponding to the set of odd parity strings in  $\{0, 1\}^{2k}$ .

**Theorem 3.1.2** (NP-hardness of Covering). *Let  $k \geq 2$ . Let  $P \subseteq 2k\text{-LIN}$  be any  $2k$ -bit predicate such there exists distributions  $\mathcal{P}_0, \mathcal{P}_1$  supported on  $\{0, 1\}^k$  with the following properties:*

1. *the marginals of  $\mathcal{P}_0$  and  $\mathcal{P}_1$  on all  $k$  coordinates is uniform,*

2. every  $a \in \text{supp}(\mathcal{P}_0)$  has even parity and every  $b \in \text{supp}(\mathcal{P}_1)$  has odd parity and furthermore, both  $a \cdot b, b \cdot a \in P$ .

Then, unless  $NP \subseteq DTIME(2^{\text{poly} \log n})$ , for all  $\varepsilon \in (0, 1/2]$ , COVERING- $P$ -CSP  $(2, \Omega(\log \log n))$  is not solvable in polynomial time.

Furthermore, the YES and NO instances of COVERING- $P$ -CSP  $(2, \Omega(\log \log n))$  satisfy the following properties.

- YES Case : There are 2 assignments such that each of them covers  $1 - \varepsilon$  fraction of the constraints and they together cover the instance.
- NO Case : Even the 2k-LIN-CSP instance with the same constraint graph as the given instance is not  $\Omega(\log \log n)$ -coverable.

The furthermore clause in the soundness guarantee is in fact a strengthening for the following reason: if two predicates  $P, Q$  satisfy  $P \subseteq Q$  and  $\Phi$  is a  $c$ -coverable  $P$ -CSP instance, then the  $Q$ -CSP instance  $\Phi_{P \rightarrow Q}$  obtained by taking the constraint graph of  $\Phi$  and replacing each  $P$  constraint with the weaker  $Q$  constraint, is also  $c$ -coverable.

The following is a simple corollary of the above theorem.

**Corollary 3.1.3.** *Let  $k \geq 2$  be even,  $x, y \in \{0, 1\}^k$  be distinct strings having even and odd parity respectively and  $\bar{x}, \bar{y}$  denote the complements of  $x$  and  $y$  respectively. For any predicate  $P$  satisfying*

$$2k\text{-LIN} \supseteq P \supseteq \{x \cdot y, x \cdot \bar{y}, \bar{x} \cdot y, \bar{x} \cdot \bar{y}, y \cdot x, y \cdot \bar{x}, \bar{y} \cdot x, \bar{y} \cdot \bar{x}\},$$

*unless  $NP \subseteq DTIME(2^{\text{poly} \log n})$ , the problem COVERING- $P$ -CSP  $(2, \Omega(\log \log n))$  is not solvable in polynomial time.*

This corollary implies the covering hardness of 4-LIN predicate proved by Dinur and Kol [DK13] by setting  $x := 00$  and  $y := 01$ . With respect to the covering hardness of 4-LIN, we note that we can considerably simplify the proof of Dinur and Kol and in fact obtain a even stronger soundness guarantee (see Theorem below). The stronger soundness guarantee in the theorem below states that there are no large ( $\geq 1/\text{poly} \log n$

fractional sized) independent sets in the constraint graph and hence, even the 4-NAE-CSP instance<sup>2</sup> with the same constraint graph as the given instance is not coverable using  $\Omega(\log \log n)$  assignments. Both the Dinur-Kol result and the above corollary only guarantee (in the soundness case) that the 4-LIN-CSP instance is not coverable.

**Theorem 3.1.4** (Hardness of Covering 4-LIN). *Assuming that  $NP \not\subseteq DTIME(2^{\text{poly} \log n})$ , for all  $\varepsilon \in (0, 1)$ , there does not exist a polynomial time algorithm that can distinguish between 4-LIN-CSP instances of the following two types:*

- *YES Case : There are 2 assignments such that each of them covers  $1 - \varepsilon$  fraction of the constraints, and they together cover the entire instance.*
- *NO Case : The largest independent set in the constraint graph of the instance is of fractional size at most  $1/\text{poly} \log n$ .*

### 3.1.2 Techniques

As one would expect, our proofs are very much inspired from the corresponding proofs in Dinur and Kol [DK13]. One of the main complications in the proof of Dinur and Kol [DK13] (as also in the earlier work of Guruswami, Håstad and Sudan [GHS02]) was the one of handling several assignments simultaneously while proving the soundness analysis. For this purpose, both these works considered the rejection probability that all the assignments violated the constraint. This resulted in a very tedious expression for the rejection probability, which made the rest of the proof fairly involved. Khot [Kho02b] observed that this can be considerably simplified if one instead proved a stronger soundness guarantee that the largest independent set in the constraint graph is small (this might not always be doable, but in the cases when it is, it simplifies the analysis). We list below the further improvements in the proof that yield our Theorems 3.1.1, 3.1.2 and 3.1.4.

**Covering hardness of 4-LIN (Theorem 3.1.4):** The simplified proof of the covering hardness of 4-LIN follows directly from the above observation of using an

---

<sup>2</sup>The  $k$ -NAE predicate over  $k$  bits is given by  $k\text{-NAE} = \{0, 1\}^k \setminus \{\bar{0}, \bar{1}\}$ .

independent set analysis instead of working with several assignments. In fact, this alternate proof eliminates the need for using results about correlated spaces [Mos10], which was crucial in the Dinur-Kol setting. We further note that the quantitative improvement in the covering hardness ( $\Omega(\log \log n)$  over  $\Omega(\log \log \log n)$ ) comes from using a LABEL-COVER instance with a better smoothness property (see Theorem 3.2.5).

**Covering UG-hardness for non-odd predicates (Theorem 3.1.1):** Having observed that it suffices to prove an independent set analysis, we observed that only very mild conditions on the predicate are required to prove covering hardness. In particular, while Dinur and Kol used the Austrin-Mossel test [AM09] which required pairwise independence, we are able to import the long-code test of Bansal and Khot [BK10] which requires only 1-wise independence. We remark that the Bansal-Khot Test was designed for a specific predicate (hardness of finding independent sets in almost  $k$ -partite  $k$ -uniform hypergraphs) and had imperfect completeness. Our improvement comes from observing that their test requires only 1-wise independence and furthermore that their completeness condition, though imperfect, can be adapted to give a 2-cover composed of 2 nearly satisfying assignments. This enlarges the class of non-odd predicates for which one can prove covering hardness (see Theorem 3.3.1). We then perform a sequence of reductions from this class of CSP instances to CSP instances over all non-odd predicates to obtain the final result. Interestingly, one of the open problems mentioned in the work of Dinur and Kol [DK13] was to devise “direct” reductions between covering problems. The reductions we employ, strictly speaking, are not “direct” reductions between covering problems, since they rely on a stronger soundness guarantee for the source instance (namely, large covering number even for the NAE instance on the same constraint graph), which we are able to prove in Theorem 3.3.1.

**Quasi-NP-hardness result (Theorem 3.1.2):** In this setting, we unfortunately are not able to use the simplification arising from using the independent set analysis and have to deal with the issue of several assignments. One of the steps in the 4-LIN proof of Dinur and Kol (as in several others results in this area) involves showing

that an expression of the form  $\mathbf{E}_{(X,Y)}[F(X)F(Y)]$  is not too negative where  $(X,Y)$  is not necessarily a product distribution but the marginals on the  $X$  and  $Y$  parts are identical. Observe that if  $(X,Y)$  was a product distribution, then the above expression reduces to  $(\mathbf{E}_X[F(X)])^2$ , a positive quantity. Thus, the steps in the proof involve constructing a tailor-made distribution  $(X,Y)$  such that the error in going from the correlated probability space  $(X,Y)$  to the product distribution  $(X \otimes Y)$  is not too much. More precisely, the quantity

$$\left| \mathbf{E}_{(X,Y)}[F(X)F(Y)] - \mathbf{E}_X[F(X)] \mathbf{E}_Y[F(Y)] \right|,$$

is small. Dinur and Kol used a distribution tailor-made for the 4-LIN predicate and used an invariance principle for correlated spaces to bound the error while transforming it to a product distribution. Our improvement comes from observing that one could use an alternate invariance principle (see [Theorem 2.4.2](#)) that works with milder restrictions and hence works for a wider class of predicates. This invariance principle for correlated spaces ([Theorem 2.4.2](#)) is an adaptation of invariance principles proved by Wenner [[Wen13](#)] and Guruswami and Lee [[GL15](#)] in similar contexts. The rest of the proof is similar to the 4-LIN covering hardness proof of Dinur and Kol.

### 3.1.3 Recent work on approximate coloring

We remark that recently, with the discovery of the short code [[BGH<sup>+</sup>12](#)], there has been a sequence of works [[DG13](#), [GHH<sup>+</sup>14](#), [KS14](#), [Var14](#)] which have considerably improved the status of the approximate coloring question, stated in the beginning of the introduction. In particular, we know that it is quasi-NP-hard to color a 2-colorable 8-uniform hypergraph with  $2^{(\log n)^c}$  colors for some constant  $c \in (0, 1)$ . Stated in terms of covering number, this result states that it is quasi-NP-hard to cover a 1-coverable 8-NAE-CSP instance with  $(\log n)^c$  assignments. It is to be noted that these results pertain to the covering complexity of specific predicates (such as NAE) whereas our results are concerned with classifying which predicates are hard to cover. It would be interesting if [Theorem 3.1.2](#) and [Theorem 3.1.4](#) can be improved to obtain similar hardness results (i.e.,  $\text{poly log } n$  as opposed to  $\text{poly log log } n$ ). The main bottleneck here

seems to be reducing the uniformity parameter (namely, from 8).

## Organization

The rest of the chapter is organized as follows. We start with some preliminaries of LABEL-COVER, covering CSPs and Fourier analysis in § 3.2. Theorems 3.1.1, 3.1.2 and 3.1.4 are proved in Sections 3.3, 3.4 and 3.5 respectively.

## 3.2 Preliminaries

### 3.2.1 Fourier Analysis

We will be dealing with functions of the form  $f : \{0, 1\}^{dL} \rightarrow \mathbb{R}$  for  $d \in \mathbb{N}$  and  $d$ -to-1 functions  $\pi : [dL] \rightarrow [L]$ . We will also think of such functions as  $f : \prod_{i \in L} \Omega_i \rightarrow \mathbb{R}$  where  $\Omega_i = \{0, 1\}^d$  consists of the  $d$  coordinates  $j$  such that  $\pi(j) = i$ . An Efron-Stein decomposition of  $f : \prod_{i \in L} \Omega_i \rightarrow \mathbb{R}$  over the uniform distribution over  $\{0, 1\}^{dL}$ , can be obtained from the Fourier decomposition as

$$f_\beta(x) = \sum_{\alpha \subseteq [dL]: \pi(\alpha) = \beta} \widehat{f}(\alpha) \chi_\alpha. \quad (3.2.1)$$

### 3.2.2 Covering CSPs

We will denote the set  $\{0, 1, \dots, q-1\}$  by  $[q]$ . For  $a \in [q]$ ,  $\bar{a} \in [q]^k$  is the element with  $a$  in all the  $k$  coordinates (where  $k$  and  $q$  will be implicit from the context).

**Definition 3.2.1** (*P-CSP*). *For a predicate  $P \subseteq [q]^k$ , an instance of  $P$ -CSP is given by a (hyper)graph  $G = (V, E)$ , referred to as the constraint graph, and a literals function  $L : E \rightarrow [q]^k$ , where  $V$  is a set of variables and  $E \subseteq V^k$  is a set of constraints. An assignment  $f : V \rightarrow [q]$  is said to cover a constraint  $e = (v_1, \dots, v_k) \in E$ , if  $(f(v_1), \dots, f(v_k)) + L(e) \in P$ , where addition is coordinate-wise modulo  $q$ . A set of assignments  $F = \{f_1, \dots, f_c\}$  is said to cover  $(G, L)$ , if for every  $e \in E$ , there is some  $f_i \in F$  that covers  $e$  and  $F$  is said to be a  $c$ -covering for  $G$ .  $G$  is said to be  $c$ -coverable if there is a  $c$ -covering for  $G$ . If  $L$  is not specified then it is the constant function which maps  $E$  to  $\bar{0}$ .*

**Definition 3.2.2** (COVERING- $P$ -CSP  $(c, s)$ ). For  $P \subseteq [q]^k$  and  $c, s \in \mathbb{N}$ , the COVERING- $P$ -CSP  $(c, s)$  problem is, given a  $c$ -coverable instance  $(G = (V, E), L)$  of  $P$ -CSP, find an  $s$ -covering.

**Definition 3.2.3** (Odd). A predicate  $P \subseteq [q]^k$  is odd if  $\forall x \in [q]^k, \exists a \in [q], x + \bar{a} \in P$ , where addition is coordinate-wise modulo  $q$ .

For odd predicates the covering problem is *trivially solvable*, since any CSP instance on such a predicate is  $q$ -coverable by the  $q$  translates of any assignment, i.e.,  $\{x + \bar{a} \mid a \in [q]\}$  is a  $q$ -covering for any assignment  $x \in [q]^k$ .

### 3.2.3 Label Cover

**Definition 3.2.4** (LABEL-COVER). An instance  $G = (U, V, E, L, R, \{\pi_e\}_{e \in E})$  of the LABEL-COVER constraint satisfaction problem consists of a bi-regular bipartite graph  $(U, V, E)$ , two sets of alphabets  $L$  and  $R$  and a projection map  $\pi_e : R \rightarrow L$  for every edge  $e \in E$ . Given a labeling  $\ell : U \rightarrow L, \ell : V \rightarrow R$ , an edge  $e = (u, v)$  is said to be satisfied by  $\ell$  if  $\pi_e(\ell(v)) = \ell(u)$ .

$G$  is said to be at most  $\delta$ -satisfiable if every labeling satisfies at most a  $\delta$  fraction of the edges.  $G$  is said to be  $c$ -coverable if there exist  $c$  labelings such that for every vertex  $u \in U$ , one of the labelings satisfies all the edges incident on  $u$ .

An instance of UNIQUE-GAMES is a label cover instance where  $L = R$  and the constraints  $\pi$  are permutations.

The hardness of LABEL-COVER stated below follows from the PCP Theorem [AS98, ALM<sup>+</sup>98], Raz's Parallel Repetition Theorem [Raz98] and a structural property proved by Håstad [Hås01, Lemma 6.9].

**Theorem 3.2.5** (Hardness of LABEL-COVER). For every  $r \in \mathbb{N}$ , there is a deterministic  $n^{O(r)}$ -time reduction from a 3-SAT instance of size  $n$  to an instance  $G = (U, V, E, [L], [R], \{\pi_e\}_{e \in E})$  of LABEL-COVER with the following properties:

1.  $|U|, |V| \leq n^{O(r)}$ ;  $L, R \leq 2^{O(r)}$ ;  $G$  is bi-regular with degrees bounded by  $2^{O(r)}$ .

2. There exists a constant  $c_0 \in (0, 1/3)$  such that for any  $v \in V$  and  $\alpha \subseteq [R]$ , for a random neighbor  $u$ ,

$$\mathbf{E}_u [|\pi_{uv}(\alpha)|^{-1}] \leq |\alpha|^{-2c_0}.$$

This implies that

$$\forall v, \alpha, \quad \Pr_u [|\pi_{uv}(\alpha)| < |\alpha|^{c_0}] \leq \frac{1}{|\alpha|^{c_0}}.$$

3. There is a constant  $d_0 \in (0, 1)$  such that,

- YES Case : If the 3-SAT instance is satisfiable, then  $G$  is 1-coverable.
- NO Case : If the 3-SAT instance is unsatisfiable, then  $G$  is at most  $2^{-d_0 r}$ -satisfiable.

Our characterization of hardness of covering CSPs is based on the following conjecture due to Dinur and Kol [DK13].

**Conjecture 3.2.6** (COVERING-UGC( $c$ )). *There exists  $c \in \mathbb{N}$  such that for every sufficiently small  $\delta > 0$  there exists  $L \in \mathbb{N}$  such that the following holds. Given an instance  $G = (U, V, E, [L], [L], \{\pi_e\}_{e \in E})$  of UNIQUE-GAMES it is NP-hard to distinguish between the following two cases:*

- YES case: *There exist  $c$  assignments such that for every vertex  $u \in U$ , at least one of the assignments satisfies all the edges touching  $u$ .*
- NO case: *Every assignment satisfies at most  $\delta$  fraction of the edge constraints.*

### 3.3 UG Hardness of Covering

In this section, we prove the following theorem, which in turn implies Theorem 3.1.1 (see below for proof).

**Theorem 3.3.1.** *Let  $[q]$  be any constant sized alphabet and  $k \geq 2$ . Recall that  $\text{NAE} := [q]^k \setminus \{\bar{b} \mid b \in [q]\}$ . Let  $P \subseteq [q]^k$  be a predicate such that there exists  $a \in \text{NAE}$  and  $\text{NAE} \supset P \supseteq \{a + \bar{b} \mid b \in [q]\}$ . Assuming COVERING-UGC( $c$ ), for every sufficiently small constant  $\delta > 0$  it is NP-hard to distinguish between  $P$ -CSP instances  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  of the following two cases:*

- *YES Case* :  $\mathcal{G}$  is  $2c$ -coverable.
- *NO Case* :  $\mathcal{G}$  does not have an independent set of fractional size  $\delta$ .

*Proof of Theorem 3.1.1.* Let  $Q$  be an arbitrary non odd predicate, i.e,  $Q \subseteq [q]^k \setminus \{h + \bar{b} \mid b \in [q]\}$  for some  $h \in [q]^k$ . Consider the predicate  $Q' \subseteq [q]^k$  defined as  $Q' := Q - h$ . Observe that  $Q' \subseteq \text{NAE}$ . Given any  $Q'$ -CSP instance  $\Phi$  with literals function  $L(e) = \bar{0}$ , consider the  $Q$ -CSP instance  $\Phi_{Q' \rightarrow Q}$  with literals function  $M$  given by  $M(e) := \bar{h}, \forall e$ . It has the same constraint graph as  $\Phi$ . Clearly,  $\Phi$  is  $c$ -coverable iff  $\Phi_{Q' \rightarrow Q}$  is  $c$ -coverable. Thus, it suffices to prove the result for any predicate  $Q' \subseteq \text{NAE}$  with literals function  $L(e) = \bar{0}^3$ . We will consider two cases, both of which will follow from Theorem 3.3.1.

Suppose the predicate  $Q'$  satisfies  $Q' \supseteq \{a + \bar{b} \mid b \in [q]\}$  for some  $a \in [q]^k$ . Then this predicate  $Q'$  satisfies the hypothesis of Theorem 3.3.1 and the theorem follows if we show that the soundness guarantee of Theorem 3.3.1 implies that in Theorem 3.1.1. Any instance in the NO case of Theorem 3.3.1, is not  $t := \log_q(1/\delta)$ -coverable even on the NAE-CSP instance with the same constraint graph. This is because any  $t$ -covering for the NAE-CSP instance gives a coloring of the constraint graph using  $q^t$  colors, by choosing the color of every variable to be a string of length  $t$  and having the corresponding assignments in each position in  $[t]$ . Hence the  $Q'$ -CSP instance is also not  $t$ -coverable.

Suppose  $Q' \not\supseteq \{a + \bar{b} \mid b \in [q]\}$  for all  $a \in [q]^k$ . Then consider the predicate  $P = \{a + \bar{b} \mid a \in Q', b \in [q]\} \subseteq \text{NAE}$ . Notice that  $P$  satisfies the conditions of Theorem 3.3.1 and if the  $P$ -CSP instance is  $t$ -coverable then the  $Q'$ -CSP instance is  $qt$ -coverable. Hence an YES instance of Theorem 3.3.1 maps to a  $2cq$ -coverable  $Q$ -CSP instance and NO instance maps to an instance with covering number at least  $\log_q(1/\delta)$ . ■

---

<sup>3</sup>This observation [DK13] that the cover- $Q$  problem for any non-odd predicate  $Q$  is equivalent to the cover- $Q'$  problem where  $Q' \subseteq \text{NAE}$  shows the centrality of the NAE predicate in understanding the covering complexity of any non-odd predicate.

We now prove [Theorem 3.3.1](#) by giving a reduction from an instance  $G = (U, V, E, [L], [L], \{\pi_e\}_{e \in E})$  of UNIQUE-GAMES as in [Definition 3.2.4](#), to an instance  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  of a  $P$ -CSP for any predicate  $P$  that satisfies the conditions mentioned. As stated in the introduction, we adapt the long-code test of Bansal and Khot [\[BK10\]](#) for proving the hardness of finding independent sets in almost  $k$ -partite  $k$ -uniform hypergraphs to our setting. The set of variables  $\mathcal{V}$  is  $V \times [q]^{2L}$ . Any assignment to  $\mathcal{V}$  is given by a set of functions  $f_v : [q]^{2L} \rightarrow [q]$ , for each  $v \in V$ . The set of constraints  $\mathcal{E}$  is given by the following test which checks whether  $f_v$ 's are long codes of a good labeling to  $V$ . There is a constraint corresponding to all the variables that are queried together by the test.

### Long Code Test $\mathcal{T}_1$

1. Choose  $u \in U$  uniformly and  $k$  neighbors  $w_1, \dots, w_k \in V$  of  $u$  uniformly and independently at random.
2. Choose a random matrix  $X$  of dimension  $k \times 2L$  as follows. Let  $X^i$  denote the  $i^{\text{th}}$  column of  $X$ . Independently for each  $i \in [L]$ , choose  $(X^i, X^{i+L})$  uniformly at random from the set

$$S := \left\{ (y, y') \in [q]^k \times [q]^k \mid y \in \{a + \bar{b} \mid b \in [q]\} \vee y' \in \{a + \bar{b} \mid b \in [q]\} \right\}. \quad (3.3.1)$$

3. Let  $x_1, \dots, x_k$  be the rows of matrix  $X$ . Accept iff

$$(f_{w_1}(x_1 \circ \pi_{uw_1}), f_{w_2}(x_2 \circ \pi_{uw_2}), \dots, f_{w_k}(x_k \circ \pi_{uw_k})) \in P,$$

where  $x \circ \pi$  is the string defined as  $(x \circ \pi)(i) := x_{\pi(i)}$  for  $i \in [L]$  and  $(x \circ \pi)(i) := x_{\pi(i-L)+L}$  otherwise.

**Lemma 3.3.2** (Completeness). *If the UNIQUE-GAMES instance  $G$  is  $c$ -coverable then the  $P$ -CSP instance  $\mathcal{G}$  is  $2c$ -coverable.*

*Proof.* Let  $\ell_1, \dots, \ell_c : U \cup V \rightarrow [L]$  be a  $c$ -covering for  $G$  as described in [Definition 3.2.4](#). We will show that the  $2c$  assignments given by  $f_v^i(x) := x_{\ell_i(v)}$ ,  $g_v^i(x) := x_{\ell_i(v)+L}$ ,  $i = 1, \dots, c$  form a  $2c$ -covering of  $\mathcal{G}$ . Consider any  $u \in U$  and let  $\ell_i$  be the labeling that covers all the edges incident on  $u$ . For any  $(u, w_j)_{j \in \{1, \dots, k\}} \in E$  and  $X$  chosen by the

long code test  $\mathcal{T}_1$ , the vector  $(f_{w_1}^i(x_1 \circ \pi_{uw_1}), \dots, f_{w_k}^i(x_k \circ \pi_{uw_k}))$  gives the  $\ell_i(u)$ th column of  $X$ . Similarly the above expression corresponding to  $g^i$  gives the  $(\ell_i(u) + L)$ th column of the matrix  $X$ . Since, for all  $i \in [L]$ , either  $i$ th column or  $(i + L)$ th column of  $X$  contains element from  $\{a + \bar{b} \mid b \in [q]\} \subseteq P$ , either  $(f_{w_1}^i(x_1 \circ \pi_{uw_1}), \dots, f_{w_k}^i(x_k \circ \pi_{uw_k})) \in P$  or  $(g_{w_1}^i(x_1 \circ \pi_{uw_1}), \dots, g_{w_k}^i(x_k \circ \pi_{uw_k})) \in P$ . Hence the set of  $2c$  assignments  $\{f_v^i, g_v^i\}_{i \in \{1, \dots, c\}}$  covers all constraints in  $\mathcal{G}$ .  $\blacksquare$

To prove soundness, we show that the set  $S$ , as defined in Equation (3.3.1), is connected, so that Theorem 2.4.1 is applicable. For this, we view  $S \subseteq [q]^k \times [q]^k$  as a subset of  $([q]^2)^k$  as follows: the element  $(y, y') \in S$  is mapped to the element  $((y_1, y'_1), \dots, (y_k, y'_k)) \in ([q]^2)^k$ .

**Claim 3.3.3.** *Let  $\Omega = [q]^2$ . The set  $S \subset \Omega^k$  is connected.*

*Proof.* Consider any  $x := (x^1, x^2), y := (y^1, y^2) \in S \subset [q]^k \times [q]^k$ . Suppose both  $x^1, y^1 \in \{a + \bar{b} \mid b \in [q]\}$ , then it is easy to come up with a sequence of strings belonging to  $S$ , starting with  $x$  and ending with  $y$  such that consecutive strings differ in at most 1 coordinate. Now suppose  $x^1, y^2 \in \{a + \bar{b} \mid b \in [q]\}$ . First we come up with a sequence from  $x$  to  $z := (z^1, z^2)$  such that  $z^1 := x^1$  and  $z^2 = y^2$ , and then another sequence for  $z$  to  $y$ .  $\blacksquare$

**Lemma 3.3.4** (Soundness). *For every constant  $\delta > 0$ , there exists a constant  $s$  such that, if  $G$  is at most  $s$ -satisfiable then  $\mathcal{G}$  does not have an independent set of size  $\delta$ .*

*Proof.* Let  $I \subseteq \mathcal{V}$  be an independent set of fractional size  $\delta$  in the constraint graph. For every variable  $v \in V$ , let  $f_v : [q]^{2L} \rightarrow \{0, 1\}$  be the indicator function of the independent set restricted to the vertices that correspond to  $v$ . For a vertex  $u \in U$ , let  $N(u) \subseteq V$  be the set of neighbors of  $u$  and define  $f_u(x) := \mathbf{E}_{w \in N(u)} [f_w(x \circ \pi_{uw})]$ . Since  $I$  is an independent set, we have

$$0 = \mathbf{E}_{u, w_1, \dots, w_k} \mathbf{E}_{X \sim \mathcal{T}_1} \left[ \prod_{i=1}^k f_{w_i}(x_i \circ \pi_{uw_i}) \right] = \mathbf{E}_u \mathbf{E}_{X \sim \mathcal{T}_1} \left[ \prod_{i=1}^k f_u(x_i) \right]. \quad (3.3.2)$$

Since the bipartite graph  $(U, V, E)$  is left regular and  $|I| \geq \delta|V|$ , we have  $\mathbf{E}_{u,x} [f_u(x)] \geq \delta$ . By an averaging argument, for at least  $\frac{\delta}{2}$  fraction of the vertices  $u \in U$ ,  $\mathbf{E}_x [f_u(x)] \geq \frac{\delta}{2}$ .

Call a vertex  $u \in U$  *good* if it satisfies this property. A string  $x \in [q]^{2L}$  can be thought as an element from  $([q]^2)^L$  by grouping the pair of coordinates  $x_i, x_{i+L}$ . Let  $\bar{x} \in ([q]^2)^L$  denotes this grouping of  $x$ , i.e.,  $j$ th coordinate of  $\bar{x}$  is  $(x_j, x_{j+L}) \in [q]^2$ . With this grouping, the function  $f_u$  can be viewed as  $f_u : ([q]^2)^L \rightarrow \{0, 1\}$ . From Equation (3.3.2), we have that for any  $u \in U$ ,

$$\mathbf{E}_{X \sim \mathcal{T}_1} \left[ \prod_{i=1}^k f_u(\bar{x}_i) \right] = 0.$$

By Claim 3.3.3, for all  $j \in [L]$  the tuple  $((\bar{x}_1)_j, \dots, (\bar{x}_k)_j)$  (corresponding to columns  $(X^j, X^{j+L})$  of  $X$ ) is sampled from a distribution whose support is a connected set. Hence for a good vertex  $u \in U$ , we can apply Theorem 2.4.1 with  $\varepsilon = \underline{\Gamma}(\delta/2)/2$  to get that there exists  $j \in [L], d \in \mathbb{N}, \tau > 0$  such that  $\text{Inf}_j^{\leq d}(f_u) > \tau$ . We will use this fact to give a randomized labeling for  $G$ . Labels for vertices  $w \in V, u \in U$  will be chosen uniformly and independently from the sets

$$\text{Lab}(w) := \left\{ i \in [L] \mid \text{Inf}_i^{\leq d}(f_w) \geq \frac{\tau}{2} \right\}, \text{Lab}(u) := \left\{ i \in [L] \mid \text{Inf}_i^{\leq d}(f_u) \geq \tau \right\}.$$

By the above argument (using Theorem 2.4.1), we have that for a good vertex  $u$ ,  $\text{Lab}(u) \neq \emptyset$ . Furthermore, since the sum of degree  $d$  influences is at most  $d$ , the above sets have size at most  $2d/\tau$ . Now, for any  $j \in \text{Lab}(u)$ , we have

$$\begin{aligned} \tau < \text{Inf}_j^{\leq d}[f_u] &= \sum_{S: j \in S, |S| \leq d} \|f_{u,S}\|^2 = \sum_{S: j \in S, |S| \leq d} \left\| \mathbf{E}_{w \in N(u)} [f_{w, \pi_{uw}^{-1}(S)}] \right\|^2 \quad (\text{By Definition.}) \\ &\leq \sum_{S: j \in S, |S| \leq d} \mathbf{E}_{w \in N(u)} \|f_{w, \pi_{uw}^{-1}(S)}\|^2 = \mathbf{E}_{w \in N(u)} \text{Inf}_{\pi_{uw}^{-1}(j)}^{\leq d}[f_w]. \quad (\text{By Convexity of square.}) \end{aligned}$$

Hence, by another averaging argument, there exists at least  $\frac{\tau}{2}$  fraction of neighbors  $w$  of  $u$  such that  $\text{Inf}_{\pi_{uw}^{-1}(j)}^{\leq d}(f_w) \geq \frac{\tau}{2}$  and hence  $\pi_{uw}^{-1}(j) \in \text{Lab}(w)$ . Therefore, for a good vertex  $u \in U$ , at least  $\frac{\tau}{2} \frac{\tau}{2d}$  fraction of edges incident on  $u$  are satisfied in expectation. Also, at least  $\frac{\delta}{2}$  fraction of vertices in  $U$  are good, it follows that the expected fraction of edges that are satisfied by this random labeling is at least  $\frac{\delta}{2} \frac{\tau}{2} \frac{\tau}{2d}$ . Choosing  $s < \frac{\delta}{2} \frac{\tau}{2} \frac{\tau}{2d}$  completes the proof.  $\blacksquare$

### 3.4 NP-Hardness of Covering

In this section, we prove [Theorem 3.1.2](#). We give a reduction from an instance of a LABEL-COVER,  $G = (U, V, E, [L], [R], \{\pi_e\}_{e \in E})$  as in [Definition 3.2.4](#), to a  $P$ -CSP instance  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  for any predicate  $P$  that satisfies the conditions mentioned in [Theorem 3.1.2](#). The reduction and proof is similar to that of Dinur and Kol [[DK13](#)]. The main difference is that they used a test and invariance principle very specific to the 4-LIN predicate, while we show that a similar analysis can be performed under milder conditions on the test distribution.

We assume that  $R = dL$  and  $\forall i \in [L], e \in E, |\pi_e^{-1}(i)| = d$ . This is done just for simplifying the notation and the proof does not depend upon it. The set of variables  $\mathcal{V}$  is  $V \times \{0, 1\}^{2R}$ . Any assignment to  $\mathcal{V}$  is given by a set of functions  $f_v : \{0, 1\}^{2R} \rightarrow \{0, 1\}$ , for each  $v \in V$ . The set of constraints  $\mathcal{E}$  is given by the following test which checks whether  $f_v$ 's are long codes of a good labeling to  $V$ .

#### Long Code Test $\mathcal{T}_2$

1. Choose  $u \in U$  uniformly and  $v, w \in V$  neighbors of  $u$  uniformly and independently at random. For  $i \in [L]$ , let  $B_{uv}(i) := \pi_{uv}^{-1}(i)$ ,  $B'_{uv}(i) := R + \pi_{uv}^{-1}(i)$  and similarly for  $w$ .
2. Choose matrices  $X, Y$  of dimension  $k \times 2dL$  as follows. For  $S \subseteq [2dL]$ , we denote by  $X|_S$  the submatrix of  $X$  restricted to the columns  $S$ . Independently for each  $i \in [L]$ , choose  $c_1 \in \{0, 1\}$  uniformly and

- (a) if  $c_1 = 0$ , choose  $(X|_{B_{uv}(i) \cup B'_{uv}(i)}, Y|_{B_{uw}(i) \cup B'_{uw}(i)})$  from  $\mathcal{P}_0^{\otimes 2d} \otimes \mathcal{P}_1^{\otimes 2d}$ ,
- (b) if  $c_1 = 1$ , choose  $(X|_{B_{uv}(i) \cup B'_{uv}(i)}, Y|_{B_{uw}(i) \cup B'_{uw}(i)})$  from  $\mathcal{P}_1^{\otimes 2d} \otimes \mathcal{P}_0^{\otimes 2d}$ .

3. Perturb  $X, Y$  as follows. Independently for each  $i \in [L]$ , choose  $c_2 \in \{*, 0, 1\}$  as follows:  $\Pr[c_2 = *] = 1 - 2\varepsilon$ , and  $\Pr[c_2 = 1] = \Pr[c_2 = 0] = \varepsilon$ . Perturb the  $i$ th matrix block  $(X|_{B_{uv}(i) \cup B'_{uv}(i)}, Y|_{B_{uw}(i) \cup B'_{uw}(i)})$  as follows:

- (a) if  $c_2 = *$ , leave the matrix block  $(X|_{B_{uv}(i) \cup B'_{uv}(i)}, Y|_{B_{uw}(i) \cup B'_{uw}(i)})$  unperturbed,

- (b) if  $c_2 = 0$ , choose  $(X|_{B'_{uv}(i)}, Y|_{B'_{uw}(i)})$  uniformly from  $\{0, 1\}^{k \times d} \times \{0, 1\}^{k \times d}$ ,
  - (c) if  $c_2 = 1$ , choose  $(X|_{B_{uv}(i)}, Y|_{B_{uw}(i)})$  uniformly from  $\{0, 1\}^{k \times d} \times \{0, 1\}^{k \times d}$ .
4. Let  $x_1, \dots, x_k$  and  $y_1, \dots, y_k$  be the rows of the matrices  $X$  and  $Y$  respectively.

Accept if

$$(f_v(x_1), \dots, f_v(x_k), f_w(y_1), \dots, f_w(y_k)) \in P.$$

**Lemma 3.4.1** (Completeness). *If  $G$  is an YES instance of LABEL-COVER, then there exists  $f, g$  such that each of them covers  $1 - \varepsilon$  fraction of  $\mathcal{E}$  and they together cover all of  $\mathcal{E}$ .*

*Proof.* Let  $\ell : U \cup V \rightarrow [L] \cup [R]$  be a labeling to  $G$  that satisfies all the constraints. Consider the assignments  $f_v(x) := x_{\ell(v)}$  and  $g_v(x) := x_{R+\ell(v)}$  for each  $v \in V$ . First consider the assignment  $f$ . For any  $(u, v), (u, w) \in E$  and  $x_1, \dots, x_k, y_1, \dots, y_k$  chosen by the long code test  $\mathcal{T}_2$ ,  $(f_v(x_1), \dots, f_v(x_k)), (f_w(y_1), \dots, f_w(y_k))$  gives the  $\ell(v)$ th and  $\ell(w)$ th column of the matrices  $X$  and  $Y$  respectively. Since  $\pi_{uv}(\ell(v)) = \pi_{uw}(\ell(w))$ , they are jointly distributed either according to  $\mathcal{P}_0 \otimes \mathcal{P}_1$  or  $\mathcal{P}_1 \otimes \mathcal{P}_0$  after Step 2. The probability that these rows are perturbed in Step 3c is at most  $\varepsilon$ . Hence with probability  $1 - \varepsilon$  over the test distribution,  $f$  is accepted. A similar argument shows that the test accepts  $g$  with probability  $1 - \varepsilon$ . Note that in Step 3, the columns given by  $f, g$ , are never re-sampled uniformly together. Hence they together cover  $\mathcal{G}$ . ■

Now we will show that if  $G$  is a NO instance of LABEL-COVER then no  $t$  assignments can cover the 2k-LIN-CSP with constraint hypergraph  $\mathcal{G}$ . For the rest of the analysis, we will use  $+1, -1$  instead of the symbols  $0, 1$ . Suppose for contradiction, there exist  $t$  assignments  $f_1, \dots, f_t : \{\pm 1\}^{2R} \rightarrow \{\pm 1\}$  that form a  $t$ -cover to  $\mathcal{G}$ . The probability that all the  $t$  assignments are rejected in Step 4 is

$$\mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^t \frac{1}{2} \left( \prod_{j=1}^k f_{i,v}(x_j) f_{i,w}(y_j) + 1 \right) \right] = \frac{1}{2^t} + \frac{1}{2^t} \sum_{\emptyset \subset S \subseteq \{1, \dots, t\}} \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{j=1}^k f_{S,v}(x_j) f_{S,w}(y_j) \right]. \quad (3.4.1)$$

where  $f_{S,v}(x) := \prod_{i \in S} f_{i,v}(x)$ . Since the  $t$  assignments form a  $t$ -cover, the LHS in Equation (3.4.1) is 0 and hence, there exists an  $S \neq \emptyset$  such that

$$\mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{j=1}^k f_{S,v}(x_j) f_{S,w}(y_j) \right] \leq -1/(2^t - 1). \quad (3.4.2)$$

Lemma 3.4.3 shows that this is not possible if  $t$  is not too large, thus proving that there does not exist a  $t$ -cover.

We will need the following technical claim. We denote the  $k \times 2d$  dimensional matrix  $X|_{B(i) \cup B'(i)}$  ( $X|_{B_{uv}(i) \cup B'_{uv}(i)}$ ) by  $X^i$  and  $Y|_{B(i) \cup B'(i)}$  ( $Y|_{B_{uv}(i) \cup B'_{uv}(i)}$ ) by  $Y^i$  (ignoring the subscript of  $B$  and  $B'$  as the distribution is the same for every edge  $(u, v)$ ). Also by  $X_j^i$ , we mean the  $j$ th row of the matrix  $X^i$  and  $Y_{-k}^i$  is the first  $k-1$  rows of  $Y^i$ . The spaces of the random variables  $X^i, X_j^i, Y_{-k}^i$  will be denoted by  $\mathcal{X}^i, \mathcal{X}_j^i, \mathcal{Y}_{-k}^i$ .

**Claim 3.4.2.** *For each  $i \in [L]$ ,*

$$\rho(\mathcal{X}^i \times \mathcal{Y}_{-k}^i, \mathcal{Y}_k^i; \mathcal{T}_2^i) \leq \sqrt{1 - \varepsilon}.$$

*Proof.* Recall the random variable  $c_2 \in \{*, 0, 1\}$  defined in Step 3 of test  $\mathcal{T}_2$ . Let  $g$  and  $f$  be the functions that satisfies  $\mathbf{E}[g] = \mathbf{E}[f] = 0$  and  $\mathbf{E}[g^2], \mathbf{E}[f^2] \leq 1$  such that  $\rho(\mathcal{X}^i \times \mathcal{Y}_{-k}^i, \mathcal{Y}_k^i; \mathcal{T}_2^i) = \mathbf{E}[|fg|]$ . Define the *Markov Operator*

$$Ug(X^i, Y_{-k}^i) = \mathbf{E}_{(\tilde{X}, \tilde{Y}) \sim \mathcal{T}_2^i} [g(\tilde{Y}_k) \mid (\tilde{X}, \tilde{Y}_{-k}) = (X^i, Y_{-k}^i)].$$

By Lemma 2.2.4, we have

$$\begin{aligned} \rho(\mathcal{X}^i \times \mathcal{Y}_{-k}^i, \mathcal{Y}_k^i; \mathcal{T}_2^i)^2 &\leq \mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2] \\ &= (1 - 2\varepsilon) \mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2 \mid c_2 = *] + \varepsilon \mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2 \mid c_2 = 0] + \\ &\quad \varepsilon \mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2 \mid c_2 = 1] \\ &\leq (1 - 2\varepsilon) + \varepsilon \mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2 \mid c_2 = 0] + \varepsilon \mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2 \mid c_2 = 1], \end{aligned}$$

where the last inequality uses the fact that  $\mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2 \mid c_2 = *] = \mathbf{E}[g^2]$  which is at most 1. Consider the case when  $c_2 = 0$ . By definition, we have

$$\mathbf{E}_{\mathcal{T}_2^i} [Ug(X^i, Y_{-k}^i)^2 \mid c_2 = 0] = \mathbf{E}_{\left(\begin{smallmatrix} X^i \\ Y_{-k}^i \end{smallmatrix}\right) \sim \mathcal{T}_2^i} \left( \mathbf{E}_{(\tilde{X}, \tilde{Y}) \sim \mathcal{T}_2^i} [g(\tilde{Y}_k) \mid (\tilde{X}, \tilde{Y}_{-k}) = (X^i, Y_{-k}^i) \wedge c_2 = 0] \right)^2.$$

Under the conditioning, for any fixed value of  $X^i, Y_{-k}^i$ , the value of  $\tilde{Y}_k|_{B'(i)}$  is a uniformly random string whereas  $\tilde{Y}_k|_{B(i)}$  is a fixed string (since the *parity* of all columns in  $B(i)$

is 1). Let  $\mathcal{U}$  be the uniform distribution on  $\{-1, +1\}^d$  and  $\mathcal{P}(X^i, Y_{-k}^i) \in \{+1, -1\}^d$  denotes the column wise parities of  $\begin{bmatrix} X^i|_{B(i)} \\ Y_{-k}^i|_{B(i)} \end{bmatrix}$ .

$$\begin{aligned}
\mathbf{E}_{\mathcal{T}_2^i}[Ug(X^i, Y_{-k}^i)^2 \mid c_2 = 0] &= \mathbf{E}_{X^i, Y_{-k}^i \sim \mathcal{T}_2^i} \left( \mathbf{E}_{(\tilde{X}, \tilde{Y}) \sim \mathcal{T}_2^i} \left[ g(\tilde{Y}_k) \mid \substack{(\tilde{X}, \tilde{Y}_{-k}) = (X^i, Y_{-k}^i) \wedge \\ c_2 = 0} \right] \right)^2 \\
&= \mathbf{E}_{\substack{X^i, Y_{-k}^i \sim \mathcal{T}_2^i, \\ z = \mathcal{P}(X^i, Y_{-k}^i)}} \left( \mathbf{E}_{r \sim \mathcal{U}} [g(-z, r)] \right)^2 \\
&= \mathbf{E}_{z \sim \mathcal{U}} \left( \mathbf{E}_{r \sim \mathcal{U}} [g(z, r)] \right)^2 \quad (\text{Since marginal on } z \text{ is uniform}) \\
&= \mathbf{E}_{z \sim \mathcal{U}} \left( \mathbf{E}_{r \in \mathcal{U}} \sum_{\alpha \subseteq B(i) \cup B'(i)} \hat{g}(\alpha) \chi_\alpha(z, r) \right)^2 \\
&= \mathbf{E}_{z \sim \mathcal{U}} \left( \sum_{\alpha \subseteq B(i) \cup B'(i)} \hat{g}(\alpha) \mathbf{E}_{r \in \mathcal{U}} [\chi_\alpha(z, r)] \right)^2 \\
&= \mathbf{E}_{z \sim \mathcal{U}} \left( \sum_{\alpha \subseteq B(i)} \hat{g}(\alpha) \chi_\alpha(z) \right)^2 \\
&= \sum_{\alpha \subseteq B(i)} \hat{g}(\alpha)^2.
\end{aligned}$$

Similarly we have,

$$\mathbf{E}_{\mathcal{T}_2^i}[Ug(X^i, Y_{-k}^i)^2 \mid c_2 = 1] = \sum_{\alpha \subseteq B'(i)} \hat{g}(\alpha)^2.$$

Now we can bound the correlation as follows:

$$\begin{aligned}
\rho(\mathcal{X}^i \times \mathcal{Y}_{-k}^i, \mathcal{Y}_k^i, \mathcal{T}_2^i)^2 &\leq (1 - 2\varepsilon) + \varepsilon \sum_{\alpha \subseteq B(i)} \hat{g}(\alpha)^2 + \varepsilon \sum_{\alpha \subseteq B'(i)} \hat{g}(\alpha)^2 \\
&\leq (1 - 2\varepsilon) + \varepsilon \sum_{\alpha \subseteq B(i) \cup B'(i)} \hat{g}(\alpha)^2 \quad (\text{Using } \hat{g}(\phi) = \mathbf{E}[g] = 0) \\
&\leq (1 - \varepsilon). \quad (\text{Using } \mathbf{E}[g^2] \leq 1 \text{ and Parseval's Identity})
\end{aligned}$$

■

**Lemma 3.4.3** (Soundness). *Let  $c_0 \in (0, 1)$  be the constant from [Theorem 3.2.5](#) and  $S \subseteq \{1, \dots, t\}, |S| > 0$ . If  $G$  is at most  $s$ -satisfiable then*

$$\mathbf{E}_{u,v,w} \mathbf{E}_{X,Y \in \mathcal{T}_2} \left[ \prod_{i=1}^k f_{S,v}(x_i) f_{S,w}(y_i) \right] \geq -O(ks^{c_0/8}) - 2^{O(k)} \frac{s^{(1-3c_0)/8}}{\varepsilon^{3/2c_0}}.$$

*Proof.* Notice that for a fixed  $u$ , the distribution of  $X$  and  $Y$  have identical marginals. Hence the value of the above expectation, if calculated according to a distribution which is the direct product of the marginals, is positive. We will first show that the expectation can change by at most  $O(ks^{c_0/8})$  in moving to an *attenuated* version of the functions (see [Claim 3.4.4](#)). Then we will show that the error incurred by changing the distribution to the product distribution of the marginals has absolute value at most  $2^{O(k)} \frac{s^{(1-3c_0)/8}}{\varepsilon^{3/2c_0}}$  (see [Claim 3.4.6](#)). This is done by showing that there is a labeling to  $G$  that satisfies an  $s$  fraction of the constraints if the error is more than  $2^{O(k)} \frac{s^{(1-3c_0)/8}}{\varepsilon^{3/2c_0}}$ .

For the rest of the analysis, we write  $f_v$  and  $f_w$  instead of  $f_{S,v}$  and  $f_{S,w}$  respectively. Let  $f_v = \sum_{\alpha \subseteq [2R]} \hat{f}_v(\alpha) \chi_\alpha$  be the Fourier decomposition of the function and for  $\gamma \in (0, 1)$ , let  $T_{1-\gamma} f_v := \sum_{\alpha \subseteq [2R]} (1-\gamma)^{|\alpha|} \hat{f}_v(\alpha) \chi_\alpha$ . The following claim is similar to a lemma of Dinur and Kol [[DK13](#), Lemma 4.11]. The only difference in the proof is that, we use the *smoothness* from Property 2 of Theorem [3.2.5](#) (which was shown by Håstad [[Hås01](#), Lemma 6.9]).

**Claim 3.4.4.** Let  $\gamma := s^{(c_0+1)/4} \varepsilon^{1/c_0}$  where  $c_0$  is the constant from [Theorem 3.2.5](#).

$$\left| \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k f_v(x_i) f_w(y_i) \right] - \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma} f_v(x_i) T_{1-\gamma} f_w(y_i) \right] \right| \leq O(ks^{c_0/8}).$$

*Proof.* We will add the  $T_{1-\gamma}$  operator to one function at a time and upper bound the absolute value of the error incurred each time by  $O(s^{c_0/8})$ . The total error is at most  $2k$  times the error in adding  $T_{1-\gamma}$  to one function. Hence, it suffices to prove the following

$$\left| \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k f_v(x_i) f_w(y_i) \right] - \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} \left[ \left( \prod_{i=1}^{k-1} f_v(x_i) f_w(y_i) \right) f_v(x_k) T_{1-\gamma} f_w(y_k) \right] \right| \leq O(s^{c_0/8}). \quad (3.4.3)$$

Recall that  $X, Y$  denote the matrices chosen by test  $\mathcal{T}_2$ . Let  $Y_{-k}$  be the matrix obtained from  $Y$  by removing the  $k$ th row and  $F_{u,v,w}(X, Y_{-k}) := \left( \prod_{i=1}^{k-1} f_v(x_i) f_w(y_i) \right) f_v(x_k)$ . Then, (3.4.3) can be rewritten as

$$\left| \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} [F_{u,v,w}(X, Y_{-k}) (I - T_{1-\gamma}) f_w(y_k)] \right| \leq O(s^{c_0/8}). \quad (3.4.4)$$

Let  $U$  be the operator that maps functions on the variable  $y_k$ , to one on the variables  $(X, Y_{-k})$  defined by

$$(Uf)(X, Y_{-k}) := \mathbf{E}_{y_k | X, Y_{-k}} f(y_k).$$

Let  $G_{u,v,w}(X, Y_{-k}) := (U(I - T_{1-\gamma})f_w)(X, Y_{-k})$ . Note that  $\mathbf{E}_{y \in \{0,1\}^{2R}} G_{u,v,w}(y) = 0$ . For the rest of the analysis, fix  $u, v, w$  chosen by the test. We will omit the subscript  $u, v, w$  from now on for notational convenience. The domain of  $G$  can be thought of as  $(\{0,1\}^{2k-1})^{2dL}$  and the test distribution on any row is independent across the blocks  $\{B_{uv}(i) \cup B'_{uv}(i)\}_{i \in [L]}$ . We now think of  $G$  as having domain  $\prod_{i \in [L]} \Omega_i$  where  $\Omega_i = (\{0,1\}^{2k-1})^{2d}$  corresponds to the set of rows in  $B_{uv}(i) \cup B'_{uv}(i)$ . Let the following be the Efron-Stein decomposition of  $G$  with respect to  $\mathcal{T}_2$ ,

$$G(X, Y_{-k}) = \sum_{\alpha \subseteq [L]} G_\alpha(X, Y_{-k}).$$

The following technical claim follows from a result similar to [DK13, Lemma 4.7] and then using [Mos10, Proposition 2.12].

**Claim 3.4.5.** *For  $\alpha \subseteq [L]$*

$$\|G_\alpha\|^2 \leq (1 - \varepsilon)^{|\alpha|} \sum_{\beta \subseteq [2R]: \tilde{\pi}_{uw}(\beta) = \alpha} \left(1 - (1 - \gamma)^{2|\beta|}\right) \hat{f}_w(\beta)^2 \quad (3.4.5)$$

where  $\tilde{\pi}_{uw}(\beta) := \{i \in [L] : \exists j \in [R], (j \in \beta \vee j + R \in \beta) \wedge \pi_{uv}(j) = i\}$ .

*Proof.* Proposition 2.2.5 shows that the Markov operator  $U$  commutes with taking the Efron-Stein decomposition. Hence,  $G_\alpha := (U((I - T_{1-\gamma})f_w))_\alpha = U((I - T_{1-\gamma})(f_w)_\alpha)$ , where  $(f_w)_\alpha$  is the Efron-Stein decomposition of  $f_w$  w.r.t the marginal distribution of  $\mathcal{T}_2$  on  $\prod_{i=1}^L \mathcal{Y}_k^i$  which is a uniform distribution. Therefore,  $(f_w)_\alpha = \sum_{\substack{\beta \subseteq [2R], \\ \tilde{\pi}_{uw}(\beta) = \alpha}} \hat{f}_w(\beta) \chi_\beta$ . Using Proposition 2.2.6 and Claim 3.4.2, we have

$$\begin{aligned} \|G_\alpha\|_2^2 &= \|U((I - T_{1-\gamma})(f_w)_\alpha)\|_2^2 \leq (\sqrt{1 - \varepsilon})^{2|\alpha|} \|(I - T_{1-\gamma})(f_w)_\alpha\|_2^2 \\ &= (1 - \varepsilon)^{|\alpha|} \sum_{\beta \subseteq [2R]: \tilde{\pi}_{uw}(\beta) = \alpha} \left(1 - (1 - \gamma)^{2|\beta|}\right) \hat{f}_w(\beta)^2, \end{aligned}$$

where the norms are with respect to the marginals of  $\mathcal{T}_2$  in the corresponding spaces. ■

Substituting the Efron-Stein decomposition of  $G, F$  into the LHS of (3.4.4) gives

$$\begin{aligned}
\left| \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} [F_{u,v,w}(X, Y_{-k}) (I - T_{1-\gamma}) f_w(y_k)] \right| &= \left| \mathbf{E}_{u,v,w} \mathbf{E}_{\mathcal{T}_2} F(X, Y_{-k}) G(X, Y_{-k}) \right| \\
&\stackrel{\text{(By orthonormality of Efron-Stein decomposition)}}{=} \left| \mathbf{E}_{u,v,w} \sum_{\alpha \subseteq [L]} \mathbf{E}_{\mathcal{T}_2} F_\alpha(X, Y_{-k}) G_\alpha(X, Y_{-k}) \right| \\
&\stackrel{\text{(By Cauchy-Schwarz inequality)}}{\leq} \mathbf{E}_{u,v,w} \sqrt{\sum_{\alpha \subseteq [L]} \|F_\alpha\|^2} \cdot \sqrt{\sum_{\alpha \subseteq [L]} \|G_\alpha\|^2} \\
&\stackrel{\text{(Using } \sum_{\alpha \subseteq [L]} \|F_\alpha\|^2 = \|F\|_2^2 = 1)}{\leq} \mathbf{E}_{u,v,w} \sqrt{\sum_{\alpha \subseteq [L]} \|G_\alpha\|^2}.
\end{aligned}$$

Using concavity of square root and substituting for  $\|G_\alpha\|^2$  from Equation (3.4.5), we get that the above is upper bounded by

$$\sqrt{\sum_{\alpha \subseteq [L]} \sum_{\substack{\beta \subseteq [2R]: \\ \tilde{\pi}_{u,w}(\beta) = \alpha}} \underbrace{\mathbf{E}_{u,v,w} (1-\varepsilon)^{|\alpha|} \left(1 - (1-\gamma)^{2|\beta|}\right) \hat{f}_w(\beta)^2}_{=: \text{Term}_{u,w}(\alpha, \beta)}}.$$

We will now break the above summation into three different parts and bound each part separately.

$$\begin{aligned}
\Theta_0 &:= \mathbf{E}_{u,w} \sum_{\alpha, \beta: |\alpha| \geq \frac{1}{\varepsilon s^{c_0/4}}} \text{Term}_{u,w}(\alpha, \beta), & \Theta_1 &:= \mathbf{E}_{u,w} \sum_{\substack{\alpha, \beta: |\alpha| < \frac{1}{\varepsilon s^{c_0/4}} \\ |\beta| \leq \frac{2}{s^{1/4} \varepsilon^{1/c_0}}}} \text{Term}_{u,w}(\alpha, \beta), \\
\Theta_2 &:= \mathbf{E}_{u,w} \sum_{\substack{\alpha, \beta: |\alpha| < \frac{1}{\varepsilon s^{c_0/4}} \\ |\beta| > \frac{2}{s^{1/4} \varepsilon^{1/c_0}}}} \text{Term}_{u,w}(\alpha, \beta).
\end{aligned}$$

**Upper bounding  $\Theta_0$ :** When  $|\alpha| > \frac{1}{\varepsilon s^{c_0/4}}$ ,  $(1-\varepsilon)^{|\alpha|} < s^{c_0/4}$ . Also since  $f_w$  is  $\{+1, -1\}$  valued, sum of squares of Fourier coefficient is 1. Hence  $|\Theta_0| < s^{c_0/4}$ .

**Upper bounding  $\Theta_1$ :** When  $|\beta| \leq \frac{2}{s^{1/4} \varepsilon^{1/c_0}}$ ,

$$1 - (1-\gamma)^{2|\beta|} \leq 1 - \left(1 - \frac{4}{s^{1/4} \varepsilon^{1/c_0}} \gamma\right) = \frac{4}{s^{1/4} \varepsilon^{1/c_0}} \gamma = 4s^{c_0/4}.$$

Again since the sum of squares of Fourier coefficients is 1,  $|\Theta_1| \leq 4s^{c_0/4}$ .

**Upper bounding  $\Theta_2$ :** From Property 2 of [Theorem 3.2.5](#), we have that for any  $v \in V$  and  $\beta$  with  $|\beta| > \frac{2}{s^{1/4}\varepsilon^{1/c_0}}$ , the probability that  $|\tilde{\pi}_{uv}(\beta)| < 1/\varepsilon s^{c_0/4}$ , for a random neighbor  $u$ , is at most  $\varepsilon s^{c_0/4}$ . Hence  $|\Theta_2| \leq s^{c_0/4}$ . ■

Fix  $u, v, w$  chosen by the test. Recall that we thought of  $f_v$  as having domain  $\prod_{i \in [L]} \Omega_i$  where  $\Omega_i = \{0, 1\}^{2d}$  corresponds to the set of coordinates in  $B_{uv}(i) \cup B'_{uv}(i)$ . Since the grouping of coordinates depends on  $u$ , we define  $\overline{\text{Inf}}_i^u[f_v] := \text{Inf}_i[f_v]$  where  $i \in [L]$  for explicitness. From Equation (3.2.1),

$$\overline{\text{Inf}}_i^u[f_v] = \sum_{\alpha \subseteq [2dL]: i \in \tilde{\pi}_{uv}(\alpha)} \widehat{f}_v(\alpha)^2,$$

where  $\tilde{\pi}_{uv}(\alpha) := \{i \in [L] : \exists j \in [R], (j \in \alpha \vee j + R \in \alpha) \wedge \pi_{uv}(j) = i\}$ .

**Claim 3.4.6.** Let  $\tau_{u,v,w} := \sum_{i \in [L]} \overline{\text{Inf}}_i^u[T_{1-\gamma}f_v] \cdot \overline{\text{Inf}}_i^u[T_{1-\gamma}f_w]$ .

$$\begin{aligned} \mathbf{E}_{u,v,w} \left| \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma}f_v(x_i) T_{1-\gamma}f_w(y_i) \right] - \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma}f_v(x_i) \right] \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma}f_w(y_i) \right] \right| \\ \leq 2^{O(k)} \sqrt{\frac{\mathbf{E}_{u,v,w} \tau_{u,v,w}}{\gamma}}. \end{aligned}$$

*Proof.* It is easy to check that  $\sum_{i \in [L]} \overline{\text{Inf}}_i^u[T_{1-\gamma}f_v] \leq 1/\gamma$  (c.f., [\[Wen13, Lemma 1.13\]](#)).

For any  $u, v, w$ , since the test distribution satisfies the conditions of [Theorem 2.4.2](#), we get

$$\left| \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma}f_v(x_i) T_{1-\gamma}f_w(y_i) \right] - \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma}f_v(x_i) \right] \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma}f_w(y_i) \right] \right| \leq 2^{O(k)} \sqrt{\frac{\tau_{u,v,w}}{\gamma}}.$$

The claim follows by taking expectation over  $u, v, w$  and using the concavity of square root. ■

From [Claim 3.4.6](#) and [Claim 3.4.4](#) and using the fact the the marginals of the test distribution  $\mathcal{T}_2$  on  $(x_1, \dots, x_k)$  is the same as marginals on  $(y_1, \dots, y_k)$ , for  $\gamma := s^{(c_0+1)/4}\varepsilon^{1/c_0}$ , we get

$$\mathbf{E}_{u,v,w} \mathbf{E}_{X,Y \in \mathcal{T}_2} \left[ \prod_{i=1}^k f_v(x_i) f_w(y_i) \right] \geq -O(ks^{c_0/8}) - 2^{O(k)} \sqrt{\frac{\mathbf{E}_{u,v,w} \tau_{u,v,w}}{\gamma}} + \mathbf{E}_u \left( \mathbf{E}_v \mathbf{E}_{\mathcal{T}_2} \left[ \prod_{i=1}^k T_{1-\gamma}f_v(x_i) \right] \right)^2. \quad (3.4.6)$$

If  $\tau_{u,v,w}$  in expectation is large, there is a standard way of decoding the assignments to a labeling to the label cover instance, as shown in [Claim 3.4.7](#).

**Claim 3.4.7.** *If  $G$  is an at most  $s$ -satisfiable instance of LABEL-COVER then*

$$\mathbf{E}_{u,v,w} \tau_{u,v,w} \leq \frac{s}{\gamma^2}.$$

*Proof.* Note that  $\sum_{\alpha \subseteq [2R]} (1-\gamma)^{|\alpha|} \widehat{f}_v(\alpha)^2 \leq 1$ . We will give a randomized labeling to the LABEL-COVER instance. For each  $v \in V$ , choose a random  $\alpha \subseteq [2R]$  with probability  $(1-\gamma)^{|\alpha|} \widehat{f}_v(\alpha)^2$  and assign a uniformly random label  $j$  in  $\alpha$  to  $v$ ; if the label  $j \geq R$ , change the label to  $j - R$  and with the remaining probability assign an arbitrary label. For  $u \in U$ , choose a random neighbor  $w \in V$  and a random  $\beta \subseteq [2R]$  with probability  $(1-\gamma)^{|\beta|} \widehat{f}_w(\beta)^2$ , choose a random label  $\ell$  in  $\beta$  and assign the label  $\tilde{\pi}_{uw}(\ell)$  to  $u$ . With the remaining probability, assign an arbitrary label. The fraction of edges satisfied by this labeling is at least

$$\mathbf{E}_{u,v,w} \sum_{i \in [L]} \sum_{(\alpha, \beta): i \in \tilde{\pi}_{uv}(\alpha), i \in \tilde{\pi}_{uw}(\beta)} \frac{(1-\gamma)^{|\alpha|+|\beta|}}{|\alpha| \cdot |\beta|} \widehat{f}_v(\alpha)^2 \widehat{f}_w(\beta)^2.$$

Using the fact that  $1/r \geq \gamma(1-\gamma)^r$  for every  $r > 0$  and  $\gamma \in [0, 1]$ , we lower bound  $\frac{1}{|\alpha|}$  and  $\frac{1}{|\beta|}$  by  $\gamma(1-\gamma)^{|\alpha|}$  and  $\gamma(1-\gamma)^{|\beta|}$  respectively. The above is then lower bounded by

$$\gamma^2 \mathbf{E}_{u,v,w} \sum_{i \in [L]} \left( \sum_{\alpha: i \in \tilde{\pi}_{uv}(\alpha)} (1-\gamma)^{2|\alpha|} \widehat{f}_v(\alpha)^2 \right) \left( \sum_{\beta: i \in \tilde{\pi}_{uw}(\beta)} (1-\gamma)^{2|\beta|} \widehat{f}_w(\beta)^2 \right) = \gamma^2 \mathbf{E}_{u,v,w} \tau_{u,v,w}.$$

Since  $G$  is at most  $s$ -satisfiable, the labeling can satisfy at most  $s$  fraction of constraints and the above equation is upper bounded by  $s$ . ■

[Lemma 3.4.3](#) follows from the above claim and [Equation 3.4.6](#). ■

*Proof of Theorem 3.1.2.* Using [Theorem 3.2.5](#), the size of the CSP instance  $\mathcal{G}$  produced by the reduction is  $N = n^r 2^{2^{O(r)}}$  and the parameter  $s \leq 2^{-d_0 r}$ . Setting  $r = \Theta(\log \log n)$ , gives that  $N = 2^{\text{poly}(\log n)}$  for a constant  $k$ . [Lemma 3.4.3](#) and [Equation 3.4.2](#) imply that

$$O(k s^{c_0/8}) + 2^{O(k)} \frac{s^{(1-3c_0)/8}}{\varepsilon^{3/2c_0}} \geq \frac{1}{2^t - 1}.$$

Since  $k$  is a constant, this gives that  $t = \Omega(\log \log n)$ . ■

### 3.5 Improvement to covering hardness of 4-LIN

In this section, we prove [Theorem 3.1.4](#). We give a reduction from an instance of LABEL-COVER,  $G = (U, V, E, [L], [R], \{\pi_e\}_{e \in E})$  as in [Definition 3.2.4](#), to a 4-LIN-CSP instance  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ . The set of variables  $\mathcal{V}$  is  $V \times \{0, 1\}^{2R}$ . Any assignment to  $\mathcal{V}$  is given by a set of functions  $f_v : \{0, 1\}^{2R} \rightarrow \{0, 1\}$ , for each  $v \in V$ . The set of constraints  $\mathcal{E}$  is given by the following test which checks whether  $f_v$ 's are long codes of a good labeling to  $V$ .

#### Long Code Test $\mathcal{T}_3$

1. Choose  $u \in U$  uniformly and neighbors  $v, w \in V$  of  $u$  uniformly and independently at random.
2. Choose  $x, x', z, z'$  uniformly and independently from  $\{0, 1\}^{2R}$  and  $y$  from  $\{0, 1\}^{2L}$ . Choose  $(\eta, \eta') \in \{0, 1\}^{2L} \times \{0, 1\}^{2L}$  as follows: Independently for each  $i \in [L]$ ,  $(\eta_i, \eta_{L+i}, \eta'_i, \eta'_{L+i})$  is set to
  - (a)  $(0, 0, 0, 0)$  with probability  $1 - 2\varepsilon$ ,
  - (b)  $(1, 0, 1, 0)$  with probability  $\varepsilon$  and
  - (c)  $(0, 1, 0, 1)$  with probability  $\varepsilon$ .
3. For  $y \in \{0, 1\}^{2L}$ , let  $y \circ \pi_{uv} \in \{0, 1\}^{2R}$  be the string such that  $(y \circ \pi_{uv})_i := y_{\pi_{uv}(i)}$  for  $i \in [R]$  and  $(y \circ \pi_{uv})_i := y_{\pi_{uv}(i-R)+L}$  otherwise. Given  $\eta \in \{0, 1\}^{2L}, z \in \{0, 1\}^{2R}$ , the string  $\eta \circ \pi_{uv} \cdot z \in \{0, 1\}^{2R}$  is obtained by taking coordinate-wise product of  $\eta \circ \pi_{uv}$  and  $z$ . Accept iff

$$f_v(x) + f_v(x + y \circ \pi_{uv} + \eta \circ \pi_{uv} \cdot z) + f_w(x') + f_w(x' + y \circ \pi_{uw} + \eta' \circ \pi_{uw} \cdot z' + 1) = 1 \pmod{2}. \quad (3.5.1)$$

(Here by addition of strings, we mean the coordinate-wise sum modulo 2.)

**Lemma 3.5.1** (Completeness). *If  $G$  is an YES instance of LABEL-COVER, then there exists  $f, g$  such that each of them covers  $1 - \varepsilon$  fraction of  $\mathcal{E}$  and they together cover all of  $\mathcal{E}$ .*

*Proof.* Let  $\ell : U \cup V \rightarrow [L] \cup [R]$  be a labeling to  $G$  that satisfies all the constraints. Consider the assignments given by  $f_v(x) := x_{\ell(v)}$  and  $g_v(x) := x_{R+\ell(v)}$  for each  $v \in V$ . On input  $f_v$ , for any pair of edges  $(u, v), (u, w) \in E$ , and  $x, x', z, z', \eta, \eta', y$  chosen by the long code test  $\mathcal{T}_3$ , the LHS in (3.5.1) evaluates to

$$x_{\ell(v)} + x_{\ell(v)} + y_{\ell(u)} + \eta_{\ell(u)} z_{\ell(v)} + x'_{\ell(w)} + x'_{\ell(w)} + y_{\ell(u)} + \eta'_{\ell(u)} z'_{\ell(w)} + 1 = \eta_{\ell(u)} z_{\ell(v)} + \eta'_{\ell(u)} z'_{\ell(w)} + 1.$$

Similarly for  $g_v$ , the expression evaluates to  $\eta_{L+\ell(u)} z_{R+\ell(v)} + \eta'_{L+\ell(u)} z'_{R+\ell(w)} + 1$ . Since  $(\eta_i, \eta'_i) = (0, 0)$  with probability  $1 - \varepsilon$ , each of  $f, g$  covers  $1 - \varepsilon$  fraction of  $\mathcal{E}$ . Also for  $i \in [L]$  whenever  $(\eta_i, \eta'_i) = (1, 1)$ ,  $(\eta_{L+i}, \eta'_{L+i}) = (0, 0)$  and vice versa. So one of the two evaluations above is 1 (mod 2). Hence the pair of assignment  $f, g$  cover  $\mathcal{E}$ . ■

**Lemma 3.5.2** (Soundness). *Let  $c_0$  be the constant from Theorem 3.2.5. If  $G$  is at most  $s$ -satisfiable with  $s < \frac{\delta^{10/c_0+5}}{4}$ , then any independent set in  $\mathcal{G}$  has fractional size at most  $\delta$ .*

*Proof.* Let  $I \subseteq \mathcal{V}$  be an independent set of fractional size  $\delta$  in the constraint graph  $\mathcal{G}$ . For every variable  $v \in V$ , let  $f_v : \{0, 1\}^{2R} \rightarrow \{0, 1\}$  be the indicator function of the independent set restricted to the vertices that correspond to  $v$ . Since  $I$  is an independent set, we have

$$\mathbf{E}_{u,v,w} \mathbf{E}_{\substack{x,x', \\ z,z', \\ \eta,\eta',y}} [f_v(x) f_v(x + y \circ \pi_{uv} + \eta \circ \pi_{uv} \cdot z) f_w(x') f_w(x' + y \circ \pi_{uw} + \eta' \circ \pi_{uw} \cdot z' + 1)] = 0. \quad (3.5.2)$$

For  $\alpha \subseteq [2R]$ , let  $\pi_{uv}^\oplus(\alpha) \subseteq [2L]$  be the set containing elements  $i \in [2L]$  such that if  $i < L$  there are an odd number of  $j \in [R] \cap \alpha$  with  $\pi_{uv}(j) = i$  and if  $i \geq L$  there are an odd number of  $j \in ([2R] \setminus [R]) \cap \alpha$  with  $\pi_{uv}(j - R) = i - L$ . It is easy to see that  $\chi_\alpha(y \circ \pi_{uw}) = \chi_{\pi_{uv}^\oplus(\alpha)}(y)$ . Expanding  $f_v$  in the Fourier basis and taking expectation over  $x, x'$  and  $y$ , we get that

$$\mathbf{E}_{u,v,w} \sum_{\alpha, \beta \subseteq [2R]: \pi_{uv}^\oplus(\alpha) = \pi_{uw}^\oplus(\beta)} \widehat{f}_v(\alpha)^2 \widehat{f}_w(\beta)^2 (-1)^{|\beta|} \mathbf{E}_{z,z',\eta,\eta'} [\chi_\alpha(\eta \circ \pi_{uv} \cdot z) \chi_\beta(\eta' \circ \pi_{uw} \cdot z')] = 0. \quad (3.5.3)$$

Now the expectation over  $z, z'$  simplifies as

$$\mathbf{E}_{u,v,w} \sum_{\alpha, \beta \subseteq [2R]: \pi_{uv}^{\oplus}(\alpha) = \pi_{uw}^{\oplus}(\beta)} \underbrace{\widehat{f}_v(\alpha)^2 \widehat{f}_w(\beta)^2 (-1)^{|\beta|} \Pr_{\eta, \eta'}[\alpha \cdot (\eta \circ \pi_{uv}) = \beta \cdot (\eta' \circ \pi_{uw}) = \bar{0}]}_{=: \text{Term}_{u,v,w}(\alpha, \beta)} = 0, \quad (3.5.4)$$

where we think of  $\alpha, \beta$  as the characteristic vectors in  $\{0, 1\}^{2R}$  of the corresponding sets. We will now break up the above summation into different parts and bound each part separately. For a projection  $\pi : [R] \rightarrow [L]$ , define  $\tilde{\pi}(\alpha) := \{i \in [L] : \exists j \in [R], (j \in \alpha \vee j + R \in \alpha) \wedge (\pi(j) = i)\}$ . We need the following definitions.

$$\begin{aligned} \Theta_0 &:= \mathbf{E}_{u,v,w} \sum_{\substack{\alpha, \beta: \\ \pi_{uv}^{\oplus}(\alpha) = \pi_{uw}^{\oplus}(\beta) = \emptyset}} \text{Term}_{u,v,w}(\alpha, \beta), \\ \Theta_1 &:= \mathbf{E}_{u,v,w} \sum_{\substack{\alpha, \beta: \\ \pi_{uv}^{\oplus}(\alpha) = \pi_{uw}^{\oplus}(\beta) \neq \emptyset, \\ \max\{|\alpha|, |\beta|\} \leq 2/\delta^{5/c_0}}} \text{Term}_{u,v,w}(\alpha, \beta), \\ \Theta_2 &:= \mathbf{E}_{u,v,w} \sum_{\substack{\alpha, \beta: \\ \pi_{uv}^{\oplus}(\alpha) = \pi_{uw}^{\oplus}(\beta) \neq \emptyset, \\ \max\{|\tilde{\pi}_{uv}(\alpha)|, |\tilde{\pi}_{uw}(\beta)|\} \geq 1/\delta^5}} \text{Term}_{u,v,w}(\alpha, \beta), \\ \Theta_3 &:= \mathbf{E}_{u,v,w} \sum_{\substack{\alpha, \beta: \\ \pi_{uv}^{\oplus}(\alpha) = \pi_{uw}^{\oplus}(\beta) \neq \emptyset, \\ \max\{|\alpha|, |\beta|\} > 2/\delta^{5/c_0}, \\ \max\{|\tilde{\pi}_{uv}(\alpha)|, |\tilde{\pi}_{uw}(\beta)|\} < 1/\delta^5}} \text{Term}_{u,v,w}(\alpha, \beta). \end{aligned}$$

**Lower bounding  $\Theta_0$ :** If  $\pi_{uw}^{\oplus}(\beta) = \emptyset$ , then  $|\beta|$  is even. Hence, all the terms in  $\Theta_0$  are positive and

$$\Theta_0 \geq \mathbf{E}_{u,v,w} \text{Term}_{u,v,w}(0, 0) = \mathbf{E}_u \left( \mathbf{E}_v \widehat{f}_v(0)^2 \right)^2 \geq \left( \mathbf{E}_{u,v} \widehat{f}_v(0) \right)^4 = \delta^4.$$

**Upper bounding  $\Theta_1$ :** Consider the following strategy for labeling vertices  $u \in U$  and  $v \in V$ . For  $u \in U$ , pick a random neighbor  $v$ , choose  $\alpha$  with probability  $\widehat{f}_v(\alpha)^2$  and set its label to a random element in  $\tilde{\pi}_{uv}(\alpha)$ . For  $w \in V$ , choose  $\beta$  with probability  $\widehat{f}_w(\beta)^2$  and set its label to a random element of  $\beta$ . If the label  $j \geq R$ , change the label to  $j - R$ . The probability that a random edge  $(u, w)$  of the label cover is satisfied by

this labeling is

$$\begin{aligned}
\mathbf{E}_{u,v,w} \sum_{\substack{\alpha,\beta: \\ \tilde{\pi}_{uv}(\alpha) \cap \tilde{\pi}_{uw}(\beta) \neq \emptyset}} \hat{f}_v(\alpha)^2 \hat{f}_w(\beta)^2 \frac{1}{|\tilde{\pi}_{uv}(\alpha)| \cdot |\beta|} &\geq \mathbf{E}_{u,v,w} \sum_{\substack{\alpha,\beta: \\ \pi_{uv}^\oplus(\alpha) = \pi_{uw}^\oplus(\beta) \neq \emptyset \\ \max\{|\alpha|, |\beta|\} \leq 2/\delta^{5/c_0}}} \hat{f}_v(\alpha)^2 \hat{f}_w(\beta)^2 \frac{\delta^{10/c_0}}{4} \\
&\geq |\Theta_1| \cdot \frac{\delta^{10/c_0}}{4}.
\end{aligned}$$

Since the instance is at most  $s$ -satisfiable, the above is upper bounded by  $s$ . Choosing  $s < \frac{\delta^{10/c_0+5}}{4}$ , will imply  $|\Theta_1| \leq \delta^5$ .

**Upper bounding  $\Theta_2$ :** Suppose  $|\tilde{\pi}_{uv}(\alpha)| \geq 1/\delta^5$ , then note that

$$\Pr_{\eta, \eta'}[\alpha \cdot (\eta \circ \pi_{uv}) = \beta \cdot (\eta' \circ \pi_{uw}) = 0] \leq \Pr_{\eta}[\alpha \cdot (\eta \circ \pi_{uv}) = 0] \leq (1 - \varepsilon)^{|\tilde{\pi}_{uv}(\alpha)|} \leq (1 - \varepsilon)^{1/\delta^5}.$$

Since the sum of squares of Fourier coefficients of  $f$  is less than 1 and  $\varepsilon$  is a constant, we get that  $|\Theta_2| \leq 1/2^{\Omega(1/\delta^5)} < O(\delta^5)$ .

**Upper bounding  $\Theta_3$ :** From the third property of [Theorem 3.2.5](#), we have that for any  $v \in V$  and  $\alpha \subseteq [2R]$  with  $|\alpha| > 2/\delta^{5/c_0}$ , the probability that  $|\tilde{\pi}_{uv}(\alpha)| < 1/\delta^5$ , for a random neighbor  $u$  of  $v$ , is at most  $\delta^5$ . Hence  $|\Theta_3| \leq \delta^5$ .

On substituting the above bounds in Equation (3.5.4), we get that  $\delta^4 - O(\delta^5) \leq 0$  which gives a contradiction for small enough  $\delta$ . Hence there is no independent set in  $\mathcal{G}$  of size  $\delta$ . ■

*Proof of [Theorem 3.1.4](#).* From [Theorem 3.2.5](#), the size of the CSP instance  $\mathcal{G}$  produced by the reduction is  $N = n^r 2^{2^{O(r)}}$  and the parameter  $s \leq 2^{-d_0 r}$ . Setting  $r = \Theta(\log \log n)$ , gives that  $N = 2^{\text{poly}(\log n)}$  and the size of the largest independent set  $\delta = 1/\text{poly}(\log n) = 1/\text{poly}(\log N)$ . ■

## Chapter 4

### Parallel Repetition

#### 4.1 Introduction

##### 4.1.1 Label-cover and general two-prover games

A label cover instance is specified by a bipartite graph  $G = ((X, Y), E)$ , a pair of alphabets  $\Sigma_X$  and  $\Sigma_Y$  and a set of constraints  $\psi_e : \Sigma_X \rightarrow \Sigma_Y$  on each edge  $e \in E$ . The goal is to label the vertices of  $X$  and  $Y$  using labels from  $\Sigma_X$  and  $\Sigma_Y$  so as to satisfy as many constraints as possible.

This problem is often viewed as a two-prover game. The verifier picks an edge  $(x, y)$  at random and sends  $x$  to the first prover and  $y$  to the second prover. They are to return a label of the vertex that they received, and the verifier accepts if the labels they returned are consistent with the constraint  $\psi_{(x,y)}$ . The value of this game  $G$ , denoted by  $\text{val}(G)$ , is given by the acceptance probability of the verifier maximized over all possible strategies of the provers. These are also called *projection games* as the constraints are functions from  $\Sigma_X$  to  $\Sigma_Y$ . They are called *general games* if the constraint on each edge is an arbitrary relation  $\psi_{(x,y)} \subseteq \Sigma_X \times \Sigma_Y$ .

These two notions are equivalent in the sense that  $\text{val}(G)$  is exactly equal to the maximum fraction of constraints that can be satisfied by any labelling.

This problem is central to the PCP Theorem [AS98, ALM<sup>+</sup>98] and almost all inapproximability results that stem from it. The (Strong) PCP Theorem can be rephrased as stating that for every  $\varepsilon > 0$ , it is NP-hard to distinguish whether a given label cover instance has  $\text{val}(G) = 1$  or  $\text{val}(G) < \varepsilon$ . An important step is a way to transform instances with  $\text{val}(G) < 1 - \varepsilon$  to instances  $G'$  with  $\text{val}(G') < \varepsilon$ . This is usually achieved

via the *Parallel Repetition Theorem*.

#### 4.1.2 Parallel Repetition

The  $k$ -fold repetition of a game  $G$ , denoted by  $G^k$ , is the following natural definition — the verifier picks  $k$  edges  $(x_1, y_1), \dots, (x_k, y_k)$  from  $E$  uniformly and independently, sends  $(x_1, \dots, x_k)$  and  $(y_1, \dots, y_k)$  to the provers respectively, and accepts if the labels returned by them are consistent on each of the  $k$  edges.

If  $\text{val}(G) = 1$  to start with then  $\text{val}(G^k)$  still remains 1. How does  $\text{val}(G^k)$  decay with  $k$  if  $\text{val}(G) < 1$ ? Turns out even this simple operation of repeating a game in parallel has a counter-intuitive effect on the value of the game. It is easy to see that  $\text{val}(G^k) \geq \text{val}(G)^k$  as provers can use a same strategy as in  $G$  to answer each query  $(x_i, y_i)$ . The first surprise is  $\text{val}(G^k)$  is *not*  $\text{val}(G)^k$ , but sometimes can be *much larger* than  $\text{val}(G)^k$ . Fortnow [For89] presented a game  $G$  for which  $\text{val}(G^2) > \text{val}(G)^2$ , Feige [Fei91] improved this by giving an example of game  $G$  with  $\text{val}(G) < 1$  but  $\text{val}(G^2) = \text{val}(G)$ . Indeed, there are known examples [Raz11] of projection games where  $\text{val}(G) = (1 - \varepsilon)$  but  $\text{val}(G^k) \geq (1 - \varepsilon\sqrt{k})$  for a large range of  $k$ .

The first non trivial upper bound on  $\text{val}(G^k)$  was proven by Verbitsky [Ver96] who showed that if  $\text{val}(G) < 1$  then the value  $\text{val}(G^k)$  must go to zero as  $k$  goes to infinity. It is indeed true that  $\text{val}(G^k)$  decays exponentially with  $k$  (if  $\text{val}(G) < 1$ ). This breakthrough was first proved by Raz [Raz98], and has subsequently seen various simplifications and improvements in parameters [Hol09, Rao11, DS14a, BG14]. The following statements are due to Holenstein [Hol09], Dinur and Steurer [DS14a] respectively.

**Theorem 4.1.1** (Parallel repetition theorem for general games). *Suppose  $G$  is a projection game such that  $\text{val}(G) \leq 1 - \varepsilon$  and let  $|\Sigma_X| |\Sigma_Y| \leq s$ . Then, for any  $k \geq 0$ ,*

$$\text{val}(G^k) \leq (1 - \varepsilon^3/2)^{\Omega(k/\log s)}.$$

**Theorem 4.1.2** (Parallel repetition theorem for projection games). *Suppose  $G$  is a projection game such that  $\text{val}(G) \leq \rho$ . Then, for any  $k \geq 0$ ,*

$$\text{val}(G^k) \leq \left( \frac{2\sqrt{\rho}}{1 + \rho} \right)^{k/2}.$$

Although a lot of these results are substantial simplifications of earlier proofs, they continue to be involved and delicate. Arguably, one might still hesitate to call them *elementary* proofs.

Recently, Moshkovitz [Mos14] came up with an ingenious method to prove a parallel repetition theorem for certain projection games by slightly modifying the underlying game via a process that she called *fortification*. The method of fortification suggested in [Mos14] was a rather mild change to the underlying game and proving parallel repetition for such *fortified projection games* was sufficient for most applications. The advantage of fortification was that parallel repetition theorem for fortified games had a simple, elementary and elegant proof as seen in [Mos14].

#### 4.1.3 Fortified games

Fortified games will be described more formally in Section 4.2, but we give a very rough overview here. Moshkovitz showed that there is an easy way to bound the value of repeated game if we knew that the game was *robust on large rectangles*. We shall first need the notion of *symmetrized projection games*.

**Symmetrized Projection games.** Given a projection game  $G$  on  $((X, Y), E)$ , the symmetrized game  $G_{\text{sym}}$  is a game on the (multi)graph  $((X, X), E')$  such that, there is an edge  $(x, x') \in E'$ , for every  $y \in Y$  with  $(x, y), (x', y) \in E$ , with the constraint  $\pi_{(x,y)}(\sigma_x) = \pi_{(x',y)}(\sigma_{x'})$ .

For projection games, it would be more convenient to work with the above symmetrized version for reasons that shall be explained shortly. It is not hard to see that  $\text{val}(G)$  and  $\text{val}(G_{\text{sym}})$  are within a quadratic factor of each other. Thus for projection games, we shall work with the game  $G_{\text{sym}}$  instead of the original game  $G$ .

**Definition 4.1.3** ( $(\delta, \varepsilon)$ -robust games). *Let  $G$  be a two-prover game on  $((X, X), E)$ . For any pair of sets  $S, T \subseteq X$ , let  $G_{S \times T}$  be the game where the verifier chooses his*

random query  $(x, x') \in E$  conditioned on the event that  $x \in S$  and  $x' \in T$ .

$G$  is said to be  $(\delta, \varepsilon)$ -robust if for every  $S, T \subseteq X$  with  $|S|, |T| \geq \delta|X|$ , we have that

$$\text{val}(G_{S \times T}) \leq \text{val}(G) + \varepsilon.$$

**Theorem 4.1.4** (Parallel repetition for robust projection games [Mos14]). *Let  $G$  be a projection game on a bi-regular bipartite graph  $((X, Y), E)$  with alphabets  $\Sigma_X$  and  $\Sigma_Y$ . For any positive integer  $k$ , if  $\varepsilon_1, \varepsilon_2, \delta > 0$  are parameters such that  $2\delta|\Sigma_Y|^{k-1} \leq \varepsilon_1$  and  $G_{\text{sym}}$  is  $(\delta, \varepsilon_2)$ -robust, then<sup>1</sup>*

$$\text{val}(G_{\text{sym}}^k) \leq (\text{val}(G_{\text{sym}}) + \varepsilon_2)^k + k\varepsilon_1.$$

Not all projection games are robust on large rectangles, but Moshkovitz suggested a neat way of slightly modifying a projection game and making it robust. This process was called *fortification*.

On a high level, for any two-prover game, the verifier chooses to verify a constraint corresponding to an edge  $(x, y)$  but is instead going to sample several other dummy vertices and give the provers two sets of  $D$  vertices  $\{x_1, \dots, x_D\}$  and  $\{y_1, \dots, y_D\}$  such that  $x = x_i$  and  $y = y_j$  for some  $i$  and  $j$ . The provers are expected to return labels of all  $D$  vertices sent to them but the verifier checks consistency on just the edge  $(x, y)$ . This is very similar to the “confuse/match” perspective of Feige and Kilian [FK94].

To derandomize this construction, Moshkovitz [Mos14] uses a pseudo-random bipartite graph where given a vertex  $w$ , the provers are expected to return labels of all its neighbours (Definition 4.2.1). The most natural candidate of such a pseudo-random graph is an  $(\delta, \varepsilon)$ -extractor, as we really want to ensure that conditioned on “large enough events”  $S$  and  $T$ , the underlying distribution on the constraints does not change much. This makes a lot of intuitive sense, since on choosing a random element of  $S$  and then a random neighbour, the extractor property guarantee that the induced distribution on vertices in  $X$  is  $\varepsilon$ -close to uniform. Thus, it is natural to expect that conditioning on the events  $S$  and  $T$  should not change the underlying distribution on the constraints by more than  $O(\varepsilon)$ . This was the rough argument in [Mos14],

---

<sup>1</sup>The following is the corrected statement from [Mos15].

which unfortunately turns out to be false. We elaborate on this in [Section 4.4.1](#) and [Section 4.6](#).

A recent updated version [\[Mos15\]](#) of [\[Mos14\]](#) provides an different argument for the fortification lemma using a stronger extractor. We discuss this at the end of [Section 4.1.4](#).

#### 4.1.4 Results

We present a fix to the approach of [\[Mos14\]](#), by describing a way to transform any given game instance  $G$  into a robust instance  $G^*$  with the same value following the framework of [\[Mos14\]](#) but using a different graph for concatenation, and a different analysis.

We first describe a concrete counter-example to the original argument of [\[Mos14\]](#) in [Section 4.4.1](#), that shows concatenating ([Definition 4.2.1](#)) with an arbitrary  $(\delta, \varepsilon)$ -extractor is insufficient. In fact, as we show in [Section 4.7](#), *concatenating* ([Definition 4.2.1](#)) with *any* left-regular graph with left-degree by  $o(1/\varepsilon\delta)$  fails to make arbitrary instances  $(\delta, \varepsilon)$ -robust. We instead use bipartite graphs called *fortifiers*, defined below.

**Definition 4.1.5** (Fortifiers). *A bipartite graph  $H = ((W, X), E_H)$  is an  $(\delta, \varepsilon_1, \varepsilon_2)$ -fortifier if for any set  $S \subseteq W$  such that  $|S| \geq \delta|W|$ , if  $\pi$  is the probability distribution on  $X$  induced by picking a uniformly random element  $w$  from  $S$ , and a uniformly random neighbor  $x$  of  $w$ , then*

$$|\pi - \mathbf{u}|_1 \leq \varepsilon_1 \quad \text{and} \quad \|\pi - \mathbf{u}\|^2 \leq \frac{\varepsilon_2}{|X|}.$$

Notice that a fortifier is an extractor, with the additional condition that the  $\ell_2$ -distance of  $\pi$  from the uniform distribution is small. This is what enables us to show that *concatenation* ([Definition 4.2.1](#)) with a fortifier produces a robust instance.

**Theorem 4.1.6** (Fortifiers imply robustness). *Suppose  $G$  is a two-prover projection game on a bi-regular graph  $((X, Y), E)$ . Then, for any  $\varepsilon, \delta > 0$ , if  $H = ((W, X), E_H)$  is a  $(\delta, \varepsilon, \varepsilon)$ -fortifier, then the symmetrized concatenated game  $G^* = (H \circ G)_{\text{sym}}$  is  $(\delta, O(\varepsilon))$ -robust.*

In particular, bipartite spectral expanders are good fortifiers, as [Lemma 4.2.5](#) shows. This gives us our main result which follows from [Lemma 4.2.5](#) and [Theorem 4.1.6](#):

**Corollary 4.1.7.** *Let  $G$  be a two-prover projection game on a bi-regular graph  $((X, Y), E)$ . For any  $\varepsilon, \delta > 0$ , if  $H = ((X, X), E_H)$  is a symmetric bipartite graph that is a  $\lambda$ -expander ([Definition 2.5.1](#)) with  $\lambda < \varepsilon\sqrt{\delta}$  then the symmetrized concatenated game  $G^* = (H \circ G)_{\text{sym}}$  is  $(\delta, 4\varepsilon)$ -robust.*

As one would expect, the condition on the fortifier can be relaxed if the underlying graph of  $G_{\text{sym}}$  is a spectral-expander. We prove the following theorem. [Theorem 4.1.6](#) follows from this theorem by setting  $\lambda_0 = 1$ .

**Theorem 4.1.8.** *Let  $G$  be a two-prover projection game on bi-regular graph  $((X, Y), E)$  where  $G_{\text{sym}}$  is a  $\lambda_0$ -expander. Then for any  $\varepsilon, \delta > 0$ , if  $H = ((W, X), E_H)$  is a  $(\delta, \varepsilon, (\varepsilon/\lambda_0))$ -fortifier, then the symmetrized concatenated game  $G^* = (H \circ G)_{\text{sym}}$  is  $(\delta, O(\varepsilon))$ -robust.*

One could ask if the definition of a fortifier is too strong, or if a weaker object would suffice. We argue in [Section 4.4](#) that if we proceed through concatenation, fortifiers are indeed necessary to make a game robust.

Bipartite Ramanujan graphs of degree  $\Theta(1/\varepsilon^2\delta)$  have  $\lambda < \varepsilon\sqrt{\delta}$  and are therefore good fortifiers. In [Section 4.7](#), we show that this is almost optimal by proving a lower bound of  $\Omega(1/\varepsilon\delta)$  on the left-degree of any graph that can achieve  $(\delta, \varepsilon)$ -robustness. This shows that our construction of using expanders to achieve robustness is almost optimal, in terms of the degree of the fortifier graph. Note that the degree of the fortifier is important as the alphabet size of the concatenated game is the alphabet size of the original game raised to the degree. There are known explicit constructions of bi-regular  $(\delta, \varepsilon)$ -extractors with left-degree  $\text{poly}(1/\varepsilon)\text{poly}\log(1/\delta)$ . But the lower bound in [Section 4.4](#) shows that  $(\delta, \varepsilon)$ -extractors are not fortifiers if  $\delta \ll \varepsilon$ , which is usually the relevant setting (see [Theorem 4.1.4](#)).

Independently, the author of [\[Mos14\]](#) came up with a different argument to obtain robustness of projection games by using a  $(\delta, \varepsilon\delta)$ -extractor. This is described in an

updated version [Mos15] present on the author's homepage.

It is also seen from Theorem 4.1.8 that bi-regular  $(\delta, \varepsilon\delta)$ -extractors are indeed  $(\delta, \varepsilon, \varepsilon)$ -fortifiers as well. Using an expander instead is arguably simpler, and is almost optimal.

**Remark 4.1.9.** *Although this fix provides a proof of a Parallel Repetition Theorem for projection games following the framework of [Mos14], the degree of the fortifier is too large to get the required PCP for proving optimal hardness of the SET-COVER problem that Dinur and Steurer [DS14a] obtained. See [Mos15] for a discussion on this.*

#### Remark about parallel repetition for general games

A fairly straightforward generalization Theorem 4.1.4 to robust general games on bi-regular graphs is the following.

**Claim 4.1.10.** *Let  $G$  be a general two-prover game on a bi-regular graph  $((X, Y), E)$  with alphabets  $\Sigma_X$  and  $\Sigma_Y$ . For any positive integer  $k$ , if  $\varepsilon, \delta > 0$  are parameters such that  $2\delta|\Sigma_X \times \Sigma_Y|^{k-1} \leq \varepsilon$  and  $G$  is  $(\delta, \varepsilon)$ -robust, then*

$$\text{val}(G^k) \leq (\text{val}(G) + \varepsilon)^k + k\varepsilon.$$

One could attempt a fortifying any game by using a fortifier on both sides. But the issue with this procedure is that it makes  $|\Sigma_X| = \exp(1/\delta)$  and in such scenarios  $\delta|\Sigma_X| \gg 1$  making it infeasible to ensure  $2\delta|\Sigma_X \times \Sigma_Y|^{k-1} \leq \varepsilon$ . Hence, though Lemma 4.1.10 may be useful in cases where we know that the game  $G$  is robust via other means, the technique of fortification via concatenation increases the alphabet size too much for Lemma 4.1.10 to be applicable.

For the case of projection games, this is not an issue as Theorem 4.1.4 only requires  $2\delta|\Sigma_Y|^{k-1} < \varepsilon$  and concatenating  $G_{\text{sym}}$  by a fortifier only increases  $|\Sigma_X|$  and keeps  $\Sigma_Y$  unchanged. Thus, one can indeed choose  $\varepsilon$  and  $\delta$  small enough to give a parallel repetition theorem for a robust version of an arbitrary projection game.

## 4.2 Preliminaries

### Notation

- For any vector  $\mathbf{a}$ , let  $|\mathbf{a}|_1 := \sum_i |\mathbf{a}_i|$ , and  $\|\mathbf{a}\| := \sqrt{\sum_i \mathbf{a}_i^2}$  be the  $\ell_1$  and  $\ell_2$ -norms respectively.
- We shall use  $\mathbf{u}_S$  to refer to the uniform distribution on a set  $S$ . Normally, the set  $S$  would be clear from context and in such case we shall drop the subscript  $S$ .
- For any vector  $\mathbf{a}$ , we shall use  $\mathbf{a}^\parallel$  to refer to the component along the direction of  $\mathbf{u}$ , and  $\mathbf{a}^\perp$  to refer to the component orthogonal to  $\mathbf{u}$ .
- We shall assume that the underlying graph for the games is bi-regular.

We define the *concatenation* operation of a two-prover games with a bipartite graph that was alluded to in [Section 4.1.3](#).

**Definition 4.2.1** (Concatenation). *Given bipartite graphs  $G = ((X, Y), E)$ ,  $H = ((W, X), E_H)$  where  $H$  is regular with left degree  $D$ , the concatenated graph  $H \circ G = ((W, Y), E')$  is a multigraph such that there is an edge  $(w, y) \in E'$ , for every pair of edges  $(w, x) \in E_H, (x, y) \in E$ .*

*Given a two-prover projection game on a graph  $G = ((X, Y), E)$  with a set of constraints  $\psi$ , a pair of alphabets  $\Sigma_X$  and  $\Sigma_Y$ , a bipartite graph  $H = ((W, X), E_H)$  with left degree  $D$ , the concatenated game is a game on the multigraph  $H \circ G = ((W, Y), E')$  with  $\Sigma_W = \Sigma_X^D$ . For any edge  $(w, y) \in E'$  which corresponds to the pair  $(w, x) \in E, (x, y) \in E_H$ , the constraint  $\pi_{(w, y)}(a) := \pi_{x, y}(a_x)$ , where  $a \in \Sigma_X^D$  and  $a_x$  is the alphabet at the coordinate corresponding to  $x$  (assuming some fixed ordering of vertices in  $X$ ). The distribution over the edges in the multigraph  $H \circ G$  is uniform.*

**Remark 4.2.2.** *The concatenated game  $H \circ G$  is also a projection game. We shall be working with the symmetrized version  $G^* = (H \circ G)_{\text{sym}}$  of this game.*

**Lemma 4.2.3** (Concatenation preserves value). *[Mos14] Given any two-prover game on  $G$ , and a biregular bipartite graph  $H$ :*

$$\text{val}(H \circ G) = \text{val}(G).$$

## Expanders, extractors and fortifiers

Recall the definition of expanders and extractors from [Chapter 2](#). Our earlier definition of a fortifier ([Definition 4.1.5](#)) has properties of both an expander and an extractor. Indeed, we can build fortifiers by just taking a product an expander and an extractor.

**Lemma 4.2.4.** *Let  $H_1 = ((V, W), E_1)$  is a bi-regular  $(\delta, \varepsilon)$ -extractor, and let  $H_2 = (W, E_2)$  is a regular  $\lambda$ -expander. Denote  $H'_2$  to be the bipartite graph  $((W, W), E_2)$ . Then the concatenated graph  $H_1 \circ H'_2$  is an  $(\delta, \varepsilon, \lambda^2 \varepsilon / \delta)$ -fortifier.*

*Proof.* Let  $H_2$  be the normalized adjacency matrix of graph  $H_2$ . Let  $\pi_S$  denotes the probability distribution on  $W$  obtained by picking an element of  $S \subseteq V$  uniformly and then choosing a random neighbour in  $H_1$ . Thus,  $H_2 \pi_S$  is the probability distribution on  $W$  induced by the uniform distribution on  $S$  and a random neighbour in  $H_1 \circ H'_2$ . We want to show for all  $S$  such that  $|S| \geq \delta|V|$ ,

$$|H_2 \pi_S - \mathbf{u}|_1 \leq \varepsilon \quad \text{and} \quad \|H_2 \pi_S - \mathbf{u}\|^2 \leq \frac{\lambda^2 \varepsilon / \delta}{|X|}.$$

The first inequality is obtained as  $|H_2 \pi_S - \mathbf{u}|_1 = |H_2(\pi_S - \mathbf{u})|_1 \leq |\pi_S - \mathbf{u}|_1 \leq \varepsilon$ , where we use the fact that  $|H_2 v|_1 \leq |v|_1$  for any  $v$  and any normalized adjacency matrix, and  $|\pi_S - \mathbf{u}|_1 \leq \varepsilon$  follows from the extractor property of  $H_1$ .

As for the second inequality, observe that

$$\|\pi_S - \mathbf{u}\|^2 \leq \max_{w \in W} (\pi_S(w)) \cdot |\pi_S - \mathbf{u}|_1 \leq \varepsilon \cdot \max_{w \in W} (\pi_S(w)).$$

For a bi-regular extractor<sup>2</sup>  $H_1$  of left-degree  $D$ , the degree of any  $w \in W$  is  $(|V| \cdot D / |W|)$  and the number of edges out of  $S$  is least  $\delta|V| \cdot D$ . Hence,  $\max_w \pi_S(w) \leq 1/(\delta|W|)$ , which is achieved if all neighbours of  $w$  are in  $S$ . Therefore,

$$\begin{aligned} \|\pi_S - \mathbf{u}\|^2 &\leq \frac{(\varepsilon/\delta)}{|W|} \\ \Rightarrow \|H_2(\pi_S - \mathbf{u})\|^2 &\leq \lambda^2 \frac{|W|}{|X|} \|\pi_S - \mathbf{u}\|^2 \leq \frac{|W|}{|X|} \cdot \frac{\lambda^2 \cdot (\varepsilon/\delta)}{|W|} = \frac{\lambda^2 \cdot (\varepsilon/\delta)}{|X|}. \quad \blacksquare \end{aligned}$$

---

<sup>2</sup>The bound on the right-degree guaranteed by bi-regularity is crucial for this claim. Without this, extractors are not sufficient for fortification ([Section 4.4.1](#)).

In particular, any bi-regular  $(\delta, \varepsilon)$ -extractor is a  $(\delta, \varepsilon, \varepsilon/\delta)$ -fortifier. Hence, if the underlying graph  $G$  of the two-prover game is a  $\sqrt{\delta}$ -expander, then [Theorem 4.1.8](#) states that merely using an  $(\delta, \varepsilon)$ -extractor as suggested in [\[Mos14\]](#) would be sufficient to make it  $(\delta, O(\varepsilon))$ -robust.

Also, since any graph is trivially a 1-expander, a bi-regular  $(\delta, \varepsilon\delta)$ -extractor is also an  $(\delta, \varepsilon, \varepsilon)$ -fortifier. The following lemma also shows that expanders are also fortifiers with reasonable parameters as well.

**Lemma 4.2.5.** *Let  $H = (X, E_H)$  be any  $\lambda$ -expander. Then, for every  $\delta > 0$ , the bipartite graph  $H' = ((X, X), E_H)$  is also a  $(\delta, \sqrt{\lambda^2/\delta}, \lambda^2/\delta)$ -fortifier. In particular, if  $\lambda \leq \varepsilon\sqrt{\delta}$ , then  $H'$  is an  $(\delta, \varepsilon, \varepsilon)$ -fortifier.*

*Proof.* Let  $H$  be the normalized adjacency matrix of  $H$ . Let  $S \subseteq W$  such that  $|S| \geq \delta|W|$ . We have,

$$\|\mathbf{u}_S^\perp\|^2 \leq \frac{1}{\delta|W|}.$$

Hence, by the expansion property of  $H$ ,

$$\|H\mathbf{u}_S - \mathbf{u}\|^2 := \|H\mathbf{u}_S^\perp\|^2 \leq \lambda^2 \cdot \frac{|W|}{|X|} \cdot \|\mathbf{u}_S^\perp\|^2 \leq \frac{\lambda^2/\delta}{|X|}.$$

$|H\mathbf{u}_S - \mathbf{u}|_1 \leq \sqrt{\lambda^2/\delta}$  follows from above and Cauchy-Schwarz inequality. ■

Although [Lemma 4.2.5](#) shows that expanders are also fortifiers for reasonable parameters, the construction in [Lemma 4.2.4](#) is more useful when the underlying graph for the two-prover game is already a good expander. For example, if the underlying graph  $G$  was a  $\delta$ -expander, then [Theorem 4.1.8](#) suggests that we only require a  $(\delta, \varepsilon, \varepsilon/\delta)$ -fortifier. [Lemma 4.2.4](#) implies that an  $(\delta, \varepsilon)$ -extractor is already a  $(\delta, \varepsilon, \varepsilon/\delta)$ -fortifier and hence is sufficient to make the game robust. The main advantage of this is the degree of  $\delta$ -expanders must be  $\Omega(1/\delta^2)$  whereas we have explicit  $(\delta, \varepsilon)$ -extractors of degree  $(1/\varepsilon^2) \exp(\text{poly log log}(1/\delta))$  which has a much better dependence in  $\delta$ . This dependence on  $\delta$  is crucial for certain applications.

### 4.3 Sub-games on large rectangles

Consider a projection game on graph  $G = ((X, Y), E)$  which is biregular with degree  $d$ . For a biregular bipartite graph  $H = ((W, X), E_H)$  with degree  $d_H$ , consider the symmetrized concatenated game  $G^* = (H \circ G)_{\text{sym}} = ((W, W), E')$ . Let  $S, T \subseteq W$  and  $\mu_S$  (or  $\mu_T$ ) denote the induced distributions on  $X$  obtained by picking a uniformly random element of  $S$  (or  $T$ ) and taking a uniformly random neighbour in  $H$ . In the next claim, we give an expression for the distribution of verifier checking the underlying constraint on  $(x, x')$  in the subgame  $(G^*)_{S \times T}$ .

**Claim 4.3.1.** *For any  $x, x' \in X$  such that there are edges  $(x, y), (x', y) \in E$ ,*

$$\pi_{x, x'} = \frac{\mu_S(x) \mu_T(x')}{\sum_{(x, x') \in G_{\text{sym}}} \mu_S(x) \mu_T(x')}. \quad (4.3.1)$$

*Proof.* Let  $d_{S, x}, d_{T, x'}$  denote the degree of  $x$  to  $S$  and  $x'$  to  $T$  respectively in  $H$ . Let  $N_H(x)$  denote the neighbor set of a vertex  $x$  in  $H$ . Then,

$$\mu_S(x) = \frac{d_{S, x}}{\sum_{z \in X} d_{S, z}}.$$

The probability  $\pi_{x, x'}$  of the verifier in  $(G^*)_{S \times T}$  checking a constraint corresponding to a constraint  $(x, x')$  in  $G_{\text{sym}}$ , is proportional to the number of edges  $(w, w')$  in the graph  $G^*$  such that  $w \in S \cap N_H(x)$ , and  $w' \in T \cap N_H(x')$ . Since every such edge in  $G^*$  was equally likely, we have:

$$\pi_{x, x'} = \frac{d_{S, x} \cdot d_{T, x'}}{\sum_{(x, x') \in G_{\text{sym}}} d_{S, x} d_{T, x'}} = \frac{\mu_S(x) \mu_T(x')}{\sum_{(x, x') \in G_{\text{sym}}} \mu_S(x) \mu_T(x')}.$$

■

One way to show that the concatenated game  $G^*$  is  $(\delta, O(\varepsilon))$ -robust would be to show that the above distribution  $\pi_{x, x'}$  is  $O(\varepsilon)$ -close to uniform whenever  $|S|, |T|$  have density at least  $\delta$  because then the distribution on constraints that the verifier is going to check in  $G^*_{S \times T}$  is  $O(\varepsilon)$  close to the distribution on constraints in  $G$ . Hence, up to additive factor of  $O(\varepsilon)$  the quantity  $\text{val}(G^*_{S \times T})$  is same as  $\text{val}(G)$ . The main question here what properties should  $H$  satisfy so that the above distribution is close to uniform?

#### 4.4 Fortifiers are necessary

To prove that fortifiers are necessary, we shall restrict ourselves to games on graphs  $G = ((X, X), E)$ . We show that if a bipartite graph  $H = ((W, X), E_H)$ , makes a game on a particular graph  $G$ ,  $(\delta, O(\varepsilon))$ -robust, then  $H$  is a good fortifier.

As mentioned earlier, if the graph  $G$  had some expansion properties, then the requirements on the graph  $H$  to concatenate with can be relaxed. Thus, naturally, the worst case graph  $G$  is one that expands the least — a matching.

**Lemma 4.4.1** (Fortifiers are necessary). *Let  $\varepsilon, \delta > 0$  be small constants. Let  $H = ((W, X), E_H)$  be a bi-regular graph, and let  $G = ((X, X), E)$  be a matching. Suppose that for every subset  $S, T \subseteq W$  with  $|S|, |T| \geq \delta|W|$ , the distribution (defined in Equation (4.3.1)) induced by the sub game on  $S \times T$  of  $G^* := (H \circ G)_{\text{sym}}$  on the edges of  $G$  is  $\varepsilon$ -close to uniform. Then, for every  $S \subseteq W$  with  $|S| \geq \delta|W|$ ,*

$$|\mu_S - \mathbf{u}|_1 = \varepsilon, \quad (4.4.1)$$

$$\|\mu_S - \mathbf{u}\|^2 = \frac{O(\varepsilon)}{|X|}. \quad (4.4.2)$$

*Proof.* It is clear that (4.4.1) is necessary as the distribution on constraints in the sub-game  $G_{S \times W}^*$  (as defined in (4.3.1)) is essentially  $\mu_S$  (as  $\mu_T$  in this case is uniform).

As for (4.4.2), let us assume that

$$\|\mu_S - \mathbf{u}\|^2 = \frac{c}{|X|}.$$

Taking  $T = S$ , we obtain that the distribution (defined in Equation (4.3.1)) induced by the game  $G_{S \times S}^*$  on the edges of  $G$  is given by

$$\pi_{x,x} = \frac{\mu_S(x)^2}{\sum_x \mu_S(x)^2} = \left( \frac{|X|}{1+c} \right) \cdot \mu_S(x)^2,$$

where the last equality used the fact that  $\|\mu_S\|^2 = \|\mu_S^\perp\|^2 + \|\mathbf{u}\|^2$ .

$$\begin{aligned}
\sum_{x \in X} \left| \left( \frac{|X|}{c+1} \right) \cdot \mu_S(x)^2 - \frac{1}{|X|} \right| &= \left( \frac{|X|}{1+c} \right) \cdot \sum_{x \in X} \left| \mu_S(x)^2 - \frac{c+1}{|X|^2} \right| \\
&= \left( \frac{|X|}{1+c} \right) \cdot \sum_{x \in X} \left| \mu_S(x) - \frac{\sqrt{c+1}}{|X|} \right| \cdot \left( \mu_S(x) + \frac{\sqrt{c+1}}{|X|} \right) \\
&\geq \left( \frac{1}{\sqrt{1+c}} \right) \cdot \sum_{x \in X} \left| \mu_S(x) - \frac{\sqrt{c+1}}{|X|} \right| \\
&\geq \left( \frac{1}{\sqrt{1+c}} \right) \cdot \left( (\sqrt{1+c} - 1) - \sum_{x \in X} \left| \mu_S(x) - \frac{1}{|X|} \right| \right) \\
&\geq \left( \frac{1}{\sqrt{1+c}} \right) \cdot ((\sqrt{1+c} - 1) - \varepsilon).
\end{aligned}$$

Thus, if the distribution on constraints is  $\varepsilon$ -close to uniform, then the above lower bound forces  $c = O(\varepsilon)$ . ■

#### 4.4.1 General (non-regular) extractors are insufficient

Suppose  $H = ((W, X), E_H)$  is an arbitrary  $(\delta, O(\varepsilon))$ -extractor and  $G^*$  is the symmetrized concatenated game. Consider a possible scenario where there is a subset  $S \subseteq W$  with  $|S| \geq \delta|W|$  such that  $\mu_S$  is of the form

$$\mu_S = \left( \varepsilon, \frac{1-\varepsilon}{|X|-1}, \dots, \frac{1-\varepsilon}{|X|-1} \right).$$

Notice that this is a legitimate distribution that may be obtained from a large subset  $S$  as  $|\mu_S - \mathbf{u}|_1$  is easily seen to be at most  $2\varepsilon$ . However, if  $G = ((X, X), E)$  was  $d$ -regular with  $d = o(|X|)$ , then using (4.3.1), the probability mass on the edge  $(1, 1)$  on the sub-game over  $S \times S$  is

$$\pi_{1,1} = \left( \frac{\varepsilon^2}{\varepsilon^2 + O\left(\frac{\varepsilon d}{|X|}\right)} \right) \approx 1.$$

In other words, if such a distribution  $\mu_S$  can be induced by the extractor, then the provers can achieve value close to 1 in the game  $G_{S \times S}^*$  by just labelling the edge  $(1, 1)$  correctly. Thus,  $G^*$  is not even  $(\delta, 0.9)$ -robust.

In Section 4.6 we show that we can adversarially construct a  $(\delta, O(\varepsilon))$ -extractor, although non-regular, that induces such a skew distribution. In Section 4.7 we also show that left-regular graphs of left-degree  $o(1/\delta\varepsilon)$  are not fortifiers.

## 4.5 Robustness from fortifiers

In this section, we show that concatenating a symmetrized two-prover game by fortifier(s) yields a robust game as claimed by [Theorem 4.1.8](#).

**Lemma 4.5.1** (Distributions from large rectangles are close to uniform). *Let  $G = ((X, X), E)$  be a graph of a symmetrized two-prover game such that  $|X| = n$ . Let  $\mu_S$  and  $\mu_T$  be two probability distributions such that*

$$\left| \mu_S^\perp \right|_1 \leq \varepsilon_1 \quad \text{and} \quad \left| \mu_T^\perp \right|_1 \leq \varepsilon_1, \quad (4.5.1)$$

$$\left\| \mu_S^\perp \right\|^2 \leq \left( \frac{\varepsilon_2}{n} \right) \quad \text{and} \quad \left\| \mu_T^\perp \right\|^2 \leq \left( \frac{\varepsilon_2}{n} \right). \quad (4.5.2)$$

*If the bipartite graph  $G$  is a  $\lambda_0$ -expander then the distribution on edges  $(x, y)$  of  $G$  given by (4.3.1) is  $(2\varepsilon_1 + \varepsilon_1^2 + 2\lambda_0 \cdot \varepsilon_2)$ -close to uniform.*

As described in [Section 4.3](#), if  $H$  is a  $(\delta, \varepsilon_1, \varepsilon_2)$ -fortifier, then for any set  $S$  and  $T$  of density at least  $\delta$ , the distribution on the constraints of  $G_{S \times T}^*$  is given by (4.3.1). Applying the above lemma for the graph of the symmetrized game yields that the value of the game on any large rectangle can change only by the above bound on the statistical distance. By setting the parameters, [Theorem 4.1.8](#) follows immediately from [Lemma 4.5.1](#). Further, [Theorem 4.1.7](#) also follows from [Lemma 4.5.1](#) and [Lemma 4.2.5](#) as any graph is trivially a 1-expander.

The rest of this section would be devoted to the proof of [Lemma 4.5.1](#). For convenience, we let  $d$  be the left-degree (and hence also, right-degree) of the bipartite graph  $G$ . We shall prove [Lemma 4.5.1](#) by proving the following two claims.

**Claim 4.5.2.**

$$\sum_{(x,y) \in G} \left| \frac{\mu_S(x)\mu_T(y)}{\sum_{(x,y) \in G} \mu_S(x)\mu_T(y)} - \frac{\mu_S(x)\mu_T(y)}{d/n} \right| \leq \lambda_0 \cdot \varepsilon_2$$

**Claim 4.5.3.**

$$\sum_{(x,y) \in G} \left| \frac{\mu_S(x)\mu_T(y)}{d/n} - \frac{1}{n \cdot d} \right| \leq 2\varepsilon_1 + \varepsilon_1^2 + \lambda_0 \cdot \varepsilon_2$$

Clearly, Lemma 4.5.1 follows from Claim 4.5.2 and Claim 4.5.3.

*Proof of Claim 4.5.2.* Let  $G$  also denote the normalized biadjacency matrix of  $G$ . Observe that  $\sum_{(x,y) \in G} \mu_S(x) \mu_T(y) = d \cdot \langle G\mu_S, \mu_T \rangle$ . If we resolve  $\mu_S$  and  $\mu_T$  in the direction of the uniform distribution and the orthogonal component, we have

$$\begin{aligned} \langle G\mu_S, \mu_T \rangle &= \langle \mathbf{u}, \mathbf{u} \rangle + \langle G\mu_S^\perp, \mu_T^\perp \rangle = \frac{1}{n} + \langle G\mu_S^\perp, \mu_T^\perp \rangle \\ \Rightarrow \left| \langle G\mu_S, \mu_T \rangle - \frac{1}{n} \right| &\leq \lambda_0 \cdot \|\mu_S^\perp\| \cdot \|\mu_T^\perp\| \\ &\leq \left( \frac{\lambda_0 \cdot \varepsilon_2}{n} \right). \quad (\text{using (4.5.2)}) \end{aligned}$$

Therefore,

$$\begin{aligned} \sum_{(x,y) \in G} \left| \frac{\mu_S(x) \mu_T(y)}{d \langle G\mu_S, \mu_T \rangle} - \frac{\mu_S(x) \mu_T(y)}{d/n} \right| &\leq \sum_{(x,y) \in G} \left( \frac{\mu_S(x) \mu_T(y)}{d \langle G\mu_S, \mu_T \rangle} \right) |1 - \langle G\mu_S, \mu_T \rangle| \\ &\leq \lambda_0 \cdot \varepsilon_2. \quad \blacksquare \end{aligned}$$

*Proof of Claim 4.5.3.*

$$\sum_{(x,y) \in G} \left| \frac{\mu_S(x) \mu_T(y)}{d/n} - \frac{1}{n \cdot d} \right| = \left( \frac{n}{d} \right) \sum_{(x,y) \in G} \left| \mu_S(x) \mu_T(y) - \frac{1}{n^2} \right|.$$

Since  $\mu_S(x) = \frac{1}{n} + \mu_S^\perp(x)$  and  $\mu_T(y) = \frac{1}{n} + \mu_T^\perp(y)$ ,

$$\begin{aligned} \left( \frac{n}{d} \right) \sum_{(x,y) \in G} \left| \mu_S(x) \mu_T(y) - \frac{1}{n^2} \right| &= \left( \frac{n}{d} \right) \sum_{(x,y) \in G} \left| \frac{\mu_S^\perp(x)}{n} + \frac{\mu_T^\perp(y)}{n} + \mu_S^\perp(x) \mu_T^\perp(y) \right| \\ (\text{Using triangle inequality}) &\leq \frac{1}{d} \sum_{(x,y) \in G} |\mu_S^\perp(x)| + \frac{1}{d} \sum_{(x,y) \in G} |\mu_T^\perp(y)| \\ &\quad + \left( \frac{n}{d} \right) \sum_{(x,y) \in G} |\mu_S^\perp(x) \mu_T^\perp(y)| \\ &= |\mu_S^\perp|_1 + |\mu_T^\perp|_1 + \left( \frac{n}{d} \right) \sum_{(x,y) \in G} |\mu_S^\perp(x) \mu_T^\perp(y)|, \end{aligned}$$

where the last equality uses the fact that  $G$  is a bi-regular graph. Define  $f_S(x) \equiv |\mu_S^\perp(x)|$  is a vector with the entrywise absolute values of  $\mu_S^\perp$ , and similarly  $f_T$ . Then, the RHS

above equation reduces to

$$\begin{aligned}
\left| \mu_S^\perp \right|_1 + \left| \mu_T^\perp \right|_1 + \left( \frac{n}{d} \right) \sum_{(x,y) \in G} \left| \mu_S^\perp(x) \mu_T^\perp(y) \right| &= \left| \mu_S^\perp \right|_1 + \left| \mu_T^\perp \right|_1 \\
&\quad + \left( \frac{n}{d} \right) \cdot \sum_{(x,y) \in G} f_S(x) f_T(y) \\
&= \left| \mu_S^\perp \right|_1 + \left| \mu_T^\perp \right|_1 + n \langle Gf_S, f_T \rangle \\
\text{(Using (4.5.1))} \quad &\leq 2\varepsilon_1 + n \cdot \langle Gf_S, f_T \rangle. \tag{4.5.3}
\end{aligned}$$

A simple bound for  $n \cdot \langle Gf_S, f_T \rangle$  would be  $n \|G\mu_S^\perp\| \|\mu_T^\perp\|$  by Cauchy-Schwarz inequality. We can use the expansion of  $G$  again to estimate this better. Consider the decomposition  $f_S = \alpha_1 \cdot \mathbf{u} + f_S^\perp$  and  $f_T = \alpha_2 \cdot \mathbf{u} + f_T^\perp$ . It follows that  $\alpha_1 = |f_S|_1$  and  $\alpha_2 = |f_T|_1$ , and hence  $\alpha_1, \alpha_2 \leq \varepsilon_1$  by (4.5.1). Hence,

$$\begin{aligned}
n \cdot \langle Gf_S, f_T \rangle &= \alpha_1 \cdot \alpha_2 + n \cdot \langle Gf_S^\perp, f_T^\perp \rangle \leq \varepsilon_1^2 + n \|Gf_S^\perp\| \|f_T^\perp\| \\
&\leq \varepsilon_1^2 + n \cdot \lambda_0 \cdot \|\mu_S^\perp\| \cdot \|\mu_T^\perp\| \\
\text{(Using (4.5.2))} \quad &\leq \varepsilon_1^2 + \lambda_0 \varepsilon_2.
\end{aligned}$$

Combining this with (4.5.3), we get

$$\sum_{(x,y) \in G} \left| \frac{\mu_S(x) \mu_T(y)}{d/n} - \frac{1}{n \cdot d} \right| \leq 2\varepsilon_1 + \varepsilon_1^2 + \lambda_0 \varepsilon_2. \quad \blacksquare$$

**Acknowledgements** We would like to thank Dana Moshkovitz for several discussions and clarifications regarding the initial counter-example. We would also like to thank Mohammad Bavarian for pointing out that our proof might generalize for general two-prover games, and would like to thank Anup Rao for pointing out subtleties involving parallel repetition for general games. We would like to thank anonymous reviewers for various suggestions to improve the write-up. We also would like to thank Prahladh Harsha, Irit Dinur and Amir Shpilka for many fruitful conversations and comments on the write-up.

#### 4.6 An explicit extractor that does not provide robustness

Let  $H = ((W, X), E_H)$  be any  $(\delta, \varepsilon)$ -extractor. Let us assume that the extractor is left-regular with left-degree  $D$ , and let  $m = |W|$  and  $n = |X|$ . For any  $x \in X$  and  $S \subseteq W$ ,

let  $d_S(x)$  denote the degree of  $x$  in  $S$ . Let us fix one  $S \subset W$  such that  $|S| = \delta|W|$ .

We will transform the graph  $H$  so that the distribution induced by the set  $S$  looks like the counter-example described in [Section 4.4.1](#) in the following two steps by altering the edges in the subgraph  $S \times X$ :

1. First change the degree into  $X$  from  $S$  to be exactly uniform.
2. Next further change the degrees into  $X$  from  $S$  to be like the counterexample

Both these operations can be achieved in a monotone fashion: for every  $x \in X$ , the neighborhood of every vertex is either a superset, or a subset of its neighborhood before each operation.

We will show that moving the edges this way does not perturb the indegree distribution from other large sets by too much, and the resulting graph is a  $(\delta, O(\varepsilon))$  extractor as long as the number of edges we relocate is at most  $O(\varepsilon\delta \cdot mD)$ . This process will preserve the left-regularity of  $H$  but would *not* preserve bi-regularity.

First let us move edges (monotonically) from  $S$  into  $X$  create the uniform distribution on  $X$ . When doing this, the degree of each vertex changes by  $\Delta_S(x) := |d_S(x) - \frac{\delta mD}{n}|$ , where  $d_S(x)$  was the old degree. From the extractor property, we know that:

$$\sum_{x \in X} \Delta_S(x) = \sum_{x \in X} (\delta mD) \left| \frac{d_S(x)}{\sum d_S(x)} - \left( \frac{1}{n} \right) \right| \leq \varepsilon\delta \cdot mD. \quad (4.6.1)$$

Every vertex  $x \in X$  now has degree  $d_{\text{avg}}^S$ . Fix some vertex  $x_1 \in X$ , and relocate from every other  $x \neq x_1$  any set of  $\varepsilon \cdot d_{\text{avg}}^S$  edges to be incident on  $x_1$ . Thus, if  $d'_S(x)$  refers to the new degrees, we have  $d'_S(x_1)$  is  $(1 + \varepsilon n)d_{\text{avg}}^S$  where as  $d'_S(x)$  is  $(1 - \varepsilon)d_{\text{avg}}^S$  for every other  $x \neq x_1$ .

The further change in degrees incurred on any  $x \in X$  is  $\Delta'_S(x) := |d'_S(x) - \frac{\delta mD}{n}|$ . Since we this process only relocates  $O(\varepsilon \cdot d_{\text{avg}}^S |X|)$  edges, we have

$$\sum_{x \in X} \Delta'_S(x) = \sum_{x \in X} |d'_S(x) - d_{\text{avg}}^S| \leq O(n \cdot \varepsilon \cdot d_{\text{avg}}^S) = O(\varepsilon\delta \cdot mD). \quad (4.6.2)$$

Thus, the neighbourhood of any vertex  $x$  has changed additively by at most  $\Delta_S(x) + \Delta'_S(x)$ . Therefore, for any subset  $T \subseteq W$  of size at least  $\delta|W|$ ,

$$\begin{aligned}
\sum_{x \in X} |d'_T(x) - d_{\text{avg}}^T| &\leq \sum_{x \in X} |d_T(x) - d_{\text{avg}}^T| + \sum_{x \in X} |d'_T(x) - d_T(x)| \\
&\leq \varepsilon|T|D + \sum_{x \in X} (\Delta_S(x) + \Delta'_S(x)) \\
&\leq \varepsilon|T|D + O(\varepsilon\delta \cdot mD) \quad (\text{using (4.6.1) and (4.6.2)}) \\
&\leq O(\varepsilon \cdot |T|D).
\end{aligned}$$

Thus, the new graph after relocating edges is still an  $(\delta, O(\varepsilon))$ -extractor. This extractor, induces a distribution similar to the one described in [Section 4.4.1](#) and hence cannot provide robustness.

#### 4.7 Lower bounds on degree of fortifiers

In this section, we will show that an attempt to make a game  $(\delta, \varepsilon)$ -robust by concatenating any left-regular graph with left degree  $D$  fails if  $D \leq o(1/\varepsilon\delta)$ .

**Lemma 4.7.1.** *Let  $H = ((W, X), E_H)$  be a left-regular bipartite graph with left-degree  $D = 1/(c \cdot \varepsilon\delta)$  for some  $c > 0$ , and small enough constants  $\varepsilon, \delta$ . Then, there exists a subset  $S \subseteq W$  with  $|S| \geq \delta|W|$  such that if  $p$  was the distribution on  $X$  induced by the uniform distribution on  $S$  then*

$$\|p - \mathbf{u}\|^2 \geq \frac{\Omega(c\varepsilon)}{|X|}.$$

*Proof.* Let  $d_{\text{avg}} = |W|D/|X|$ . Note that at most  $|X|/2$  vertices  $x$  satisfy  $\deg(x) \geq 2d_{\text{avg}}$ . Further, if there is a set  $S$  of  $|X|/4$  vertices  $x$  that  $\deg(x) < (0.5)d_{\text{avg}}$ , then if  $p$  is the distribution on  $X$  induced by the uniform distribution on  $W$ , then  $\|p - \mathbf{u}\|_1 > 1/4$  which implies that  $\|p - \mathbf{u}\|_2^2 \geq \frac{1}{4|X|}$  by Cauchy-Schwarz.

Otherwise, there exists  $X' \subset X$  such that  $|X'| = c\varepsilon\delta^2|X|$  and for each  $x \in X'$  we have  $(0.5)d_{\text{avg}} < \deg(x) < 2d_{\text{avg}}$ . Consider the set  $S_0$  of all neighbours of  $X'$ . If  $D < 1/(c\varepsilon\delta)$ , we have  $|S_0| \leq 2c\delta^2\varepsilon \cdot |W|D = 2\delta|W|$  which is a very small fraction of  $|W|$  when  $\delta$  is small enough. Consider an arbitrary set  $S_1 \subseteq W$  such that  $|S_1| = \delta m$ ,

with  $S_1 \cap S_0 = \emptyset$ . Let  $S_2 = S_0 \cup S_1$ . Let  $\pi_1, \pi_2$  be the probability distribution on  $X$  induced by  $S_1, S_2$  respectively. Note that  $|S_2| \leq 3\delta|W|$ .

For every  $x \in X'$ , we know that  $\pi_1(x) = 0$  and  $\pi_2(x) = \Omega\left(\frac{1}{\delta|X|}\right)$ . Therefore,

$$\|\pi_1 - \pi_2\|^2 \geq \Omega\left(\frac{c\delta^2\varepsilon|X|}{\delta^2|X|^2}\right) = \frac{\Omega(c\varepsilon)}{|X|}.$$

Since  $\|\pi_1 - \pi_2\| \leq \|\pi_1 - \mathbf{u}\| + \|\pi_2 - \mathbf{u}\|$ , we have that one of the sets  $S_1$  or  $S_2$  shows the validity of the lemma ■

We thus immediately infer the following:

**Corollary 4.7.2.** *For all small enough  $\delta, \varepsilon > 0$ , no left-regular graph  $H = ((W, X), E_H)$  with left-degree  $D = o(1/\varepsilon\delta)$  is an  $(\delta, *, \varepsilon)$ -fortifier.*

Note that any  $(\delta, \varepsilon, \varepsilon)$ -fortifier is in particular an  $(\delta, \varepsilon)$ -extractor, and hence we also have that  $D = \Omega((1/\varepsilon^2) \log(1/\delta))$  [RT00]. We also point out that the construction of Lemma 4.2.5 has left-degree  $D = \tilde{O}(1/\varepsilon^2\delta)$ . The above essentially shows this construction is almost optimal.

## Chapter 5

### Inapproximability of Bi-Covering

#### 5.1 Introduction

In this chapter, we study the BI-COVERING problem - Given a graph  $G(V, E)$ , find two (not necessarily disjoint) sets  $A, B \subseteq V$  such that  $A \cup B = V$  and that every edge  $e \in E$  belongs to either the graph induced by  $A$  or to the graph induced by  $B$ . The goal is to minimize  $\max\{|A|, |B|\}$ .

The problem we study is closely related to the problem of *Channel Allocation* which was studied in [GKSW06]. The Channel Allocation Problem can be described as follows: there is a universe of topics, a fixed number of channels and a set of requests where each request is a subset of topics. The task is to send a subset of topics through each channel such that each request is satisfied by set of topics from one of the channel i.e. for every request there must exist at least one channel such that the set of topics present in that channel is a superset of the set of topics from the request. Of course, one can achieve this task trivially by sending all topics through one channel. But, the optimization version of Channel Allocation Problem asks for a way to satisfy all the request by minimizing the maximum number of topics sent through a channel.

Any *connected* undirected graph  $G(V, E)$  on  $n$  vertices and  $m$  edges along with an integer  $k$  can be viewed as a special case of channel allocation problem - The set of topics is a set of  $n$  vertices, each edge represents a request, where the requested set of topics corresponding to an edge is a pair of its endpoints and the number of channels is  $k$ . If we fix the number of channels to  $k = 2$  then the optimization problem exactly corresponds to the BI-COVERING problem. Specifically, the optimization problem asks for two subsets  $A$  and  $B$  of  $V$  minimizing  $\max\{|A|, |B|\}$  such that  $A \cup B = V$  and every edge is totally contained in a graph induced by either  $A$  or  $B$ .

## 5.2 Results

Getting 2 approximation for BI-COVERING problem is trivial (by setting  $A = B = V$ ). We show that BI-COVERING problem is hard to approximate within any factor strictly less than 2 assuming a strong Unique Games Conjecture (UGC) similar to the one in [BK09] (see Conjecture 5.3.3).

**Theorem 5.2.1.** *Let  $\varepsilon > 0$  be any small constant. Assuming a strong Unique Games Conjecture (Conjecture 5.3.3), given a graph  $G(V, E)$ , it is NP-hard to distinguish between following two cases:*

1.  $G$  has BI-COVERING of size at most  $(1/2 + \varepsilon)|V|$ .
2. Any BI-COVERING of  $G$  has size at least  $(1 - \varepsilon)|V|$ .

*In particular, it is NP-hard (assuming strong UGC) to approximate BI-COVERING within a factor  $2 - \varepsilon$  for every  $\varepsilon > 0$ .*

Given this structural hardness result, we get a  $\frac{3}{2} - \varepsilon$  hardness of BI-COVERING restricted to bipartite graphs by transforming a hard instance from Theorem 5.2.1 into a bipartite graph in a natural way (getting a  $\frac{3}{2}$ -approximation is easy - given a bipartite graph on  $X$  and  $Y$  with  $|X| \geq |Y|$ , one can take arbitrary partition  $X$  into two equal sized parts  $X_1$  and  $X_2$  and set the BI-COVERING to be  $X_1 \cup Y$  and  $X_2 \cup Y$ ).

**Theorem 5.2.2.** *Assuming the strong Unique Games Conjecture, for every  $\varepsilon > 0$ , BI-COVERING is NP-hard to approximate within a factor  $\frac{3}{2} - \varepsilon$  for bi-partite graphs.*

Our Theorem 5.2.1 implies hardness result for the following well known problem:

MAX-BI-CLIQUE problem is as follows:

**Input:** A bipartite graph  $G(X, Y, E)$  with  $|X| = |Y| = n$ .

**Output:** Find largest  $k$  such that there exists two subsets  $A \subseteq X, B \subseteq Y$  of size  $k$  and the graph induced on  $(A, B)$  is a complete bipartite graph.

Inapproximability of MAX-BI-CLIQUE problem has been studied extensively [AFWZ95, BS02, Fei02, Kho06]. Feige[Fei02] showed that using an assumption of average case

hardness of 3SAT instance, MAX-BI-CLIQUE cannot be approximated within any constant factor in polynomial time (and hence within  $n^\delta$  for some  $\delta > 0$  using known amplification technique [AFWZ95, BS02]). Feige-Kogan [FK04] showed that assuming  $SAT \notin DTIME(2^{n^{3/4+\varepsilon}})$  there is no  $2^{(\log n)^\delta}$  approximation for MAX-BI-CLIQUE. They also showed that it is NP-hard to approximate MAX-BI-CLIQUE within any constant factor assuming MAX-CLIQUE (finding a maximum sized clique in a graph) does not have a  $n/2^{c\sqrt{\log n}}$ -approximation. Khot [Kho06] later proved a similar inapproximability result but assuming  $NP \not\subseteq \cap_{\varepsilon>0} BPTIME(2^{n^\varepsilon})$  using a *quasi-random PCP*. It is an important open problem to extend similar hardness results based on weaker complexity assumptions [AMS11]. In particular, it is still not known if UGC implies a constant factor hardness for MAX-BI-CLIQUE. A straightforward corollary from Theorem 5.2.1 (see 5.5) implies that we get similar hardness results for MAX-BI-CLIQUE based on Conjecture 5.3.3.

**Corollary 5.2.3.** *Assuming strong Unique Games Conjecture, it is NP-hard to approximate MAX-BI-CLIQUE within any constant factor.*

As mentioned above, the hardness factor can be boosted to  $n^\delta$  for some  $\delta > 0$  using known techniques. (such as described in [AFWZ95, BS02])

### UGC and strong UGC:

Unique games conjecture so far helped in understanding the tight inapproximability factors of many problems including, but not limited to, Vertex Cover [KR08], optimal algorithm for every Max-CSP [Rag08], Ordering CSPs [GHM<sup>+</sup>11], characterizing strong approximation resistance of CSPs [KTW14] etc. The inherent difficulty in showing hardness results assuming UNIQUE GAMES CONJECTURE for the problems that we study is that we need some kind of expansion property on the unique games instance which it lacks. It is shown that Unique Games are easy when the constraint graph is an expander [AKK<sup>+</sup>08]. In general, in [ABS10] it is shown that Unique Games are easy when a normalized adjacency matrix of a constraint graph has very few eigenvalues close to 1. So the natural direction is to modify the unique games instance to get some

expansion property but weak enough so that it is not tractable by the techniques of [AKK<sup>+</sup>08], [ABS10]. A similar STRONG UNIQUE GAMES CONJECTURE, which has a *weak expansion property*, has been used earlier in [BK09] and [Sve10] to show inapproximability results for minimizing weighted completion time on a single machine with precedence constraints and minimizing makespan in precedence constrained scheduling on identical machines respectively. Our result adds another interesting implication of UNIQUE GAMES CONJECTURE with weak expansion property, namely inapproximability of MAX-BI-CLIQUE and BI-COVERING. We hope that our results will help motivate study of STRONG UNIQUE GAMES CONJECTURE and ultimately answering the question about its equivalence to the UNIQUE GAMES CONJECTURE.

### 5.3 Preliminaries

Let  $q$  be any prime for convenience. We are interested in space of functions from  $\mathbb{F}_q^n$  to  $\mathbb{R}$ . Define inner product on this space as  $\langle f, g \rangle = \frac{1}{q^n} \sum_{x \in \mathbb{F}_q^n} f(x)g(x)$ . Let  $\omega_q$  be the  $q^{th}$  root of unity. For a vector  $\alpha \in \mathbb{F}_q^n$ , we will denote  $\alpha_i$  the  $i^{th}$  coordinate of vector  $\alpha$ . The *Fourier decomposition* of a function  $f : \mathbb{F}_q^n \rightarrow \mathbb{R}$  is given as

$$f(x) = \sum_{\alpha \in \mathbb{F}_q^n} \hat{f}(\alpha) \chi_\alpha(x)$$

where  $\chi_\alpha(x) := \omega_q^{\langle \alpha, x \rangle}$  and a *Fourier coefficient*  $\hat{f}(\alpha) := \langle f, \chi_\alpha \rangle$ .

Our hardness result is based on a variant of Unique Games conjecture. First, we define what the Unique game is:

**Definition 5.3.1** (UNIQUE-GAMES). *An instance  $G = (U, V, E, [L], \{\pi_e\}_{e \in E})$  of the UNIQUE-GAMES constraint satisfaction problem consists of a bi-regular bipartite graph  $(U, V, E)$ , a set of alphabets  $[L]$  and a permutation map  $\pi_e : [L] \rightarrow [L]$  for every edge  $e \in E$ . Given a labeling  $\ell : U \cup V \rightarrow [L]$ , an edge  $e = (u, v)$  is said to be satisfied by  $\ell$  if  $\pi_e(\ell(v)) = \ell(u)$ .*

$G$  is said to be at most  $\delta$ -satisfiable if every labeling satisfies at most a  $\delta$  fraction of the edges.

The following is a conjecture by Khot [Kho02a] which has been used to prove many *tight* inapproximability results.

**Conjecture 5.3.2** (UNIQUE GAMES CONJECTURE[Kho02a]). *For every sufficiently small  $\delta > 0$  there exists  $L \in \mathbb{N}$  such that the following holds. Given an instance  $\mathcal{G} = (U, V, E, [L], \{\pi_e\}_{e \in E})$  of UNIQUE-GAMES it is NP-hard to distinguish between the following two cases:*

- *YES case: There exist an assignment that satisfies at least  $(1 - \delta)$  fraction of the edges.*
- *NO case: Every assignment satisfies at most  $\delta$  fraction of the edge constraints.*

Our hardness results are based on the following stronger conjecture which is similar to the one in Bansal-Khot [BK09]. We refer readers to [BK09] for more discussion on comparison between these two conjectures.

**Conjecture 5.3.3** (STRONG UNIQUE GAMES CONJECTURE). *For every sufficiently small  $\delta, \gamma, \eta > 0$  there exists  $L \in \mathbb{N}$  such that the following holds: Given an instance  $\mathcal{G} = (U, V, E, [L], \{\pi_e\}_{e \in E})$  of UNIQUE-GAMES which is bi-regular, it is NP-hard to distinguish between the following two cases:*

- *YES case: There exist sets  $V' \subseteq V$  such that  $|V'| \geq (1 - \eta)|V|$  and an assignment that satisfies all edges connected to  $V'$ .*
- *NO case: Every assignment satisfies at most  $\gamma$  fraction of the edge constraints. Moreover, the instance satisfies the following expansion property. For every set  $S \subseteq V$ ,  $|S| = \delta|V|$ , we have  $|\Gamma(S)| \geq (1 - \delta)|U|$ , where  $\Gamma(S) := \{u \in U \mid \exists v \in S \text{ s.t. } (u, v) \in E\}$ .*

**Remark 5.3.4.** *We would like to point out that the above conjecture differs from the one in [BK09] in the completeness case. In [BK09], the Yes instance has a guarantee that there exists sets  $V' \subseteq V, U' \subseteq U$  with  $|V'| \geq (1 - \eta)|V|, |U'| \geq (1 - \eta)|U|$  such that all edges between  $V'$  and  $U'$  are satisfied.*

The BI-COVERING problem is:

**Input:** A graph  $G(V, E)$

**Output:** Two subsets  $A, B \subseteq V$  such that  $A \cup B = V$  and every edge  $(u, v) \in E$  either  $\{u, v\} \subseteq A$  or  $\{u, v\} \subseteq B$ . Minimize  $\max\{|A|, |B|\}$ .

The optimal value of a BI-COVERING on instance  $G(V, E)$  is always at least  $|V|/2$  and hence getting a 2-approximation for this problem is *trivial* by setting  $A = V$  and  $B = \emptyset$ . In order to beat 2-approximation, one should be able to solve the following weaker problem.

### Problem

For small enough  $\varepsilon > 0$ , given an undirected graph  $G(V, E)$ , distinguish between the following two cases:

1. There exists two disjoint sets  $A, B \subseteq V$ ,  $|A|, |B| \geq (1/2 - \varepsilon)|V|$  such that there are no edges between  $A$  and  $B$ .
2. There exists no two disjoint sets  $A, B \subseteq V$   $|A|, |B| \geq \varepsilon|V|$  such that there are no edges between  $A$  and  $B$ .

In this section, we show that it is UG-Hard to distinguish between (1) and (2) for any constant  $\varepsilon > 0$  proving [Theorem 5.2.1](#).

## 5.4 Dictatorship Test

In order to prove the  $(2 - \varepsilon)$  hardness, we first start with a *dictatorship test* that we will use as a gadget in the actual reduction.

We design a dictatorship test for the problem BI-COVERING. We are interested in functions  $f : \mathbb{F}_q^n \rightarrow \mathbb{R}$ .  $f$  is called a dictator if it is of the form  $f(x_1, x_2, \dots, x_n) = x_i$  for some  $i \in [n]$ .

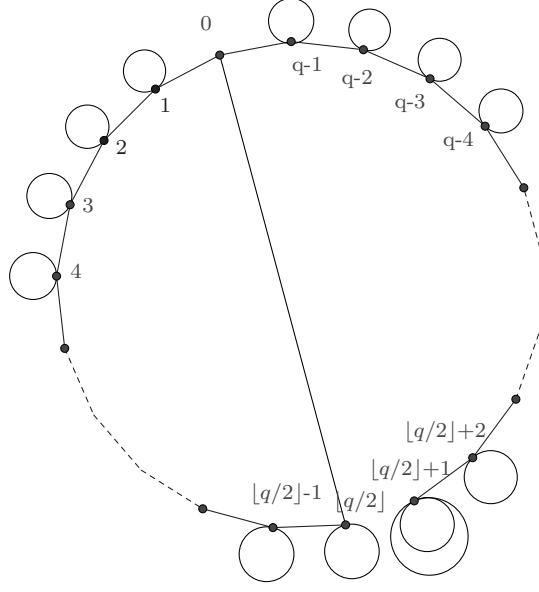


Figure 5.1: Dictatorship Gadget

**Dictatorship gadget:**

For convenience, we will let  $q > 2$  be any prime number for the description of the dictatorship gadget. Let  $G(\mathbb{F}_q, \mathcal{E})$  be a 3-regular graph on  $\mathbb{F}_q$  (where we identify the elements of  $\mathbb{F}_q$  by  $\{0, 1, \dots, q-1\}$ ) with self loops as shown in Figure 5.1:

It is constructed as follows : Take a cycle on  $0, 1, 2, \dots, q-1, 0$ , then add a self loop to every vertex except to the vertex 0. Remove the edge  $(\lfloor q/2 \rfloor, \lfloor q/2 \rfloor + 1)$ , add an edge  $(0, \lfloor q/2 \rfloor)$ . Finally, to make it 3-regular, add a self loop to the vertex  $\lfloor q/2 \rfloor + 1$ . This completes the description of graph  $G$ . Since the graph  $G$  is connected and non-bipartite, the symmetric Markov operator  $T$  defined by the random walk in  $G$  has  $r(T) < 1$ . One crucial thing about  $G$  is that it has two large disjoint subsets of vertices, namely  $\{1, 2, \dots, \lfloor q/2 \rfloor\}$  and  $\{\lfloor q/2 \rfloor + 1, \lfloor q/2 \rfloor + 2, \dots, q-1\}$ , with no edges in between.

Consider the vertex set  $V = \mathbb{F}_q^R$  for some constant  $R$ . We will construct a graph  $H$  on  $V$  as follows :  $(x, y) \in (\mathbb{F}_q^R)^2$  forms an edge in  $H$  iff they satisfy the following condition:

$$\forall i \in [R], (x_i, y_i) \in \mathcal{E},$$

$x$  is adjacent to  $y$  iff  $T^{\otimes R}(x \leftrightarrow y) \neq 0$ .

**Completeness:**

Let  $f : \mathbb{F}_q^R \rightarrow \mathbb{R}$  be any dictator, say  $i^{th}$  dictator i.e.  $f(x) = x_i$ . By letting set  $A$  to be  $f^{-1}(0) \cup f^{-1}(1) \cup \dots \cup f^{-1}(\lfloor q/2 \rfloor)$  and set  $B$  to be  $f^{-1}(0) \cup f^{-1}(\lfloor q/2 \rfloor + 1) \cup f^{-1}(\lfloor q/2 \rfloor + 2) \cup \dots \cup f^{-1}(q-1)$ , it can be seen easily that there is no edge between sets  $A \setminus B$  and  $B \setminus A$ . More precisely,

$$\begin{aligned} A \setminus B &= \{x \in \mathbb{F}_q^R \mid x_i \in \{1, 2, \dots, \lfloor q/2 \rfloor\}\} \\ B \setminus A &= \{y \in \mathbb{F}_q^R \mid y_i \in \{\lfloor q/2 \rfloor + 1, \lfloor q/2 \rfloor + 2, \dots, q-1\}\} \end{aligned}$$

By the property of Markov operator  $T^{\otimes R}$ ,  $(x, y)$  are not adjacent if  $(x_i, y_i) \notin \mathcal{E}$  for some  $i \in [R]$ . Hence, there are no edges between  $A \setminus B$  and  $B \setminus A$ . Thus, the optimal value is at most

$$\frac{1}{|V|} \cdot \max\{|A|, |B|\} = \frac{1}{2} + \frac{1}{2q}.$$

**Soundness:**

Let  $A, B \subseteq V$  such that  $A \cup B = V$  and  $f, g : \mathbb{F}_q^R \rightarrow \{0, 1\}$  be the indicator functions of sets  $A \setminus B$  and  $B \setminus A$  respectively. Suppose  $|A \setminus B| = \varepsilon|V|$  and  $|B \setminus A| = \varepsilon|V|$  for some  $\varepsilon > 0$  and that there are no edges in between  $A \setminus B$  and  $B \setminus A$ . We will show that in this case,  $f$  and  $g$  must have a common influential co-ordinate. Since, there are no edges between these sets, we have

$$\mathbf{E}_{\substack{x \sim \mathbb{F}_q^R, \\ y \sim T^{\otimes R}(x)}} [f(x)g(y)] = \langle f, T^{\otimes R}g \rangle = 0$$

For the application of Invariance principle, [Theorem 2.4.6](#), in our case we have  $\mathbf{E}[f] = \mathbf{E}[g] = \varepsilon > 0$  and  $\rho = r(T) < 1$ . Thus, for small enough  $\tau := \tau(\rho, \varepsilon) > 0$ ,

$$\underline{\Gamma}_\rho(\varepsilon, \varepsilon) - \tau > 0.$$

We can now apply [Theorem 2.4.6](#) to conclude that there exists  $i \in [R]$  and  $k \in \mathbb{N}$  independent of  $R$  such that

$$\min(\mathbf{Inf}_i^{\leq k}(f), \mathbf{Inf}_i^{\leq k}(g)) \geq \delta,$$

for some  $\delta(\tau) > 0$ . Hence, unless  $f$  and  $g$  have a common influential co-ordinate,  $\frac{1}{|V|} \cdot \max\{|A|, |B|\} \geq 1 - \varepsilon$ . Thus, the optimum value is at least  $1 - \varepsilon$

## 5.5 $(2 - \varepsilon)$ - inapproximability

The above dictatorship test for large enough  $q$  can be composed with the Unique Games instance having some stronger guarantee ([Conjecture 5.3.3](#)) in a straightforward way that gives  $(2 - \varepsilon)$  hardness for every constant  $\varepsilon > 0$  assuming UGC. Details as follows:

Let  $\mathcal{G} = (U, V, E, [L], \{\pi_e\}_{e \in E})$  be the given instance of UNIQUE-GAMES with parameters  $\delta < \frac{\varepsilon}{4}, \gamma, \eta > 0$  from [Conjecture 5.3.3](#). We replace each vertex  $v \in V$  by a block of  $q^L$  vertices, namely by a hypercube  $[q]^L$ . We will denote this block by  $[v]$ . As defined in the dictatorship test, let  $G$  be the graph on  $\mathbb{F}_q$  and  $T$  be the induced symmetric Markov operator. For every pair of edges  $e_1(u, v_1)$  and  $e_2(u, v_2)$  in  $\mathcal{G}$ , we will add the following edges between  $[v_1]$  and  $[v_2]$ : Let  $\pi_1$  and  $\pi_2$  be the permutation constraint associated with  $e_1$  and  $e_2$  respectively.  $x \in [v_1]$  and  $y \in [v_2]$  are connected by an edge iff  $T^{\otimes L}((x \circ \pi_1^{-1}) \leftrightarrow (y \circ \pi_2^{-1})) \neq 0$  (where  $(x \circ \pi^{-1})_i = x_{\pi^{-1}(i)}$  for all  $i \in [L]$ ) i.e. for every  $i \in [L]$ ,  $x_{\pi_1^{-1}(i)}$  and  $y_{\pi_2^{-1}(i)}$  are connected by an edge in graph  $G$ . This completes the description of a graph. Let's denote this graph by  $H$ .

**Lemma 5.5.1** (Completeness). *If there exists an assignment to vertices in  $\mathcal{G}$  that satisfies all edges connected to  $(1 - \eta)$  fraction of vertices in  $V$  then  $H$  has a Bi-COVERING of size at most  $(1 - \eta)(1/2 + 1/2q) + \eta$ .*

*Proof.* Fix a labeling  $\ell$  such that for at least  $(1 - \eta)$  fraction of vertices in  $V$  in  $\mathcal{G}$ , all edges attached to them are satisfied. Suppose  $X$  be the set of remaining  $\eta$  fraction of

vertices of  $V$  in  $\mathcal{G}$ . For every vertex  $v \in V$ , consider the following two partitions of  $[v]$ :

$$\begin{aligned} A_v &= \{x \in [q]^L : x_{\ell(v)} \in \{1, \dots, \lfloor q/2 \rfloor\}\} \\ B_v &= \{x \in [q]^L : x_{\ell(v)} \in \{\lfloor q/2 \rfloor + 1, \lfloor q/2 \rfloor + 2, \dots, q\}\} \\ C_v &= \{x \in [q]^L : x_{\ell(v)} = 0\} \end{aligned}$$

Let  $A = \cup_{v \in V} (A_v \cup C_v) \cup_{z \in X} [z]$  and  $B = \cup_{v \in V} (B_v \cup C_v) \cup_{z \in X} [z]$ . The claim is that this is the required edge separating sets. To see this, consider any vertex pair  $(a, b)$  such that  $a \in A \setminus B$  and  $b \in B \setminus A$ . We need to show that  $(a, b)$  must not be adjacent in  $H$ . Suppose  $a \in [v_1]$  and  $b \in [v_2]$ . If  $v_1$  and  $v_2$  don't have a common neighbor then clearly, there is no edge between  $a$  and  $b$ . Suppose they have a common neighbor  $u$  and let  $e_1 = (u, v_1)$  and  $e_2 = (u, v_2)$  be the edges and  $\pi_1$  and  $\pi_2$  be the associated permutation constraints. Since  $X \subseteq A \cap B$ ,  $v_1, v_2 \notin X$ . Hence  $\ell$  satisfies all constraints associated with  $v_1$  and  $v_2$ . In particular,  $\pi_1(\ell(v_1)) = \pi_2(\ell(v_2)) =: j$  for some  $j \in [L]$ . Since  $a \in A_{v_1}$ , we have  $a_{\pi_1^{-1}(j)} = a_{\ell(v_1)} \in \{1, \dots, \lfloor q/2 \rfloor\}$ . Similarly,  $b_{\pi_2^{-1}(j)} \in \{\lfloor q/2 \rfloor + 1, \lfloor q/2 \rfloor + 2, \dots, q\}$ . By the construction of edges in  $H$ ,  $a$  and  $b$  are not adjacent.

For any  $v$ ,  $|A_v \cup C_v| = |B_v \cup C_v| = (\frac{1}{2} + \frac{1}{2q})q^L$ . Thus,

$$|A| = |B| \leq \left( \eta + (1 - \eta) \left( \frac{1}{2} + \frac{1}{2q} \right) \right) |V| q^L$$

■

**Lemma 5.5.2** (Soundness). *For every constant  $\varepsilon > 0$ , there exists a constant  $\gamma$  such that, if  $\mathcal{G}$  is at most  $\gamma$ -satisfiable then  $H$  has BI-COVERING of size at least  $1 - \varepsilon$ .*

*Proof.* Suppose for contradiction, there exists an BI-COVERING of size at most  $(1 - \varepsilon)$ . This means there exists two disjoint sets  $X, Y$  of size at least  $\varepsilon$  fraction of vertices in  $H$  such that there are no edges in between  $X$  and  $Y$ . Let  $X^*$  be the set of vertices in  $v \in V$  such that  $[v] \cap X \geq \frac{\varepsilon}{2} |[v]|$ . Similarly,  $Y^*$  be the set of vertices in  $v \in V$  such that  $[v] \cap Y \geq \frac{\varepsilon}{2} |[v]|$ . By simple averaging argument,  $|X^*| \geq \frac{\varepsilon}{2} |V|$  and  $|Y^*| \geq \frac{\varepsilon}{2} |V|$ .

**Lemma 5.5.3.** *The total fraction of edges connected to  $X^*$  whose other end point is in  $\Gamma(X^*) \cap \Gamma(Y^*)$  is at least  $\frac{1}{2}$ .*

*Proof.* Let  $\mathcal{G}$  has left-degree  $d_1$  and right-degree  $d_2$ . We have  $d_1 = \frac{d_2|V|}{|U|}$ . Suppose the claim is not true, then at least  $\frac{1}{2}$  fraction of edges have their endpoint in  $U \setminus \Gamma(Y^*)$ . As,  $|U \setminus \Gamma(Y^*)| \leq \delta|U|$ , the average degree of a vertex in  $U \setminus \Gamma(Y^*)$  is at least  $\frac{(1/2)d_2|X^*|}{\delta|U|} \geq \frac{(d_2/2) \cdot (\varepsilon/2)|V|}{\delta|U|}$  which is greater than  $d_1$  as  $\varepsilon > 4\delta$ . ■

For  $v \in X^* \cup Y^*$ , let  $f_v : [q]^L \rightarrow \{0, 1\}$  be the indicator function of a set  $[v] \cap (X \cup Y)$ . Define the following label set for  $v \in X^* \cup Y^*$  for some  $\tau' > 0$  and  $k \in \mathbb{N}$ :

$$\mathcal{F}(v) := \{i \in [L] \mid \mathbf{Inf}_i^{\leq k}(f_v) \geq \tau'\}$$

We have  $|\mathcal{F}(v)| \leq \frac{\tau'}{k}$  as  $\sum_i \mathbf{Inf}_i^{\leq k}(f_v) \leq k$ .

**Lemma 5.5.4.** *There exists a constant  $\tau' := \tau'(q, \varepsilon)$  and  $k := k(q, \varepsilon)$  such that for every  $u \in U$  and edges  $e_1(u, v), e_2(u, w)$  such that  $v \in X^*$  and  $w \in Y^*$ , we have*

$$\pi_{e_1}(\mathcal{F}(v)) \cap \pi_{e_2}(\mathcal{F}(w)) \neq \emptyset$$

*Proof.* As there are no edges between  $X$  and  $Y$ , we have

$$\mathbf{E}_{\substack{(x \circ \pi_{e_1}^{-1}) \sim \mathbb{F}_q^L, \\ (y \circ \pi_{e_2}^{-1}) \sim T^{\otimes L}(x \circ \pi_{e_1}^{-1})}} [f_v(x \circ \pi_{e_1}^{-1}) f_w(y \circ \pi_{e_2}^{-1})] = 0$$

By the soundness analysis of the dictatorship test, it follows that there exists  $i \in [L]$  such that

$$\min(\mathbf{Inf}_{\pi_{e_1}^{-1}(i)}^{\leq k}(f_v), \mathbf{Inf}_{\pi_{e_2}^{-1}(i)}^{\leq k}(f_w)) \geq \tau',$$

for some  $\tau', k$  as a function of  $q$  and  $\varepsilon$ . Thus,  $i \in \pi_{e_1}(\mathcal{F}(v))$  and  $i \in \pi_{e_2}(\mathcal{F}(w))$ . ■

■

### Labeling:

Fix  $\tau'$  and  $k$  from Lemma 5.5.4. We now define a labeling  $\ell$  to vertices in  $X^* \subseteq V$  and in  $\Gamma(X^*) \cap \Gamma(Y^*) \subseteq U$  as follows: For a vertex  $v \in X^*$  set  $\ell(v)$  to be a uniformly random label from  $\mathcal{F}(v)$ . For  $u \in \Gamma(X^*) \cap \Gamma(Y^*)$ , select an arbitrary neighbor  $w$  of  $u$  in  $Y^*$  and set  $\ell(u)$  to be a uniformly random label from the set  $\pi_{(u,w)}(\mathcal{F}(w))$  of size at most  $\frac{k}{\tau'}$ . Fix an edge  $(u, v)$  such that  $u \in \Gamma(X^*) \cap \Gamma(Y^*)$  and  $v \in X^*$ . By Lemma 5.5.4,

for any  $w \in Y^*$  since  $\pi_{(u,w)}(\mathcal{F}(w)) \cap \pi_{(u,v)}(\mathcal{F}(v)) \neq \emptyset$ , The probability that the edge is satisfied by the randomized labeling is at least  $\left(\frac{\tau'}{k}\right)^2$ . Thus in expectation, at least  $\left(\frac{\tau'}{k}\right)^2$  fraction of edges between  $X^*$  and  $\Gamma(X^*) \cap \Gamma(Y^*)$  are satisfied. By [Lemma 5.5.3](#), at least  $\frac{1}{2}$  fraction of edges connected to  $X^*$  are in between  $X^*$  and  $\Gamma(X^*) \cap \Gamma(Y^*)$ . Finally using bi-regularity, this labeling satisfies at least  $\frac{1}{2} \frac{\varepsilon}{2} \left(\frac{\tau'}{k}\right)^2$  fraction of edges in  $\mathcal{G}$ . Setting  $\gamma < \frac{1}{2} \frac{\varepsilon}{2} \left(\frac{\tau'}{k}\right)^2$  completes the proof.  $\blacksquare$

### Proof of [Theorem 5.2.1](#):

The proof follows from [Lemma 5.5.1](#), [Lemma 5.5.2](#) and [Conjecture 5.3.3](#).

### Proof of [Theorem 5.2.2](#):

Given an input as a bipartite graph, there is a trivial  $3/2$  approximation for BI-COVERING - Take set  $A$  to be the union of a smaller part and half of the larger bi partition and  $B$  to be union of smaller part and remaining half of the larger part. It is easy to see these two sets  $A$  and  $B$  satisfy the property of being a BI-COVERING. As  $\max\{|A|, |B|\} \leq \frac{3}{4}|V|$ , this is a  $\frac{3}{2}$  approximation as OPT is at least  $\frac{|V|}{2}$ .

The  $\frac{3}{2} + \varepsilon$  inapproximability follows easily from the above  $(2 - \varepsilon)$  inapproximability for the general case. The reduction is as follows: Let  $G(V, E)$  be the given instance of a BI-COVERING. Construct a natural bipartite graph  $G'$  between  $V \times V$  where  $(i, j)$  forms an edge if  $(i, j) \in E$  (or  $(j, i) \in E$ ). Fix a small enough constant  $\varepsilon > 0$ . It is easy to see that if  $G$  has a solution of fractional size  $1/2 + \varepsilon$  then so does  $G'$ . Next, if there are sets  $A'$  and  $B'$  where  $\frac{1}{2|V|} \max\{|A'|, |B'|\} \leq \frac{3}{4} - \varepsilon$  which satisfy the BI-COVERING property, we have  $\frac{1}{2|V|}|A' \setminus B'| = \frac{1}{2|V|}(2|V| - |B'|) \geq 1 - (\frac{3}{4} - \varepsilon) = \frac{1}{4} + \varepsilon$  and similarly  $\frac{1}{2|V|}|B' \setminus A'| \geq \frac{1}{4} + \varepsilon$ . Note that  $A' \setminus B'$  and  $B' \setminus A'$  are two disjoint sets whose size of union is at least  $(1 + 2\varepsilon)|V|$ . Thus, we can find two sets, say  $X'$  and  $Y'$  (namely  $X'$  is intersection of  $A' \setminus B'$  with left part of the bipartite graph and  $Y'$  is the intersection of  $B' \setminus A'$  with right part) of size at least  $\varepsilon|V|$  each, where  $X'$  is from left side and  $Y'$  is from right side with no edges in between. We now think of  $X'$  and  $Y'$  as a subset of  $V$ . Let  $Z = X' \cap Y'$ . Partition  $Z$  into  $Z_1$  and  $Z_2$  of equal sizes. Take  $X = Z_1 \cup (X' \setminus Y')$  and  $Y = Z_2 \cup (Y' \setminus X')$ . It is now easy to verify that there are

no edges in between  $X$  and  $Y$  in  $G$  and  $\frac{1}{|V|} \min\{|X|, |Y|\} \geq \frac{\varepsilon}{2}$ . Hence, if we can find a solution of fractional cost  $\frac{3}{4} - \varepsilon$  in  $G'$  in polynomial time then we can also find a solution of fractional cost  $1 - \frac{\varepsilon}{2}$  in  $G$  in polynomial time and this gives a polynomial time algorithm with approximation factor  $2 - \frac{\varepsilon}{2}$  for small enough constant  $\varepsilon > 0$ . As BI-COVERING is UG hard to approximate within  $(2 - \varepsilon)$  for all  $\varepsilon > 0$  for general graph, this gives a  $\frac{3}{2} + \varepsilon$  hardness for BI-COVERING in bipartite graph.

**Proof of Corollary 5.2.3:**

We prove it by giving reduction from BI-COVERING. Let  $G(V, E)$  be the given instance of BI-COVERING. Construct a bipartite graph  $H$  between  $V \times V$  where  $(i, j)$  forms an edge if  $(i, j) \notin E$ . Fix a small enough constant  $\varepsilon > 0$ . In one direction, if  $G$  has a BI-COVERING of fractional size at most  $(1/2 + \varepsilon)$  then  $H'$  contains a  $(1/2 - \varepsilon)|V| \times (1/2 - \varepsilon)|V|$  bipartite clique. In other direction, if  $H'$  has a bipartite clique of size  $2\varepsilon|V| \times 2\varepsilon|V|$  then let  $X'$  and  $Y'$  be the subset of vertices from left and right side of bipartite clique. As before, let  $Z = X' \cap Y'$  and  $Z_1$  and  $Z_2$  be the partition of  $Z$  of equal size. Let  $X = (X' \setminus Y') \cup Z_1$  and  $Y = (Y' \setminus X') \cup Z_2$ . It follows that  $|X|, |Y|$  is at least  $\varepsilon|V|$  and are disjoint viewed as a subset of  $V$ . Also, there are no edges between  $X$  and  $Y$ . Therefore,  $V \setminus X$  and  $V \setminus Y$  each of size at most  $(1 - \varepsilon)|V|$  gives a BI-COVERING of  $G$ . Thus, Theorem 5.2.1 implies that it is hard to distinguish between BI-CLIQUE of size  $(1/2 - \varepsilon)|V|$  and  $\varepsilon|V|$  which completes the proof of corollary.

## Chapter 6

### Low Degree Test

#### 6.1 Introduction

In this chapter, we present cube vs cube low degree test. Low degree tests are local tests for the property of being a low degree function. These were the first property testing results that were discovered, and are an important component in PCP constructions. Such tests were studied in the 1990's and their ballpark soundness behavior was more or less understood. In this work we revisit these tests and give a new and arguably simpler analysis for the cube vs. cube low degree test. Our proof method allows us to get a soundness guarantee that is much closer to the conjectured optimal value. Discovering the precise point in which soundness starts to hold is an intriguing open question that captures an interesting aspect of local-testing in the small soundness regime.

Let us begin with a short introduction to low degree tests. A low degree test can be described as a game between a prover and a verifier, in which the prover wants to convince the verifier that a function  $f : \mathbb{F}^m \rightarrow \mathbb{F}$  is a low degree polynomial. The most straightforward way for the prover to specify  $f$  would be to give its value on each point  $x \in \mathbb{F}^m$ . However, in this way, to check that  $f$  has degree at most  $d$  the verifier would have to read  $f$  on at least  $d + 2$  points. If we want a verifier that makes fewer queries while keeping the error small, it is useful to move to a more redundant representation of  $f$ . For example, the verifier can ask the prover to specify for every cube (affine subspace of dimension 3)  $C \subset \mathbb{F}^m$ , a function  $f_C : C \rightarrow \mathbb{F}$  that is defined on the cube and is obtained by restricting  $f$  to that cube. This is called a “cubes-table”, and similarly one can consider a lines table (with an entry for every line), or a planes table (with an entry for each plane).

Thus, in the cubes representation of a low degree function  $f : \mathbb{F}^m \rightarrow \mathbb{F}$ , we have

a table entry  $T(C)$  for every cube  $C$  and the value of that entry is supposed to be  $T(C) = f|_C$ . A general cubes table is a table  $T(\cdot)$  indexed by all possible cubes and the  $C$ -th entry is a low degree function on the cube  $C$ . Each  $T(C)$  is viewed as a local function. Indeed the number of bits needed to specify  $T(C)$  is only  $O(d^3 \log |\mathbb{F}|)$  which is much smaller than  $\binom{m+d}{d} \log |\mathbb{F}|$  - the number of bits needed to represent a general degree  $d$  function  $f$  on  $\mathbb{F}^m$ .

The prover may cheat, as provers do, by giving a cubes table whose entries cannot be “glued together” into any one global low degree function. This is where the *agreement* test comes in. The verifier can check the table by reading two entries corresponding to two cubes that have a non-trivial intersection, and checking that the function  $T(C_1)$  and the function  $T(C_2)$  agree on points in the intersection of  $C_1 \cap C_2$ .

---

**Test 1** Cube vs. Cube agreement test.

---

1. Select a point  $x \in \mathbb{F}^m$ .
2. Pick affine cubes  $C_1, C_2$  randomly conditioned on  $C_1, C_2 \ni x$ .
3. Read  $T(C_1), T(C_2)$  from the table and accept iff  $T(C_1)(x) = T(C_2)(x)$ .

Let  $\alpha_{\mathcal{C}x\mathcal{C}}(T)$  be the *agreement* of the table  $T$ , i.e. the probability of acceptance of the test.

---

The test is local in that it accesses only two cubes. Different tests may differ in the distribution underlying the agreement test (for example, Raz and Safra look at two planes that intersect in a line, which clearly is a different distribution from choosing two planes that intersect in a point), but they all check agreement on the intersection, so we generally refer to all of these as agreement tests.

The interesting point, as proven by both Raz and Safra in [RS97], and by Arora and Sudan in [AS97], is that such tests have small soundness error. For example, the plane vs. plane theorem of Raz Safra is as follows,

**Theorem 6.1.1** (Raz-Safra [RS97]). *There is some  $\delta > 0$  such that for every  $d$  and prime power  $q$  and every  $m \geq 3$  the following holds. Let  $\mathbb{F}$  be a finite field  $|\mathbb{F}| = q$ , and let  $T(\cdot)$  be a planes table, assigning to each plane  $P \subset \mathbb{F}^m$  a bivariate degree  $d$*

polynomial  $T(P) : P \rightarrow \mathbb{F}$ . Let  $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$  be as defined in [Test 2](#).

For every  $\varepsilon \geq (md/q)^\delta$ , if  $\alpha_{\mathcal{P}\ell\mathcal{P}}(T) \geq \varepsilon$  then there is a degree  $d$  function  $g : \mathbb{F}^m \rightarrow \mathbb{F}$  such that  $T(P) = g|_P$  on an  $\Omega(\varepsilon)$  fraction of the planes.

---

**Test 2** The Raz-Safra Plane vs. Plane agreement test.

---

1. Select an affine line  $\ell \subset \mathbb{F}^m$ .
2. Choose affine planes  $P_1, P_2$  randomly conditioned on  $P_1, P_2 \supset \ell$ .
3. Read  $T(P_1), T(P_2)$  from the table and accept iff  $T(P_1)(x) = T(P_2)(x)$  for all  $x \in \ell$ .

Let  $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$  be the *agreement* of the table  $T$ , i.e. the probability of acceptance of the test.

---

A similar theorem was proven by Arora and Sudan for  $T$  a lines table and for a natural test that checks if two intersecting lines agree on the point of intersection.

These results are called low degree tests although it makes sense to think of them as theorems relating local agreement to global agreement. We refer to them as low degree *agreement* test theorems.

### Towards the soundness threshold.

The most important aspect of the low degree agreement theorems of [\[RS97, AS97\]](#) is the fact that they have small soundness. Small soundness means that a cheating prover won't be able to fool the verifier into accepting with even a tiny  $\varepsilon > 0$  probability, unless the table has some non-trivial agreement with a global low degree function. Small soundness of low degree tests was used inside PCP constructions for getting PCPs with the smallest known soundness error. The fact that soundness holds for all values of  $\varepsilon \geq (d/q)^\delta$  was sufficient for the PCP constructions of [\[RS97, AS97\]](#). It is likely that finding the minimal threshold beyond which soundness is guaranteed to hold will be important for determining the best possible PCP gaps.

Regardless of the PCP application, this encoding of a function  $f$  by its restrictions to cubes (or to planes) is quite natural, and is a rare example of a property that has such strong testability. The low degree agreement test theorems guarantee that even the

passing of the test with tiny  $\varepsilon$  probability has non-trivial structural consequences. Perhaps the best known comparable scenario is that of the long code, defined in [BGS98], that has similar properties, and for which an extensive line of work has been able to determine the precise threshold of soundness. Another setting with a similarly strong soundness is related to the inverse theorems for the Gowers uniformity norms. In that setting the function is given as a points-table, and the Gowers norm measures success in a low degree test, so it is not altogether dissimilar from the situation here.

To summarize, one of our goals is to pinpoint the absolute minimal soundness value for which a theorem as above holds. Can this threshold be, as it is in the aforementioned cases, as small as the value of a random assignment? In other words, could it be true that for every table whose agreement parameter is an additive  $\varepsilon > 0$  above the value that we expect from a random table, already some structure exists?

The best known value for  $\delta$  for the plane vs. plane test is due to Moshkovitz and Raz who proved in [MR08] that the plane vs. plane test has soundness for all  $\varepsilon \geq \text{poly}(d)/q^{1/8}$ . But what is the correct exponent of  $q$ ?

We make progress on this question not for the plane vs. plane test but rather for the cube vs. cube test. For our test, since the intersection consists of one point, the soundness can not go below  $1/q$  because the agreement of *every* table, even a random one, is always at least  $1/q$ .

Our main theorem is,

**Theorem 6.1.2.** *There exist constants  $\beta_1, \beta_2 > 0$  such that for every  $d$ , large enough prime power  $q$  and every  $m \geq 3$  the following holds:*

*Let  $\mathbb{F}$  be a finite field,  $|\mathbb{F}| = q$ . Let  $T$  be a cubes table, assigning to each cube  $C \subset \mathbb{F}^m$  a degree  $d$  polynomial  $T(C) : C \rightarrow \mathbb{F}$ . Let  $\alpha_{CxC}(T)$  be as defined in [Test 1](#). If  $\alpha_{CxC}(T) \geq \varepsilon$  for  $\varepsilon \geq \beta_1 d^4 / q^{1/2}$ , then there is a degree  $d$  function  $g : \mathbb{F}^m \rightarrow \mathbb{F}$  such that  $T(C) = g|_C$  on an  $\beta_2 \varepsilon$  fraction of the cubes.*

The improvement over previous theorems is that the dependence on  $q$  is  $1/q^{1/2}$  compared to  $1/q^{1/8}$ . It is an intriguing question whether the dependence on  $q$  can be made inversely linear, i.e.  $1/q$ .

**Remark 6.1.3.** *We don't know the precise dependence of  $\varepsilon$  on the degree  $d$ . In this work we made no attempt to optimize this dependence. We would like to point out that our proof can be modified to change the dependence from  $d^4$  to  $d^3$ . See [Remark 6.3.14](#) for more details.*

### Simplified analysis.

While the line vs. line test considered by Arora and Sudan [AS97] is the most natural to come up with, it is rather difficult to analyze. In contrast, one of the captivating aspects of the Raz-Safra proof is that it is combinatorial, and the low degree aspect of the table plays a role only in that it guarantees distance between distinct polynomials on a line. Our analysis continues this combinatorial approach, and further simplifies it. Unlike the Raz-Safra proof, we do not need to use induction on the dimension of the ambient space  $m$  but rather recover the global structure from  $T$  “in one shot”. We rely on ideas from direct product testing, [DG08, IKW12, DS14b], and on some spectral properties of incidence graphs such as the cube-point graph.

### Proof Outline.

Given a table  $T$ , whose agreement is some small  $\varepsilon$ , the proof must somehow come up with the global low degree function  $g : \mathbb{F}^m \rightarrow \mathbb{F}$  and then argue that on many of the cubes indeed  $T(C) = g|_C$ . Naively, we might try to define  $g$  at each point  $x$  according to the most common value among all cubes containing  $x$ . This is a viable approach when the agreement is close to 1, as is done, e.g. in the linearity testing theorem of [BLR90]. However, when the agreement is a small  $\varepsilon > 0$ , this will simply not work as we can see by considering the table half of whose entries are  $T(C) \equiv 0$  and the other half  $T(C) \equiv 1$ . The agreement of this table is an impressive  $\alpha_{\mathcal{C}\mathcal{C}}(T) = 1/2$ , and yet the suggested definition of  $g$  according to majority will yield a random function that might be quite far from any low degree function.

We get around this problem by taking a *conditional majority*. For every point  $x \in \mathbb{F}^m$  and value  $\sigma \in \mathbb{F}$  we consider only cubes containing  $x$  for which  $T(C)(x) = \sigma$ . These cubes already agree with each other on  $x$  and are thus likely to agree on any

other point of their intersection. Since the cubes containing  $x$  cover every  $y \in \mathbb{F}^m$ , we can define a function  $f_{x,\sigma} : \mathbb{F}^m \rightarrow \mathbb{F}$  on the entire space  $\mathbb{F}^m$  by taking the most popular value among these cubes (i.e. the set of cubes whose value on  $x$  is  $\sigma$ ). We choose a best  $\sigma$  for each  $x$  and are left with a global function  $f_x$  for each  $x$ .

The proof proceeds in three steps.

- **Local structure:** We show that this conditional majority definition is good, obtaining for each  $x$  and  $\sigma$  a function  $f_x : \mathbb{F}^m \rightarrow \mathbb{F}$  that is “local” in that it comes from the cubes containing a point  $x$ . This is done in [Section 6.3.1](#).
- **Global Structure:** We then show that there are many pairs  $x, y$  for which  $f_x \approx f_y$  thus finding a global  $g$  that agrees with many of the cubes. This is done in [Section 6.3.2](#).
- **Low Degree:** Finally, we show that  $g$  is very close to a true low degree function. This is done by reduction to the Rubinfeld-Sudan low degree test [\[RS96\]](#) that works in the high-soundness regime. This is done in [Section 6.3.3](#).

### **Agreement tests: low degree tests and direct product tests.**

The proof outline above resembles works on direct product testing, and this is no coincidence. The low degree testing setting can be generalized to a more abstract “agreement testing” in which a function  $f : X \rightarrow \Sigma$  is represented not as a truth table but as a collection of restrictions  $(f|_S)_{S \in \mathcal{S}}$  where  $\mathcal{S} = \{S \subset X\}$  is a collection of subsets of  $X$ . A natural agreement test can be defined and studied. This type of question was first suggested in work of Goldreich and Safra [\[GS97\]](#) in an attempt to separate the algebraic aspect of the low degree test from the combinatorial. There has been a follow-up line of work on this, [\[DR06, DG08, IKW12, DS14b\]](#), focusing especially on the case where  $X$  is a finite set,  $X = [n]$ , and  $\mathcal{S}$  is the collection of all  $k$ -element subsets of  $X$ .

In the work here we bring some of the ideas from that line of work, most notably from [\[IKW12\]](#), back to the low degree testing question. The fact that our table entries

have low degree gives us extra power which makes our proof simpler than that in the abstract setting, yielding a particularly direct proof of a low degree agreement test.

Our proof makes an explicit use of the expansion properties of the relevant incidence graphs (cube vs. line, cube vs. point etc.). This allows us to prove that for every table  $T$ , different tests have similar agreement.

**Lemma 6.1.4.** *Let  $T$  be a planes table, and let  $\alpha_{\mathcal{P}_x\mathcal{P}}(T)$  be the success probability of a test with two planes that intersects on a point. Let  $\alpha_{\mathcal{P}\ell\mathcal{P}}(T)$  be the success probability of [Test 2](#), then*

$$\alpha_{\mathcal{P}_x\mathcal{P}}(T) \left(1 - \frac{d}{q}\right) \leq \alpha_{\mathcal{P}\ell\mathcal{P}}(T) \leq \alpha_{\mathcal{P}_x\mathcal{P}}(T) + \frac{1}{q}(1 + o(1)).$$

In fact, we proved a more general equivalence between tests, the general statement appears on [Section 6.4](#).

## 6.2 Preliminaries and Notations

### 6.2.1 Notations

All the graphs we discuss throughout the chapter are bipartite bi-regular graphs. Given such graph  $G$ , whose sides are  $A, B$  we denote by  $\mathbf{1}$  the all one vector, its size will be implied by the context. For a subset of vertices  $A' \subset A$ , we denote by  $\mathbf{1}_{A'}$  the indicator vector for  $A'$ . For a vertex  $a \in A$ , we denote by  $N(a) \subseteq B$  the neighbors of  $a$  in  $G$ .

We use normalized inner product, such that for  $x, y \in \mathbb{R}^n$ ,  $\langle x, y \rangle = \frac{1}{n} \sum_i x_i y_i$ , which means that  $\langle \mathbf{1}, \mathbf{1} \rangle = 1$ . The norm is defined by  $\|x\| = \sqrt{\langle x, x \rangle}$ .

We use the notation  $x \sim S$  to denote  $x$  being sampled uniformly at random (u.a.r) from the set  $S$ , in case this set  $S$  equals the entire space, we omit this symbol and simply write  $\Pr_a$  or  $\mathbf{E}_a$  to describe choosing a uniform vertex  $a \in A$ . We use the notation  $\mathbb{I}(E)$  to denote the indicator random variable of the event  $E$ .

For two vectors  $u, v$ , we use the notation  $u \overset{\gamma}{\approx} v$  if  $u$  and  $v$  are equal on at least  $1 - \gamma$  of the coordinates.

Fix a vector space  $\mathbb{F}^m$ . An affine space of  $S$  dimension  $k$  is defined by  $k + 1$  vectors

$x_0, x_1, \dots, x_k$  such that  $x_1, \dots, x_k$  are linearly independent,

$$S = x_0 + \text{span}(x_1, \dots, x_k) = \{x_0 + t_1 x_1 + \dots + t_k x_k \mid t_1, \dots, t_k \in \mathbb{F}\}$$

A *line* is a 1-dimensional affine space, a *plane* is a 2-dimensional affine space, and a *cube* is a 3-dimensional affine space. We will denote the set of all lines and cubes by  $\mathcal{L}$  and  $\mathcal{C}$  be respectively. For a point  $x \in \mathbb{F}^m$  let

$$\mathcal{L}_x = \{\ell \in \mathcal{L} \mid \ell \ni x\} \quad \mathcal{C}_x = \{C \in \mathcal{C} \mid C \ni x\}.$$

Similarly for a line  $\ell \in \mathcal{L}$  let  $\mathcal{C}_\ell$  be the set of all cubes that contains  $\ell$ .

### 6.2.2 Spectral Expansion Properties

In this section, we prove two properties of bi-regular bipartite graphs with good spectral parameters. In an expander, the following is well known: if we sample a random neighbor of a small, but not too small, set of vertices, we get a nearly uniform distribution over the entire set of vertices. For our purposes, we will require something more. We need to consider not only the distribution over the vertices, but also the distribution over the edges. This is done in two lemmas below.

**Definition 6.2.1.** Let  $G = (A \cup B, E)$  be a bi-regular bipartite graph, and let  $M \in \mathbb{R}^{A \times B}$  be the adjacency matrix normalized such that  $\|M\mathbf{1}\| = 1$ , denote by  $\lambda(G)$  the value

$$\lambda(G) = \max_{v \perp \mathbf{1}} \left\{ \frac{\|Mv\|}{\|v\|} \right\}.$$

This is really the second largest singular value of  $M$ , with a different normalization (such that the maximal singular value equals 1).

**Definition 6.2.2.** Let  $G = (A \cup B, E)$  be a bi-regular bipartite graph and let  $B' \subseteq B$  be a subset of vertices. Define the following two distributions  $D_i : A \times B \cup \perp \rightarrow [0, 1]$  for  $i = 1, 2$ .

- $D_1$  : Pick  $b \in B'$  u.a.r. then pick  $a \in N(b)$  u.a.r.
- $D_2$  : Pick  $a \in A$  u.a.r. If  $B' \cap N(a) = \emptyset$ , return  $\perp$ . Else, pick  $b \in N(a) \cap B'$  u.a.r.

Clearly if  $B' = B$  then  $D_1 = D_2$ . Moreover, if  $G$  is sufficiently expanding, then even for smaller  $B' \subsetneq B$ , the distributions are similar. Indeed, for any event defined on the edges, i.e. a subset  $E' \subset E$ , the following lemma shows that the probability of  $E'$  is roughly the same under the two distributions.

**Lemma 6.2.3.** *Let  $D_1, D_2$  as defined in Definition 6.2.2. Let  $G = (A \cup B, E)$  be a bi-regular bipartite graph, then for every subset  $B' \subset B$  of measure  $\mu > 0$  and every  $E' \subset E$*

$$\left| \Pr_{(a,b) \sim D_1} [(a,b) \in E'] - \Pr_{(a,b) \sim D_2} [(a,b) \in E'] \right| \leq \frac{\lambda(G)}{\sqrt{\mu}}.$$

Where it is understood that if  $D_2$  output  $\perp$ , we treat it as if  $(a,b) \notin E'$ .

We now state a similar lemma, for sampling two adjacent edges instead of a single edge. We will need the graph to satisfy one more requirement.

**Definition 6.2.4.** *Let  $G = (A \cup B, E)$  be a bi-regular bipartite graph, such that every two distinct  $b_1, b_2 \in B$  have exactly the same number of common neighbors (i.e for all distinct  $b_1, b_2 \in B$ ,  $|N(b_1) \cap N(b_2)|$  is the same), and this number is non-zero. Let  $B' \subseteq B$  be a subset of vertices, we define the following distributions  $D_i : (A \times B \times B) \cup \perp \rightarrow [0, 1]$ , for  $i = 3, 4$ .*

- $D_3$  : Pick  $b_1, b_2 \in B'$  u.a.r. then pick  $a \in N(b_1) \cap N(b_2)$  u.a.r.
- $D_4$  : Pick  $a \in A$  u.a.r. If  $B' \cap N(a) = \emptyset$ , return  $\perp$ . Else, pick  $b_1, b_2 \in N(a) \cap B'$  u.a.r.

**Lemma 6.2.5.** *Let  $D_3, D_4$  be as defined in Definition 6.2.4. Let  $G = (A \cup B, E)$  be a bi-regular bipartite graph, such that every two distinct  $b_1, b_2 \in B$  have exactly the same number of common neighbors (i.e for all distinct  $b_1, b_2 \in B$ ,  $|N(b_1) \cap N(b_2)|$  is the same), and this number is non-zero. Then for every subset  $B' \subset B$  of measure  $\mu > 0$  and every  $E' \subset E$*

$$\left| \Pr_{a, b_1, b_2 \sim D_3} [(a, b_1)(a, b_2) \in E'] - \Pr_{a, b_1, b_2 \sim D_4} [(a, b_1)(a, b_2) \in E'] \right| \leq \frac{2\lambda(G)}{\mu} + \frac{1}{\mu^2 d_A} + \frac{1}{\mu^2 |B|},$$

where  $d_A$  is the degree on  $A$  side, and it is understood that if  $D_4$  output  $\perp$ , we treat it as if  $(a, b) \notin E'$ .

The proofs of these two lemmas appear in Section 6.6.

### 6.2.3 Inclusion Graphs and Their Spectral Gap

We record here the expansion of several bi-partite *inclusion graphs* that will be relevant for our analysis. We prove the claims about these spectral gaps in [Section 6.5](#). Unless otherwise stated,  $G(A, B)$  denotes a bipartite inclusion graph between  $A$  and  $B$  where  $a \in A$  is connected to  $b \in B$  if  $a \subseteq b$ . The relation of containment will be clear from the sets  $A$  and  $B$ .

For example, the in the graph  $G_1(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$ , the left side vertices  $A$  are all the lines that do not contain  $x \in \mathbb{F}^m$ , and the right side vertices are all the cubes that contain  $x$ . There is an edge between a line  $\ell$  and a cube  $C$  if  $\ell \subset C$ .

Recall [Definition 6.2.1](#) of  $\lambda(G)$  for a bipartite graph  $G$ .

**Lemma 6.2.6.** *We have for every  $m \geq 6$ ,*

- (1) *For  $G_1(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$  ,  $\lambda(G_1) \approx \frac{1}{\sqrt{q}}$ .*
- (2) *For  $G_2(\mathcal{L}_x, \mathcal{C}_x)$  ,  $\lambda(G_2) \approx \frac{1}{q}$ .*
- (3) *For  $G_3(\mathbb{F}^m \setminus \ell, \mathcal{C}_\ell)$  ,  $\lambda(G_3) \approx \frac{1}{\sqrt{q}}$ .*
- (4) *For  $G_4(\mathbb{F}^m, \mathcal{C})$  ,  $\lambda(G_4) \approx \frac{1}{q^{3/2}}$ .*
- (5) *For  $G_5(\mathbb{F}^m \setminus \{x\}, \mathcal{C}_x)$  ,  $\lambda(G_5) \approx \frac{1}{q}$ .*

*And for every  $m \geq 3$*

- (6) *For  $G_6(\mathbb{F}^m, \mathcal{L})$ ,  $\lambda(G_6) \approx \frac{1}{\sqrt{q}}$ .*

*where  $\approx$  denotes equality up to a multiplicative factor of  $1 \pm o(1)$ , and  $o(1)$  denotes a function that approaches zero as  $q \rightarrow \infty$ .*

In general one can see that  $\lambda \approx \frac{1}{\sqrt{q^p}}$  where  $p$  is the number of degrees of freedom left after choosing a left hand vertex. We prove this lemma in [Section 6.5](#).

## 6.3 Proof of the Main Theorem

In this section we prove [Theorem 6.1.2](#) in three steps - local structure, global structure and finally proving the agreement with a low degree polynomial. These parts are proved

in the subsequent subsections.

Let  $T$  be a degree  $d$  cubes table, i.e. for every  $C \in \mathcal{C}$ ,  $T(C) : C \rightarrow \mathbb{F}$  is a degree  $d$  polynomial. Further assume that  $\alpha_{\mathcal{C}x\mathcal{C}}(T) \geq \varepsilon$ , where  $\varepsilon = \Omega(d^4/\sqrt{q})$ .

### 6.3.1 Local Structure

In this section we show that for many points  $x \in \mathbb{F}^m$ , there exists a function  $f_x : \mathbb{F}^m \rightarrow \mathbb{F}$  for which  $f_x|_C \stackrel{2\gamma}{\approx} T(C)$  for a good fraction of the cubes containing  $x$ , for  $\gamma = \Omega(1/d^3)$ . Recall that  $\stackrel{2\gamma}{\approx}$  means that the two functions agree on  $1 - 2\gamma$  fraction of the points in their domain.

For each  $x \in \mathbb{F}^m$  and  $\sigma \in \mathbb{F}$ , we define

$$\mathcal{C}_{x,\sigma} = \{C \in \mathcal{C}_x | T(C)(x) = \sigma\}.$$

Following [IKW12] we have the following important definition,

**Definition 6.3.1** (Excellent pair).  $(x, \sigma)$  is  $(\frac{\varepsilon}{2}, \gamma)$ -excellent if:

1.  $\Pr_{C \in \mathcal{C}_x}[C \in \mathcal{C}_{x,\sigma}] \geq \frac{\varepsilon}{2}$ .
2. Let  $C_1, \ell, C_2$  be chosen by the following probability distribution,  $C_1 \in \mathcal{C}_{x,\sigma}$  u.a.r,  $\ell \subset C_1$  a random line that contains  $x$  and  $C_2 \in \mathcal{C}_{x,\sigma} \cap \mathcal{C}_\ell$  (a random cube in  $\mathcal{C}_{x,\sigma}$  that contains  $\ell$ ).

$$\Pr_{C_1, \ell, C_2} [T(C_1)|_\ell \neq T(C_2)|_\ell] \leq \gamma.$$

A point  $x \in \mathbb{F}^m$  is  $(\frac{\varepsilon}{2}, \gamma)$ -excellent, if exists  $\sigma \in \mathbb{F}$  such that  $(x, \sigma)$  is  $(\frac{\varepsilon}{2}, \gamma)$ -excellent.

Note that in the definition of excellent, the marginal distribution of both  $C_1, C_2$  is uniform in  $\mathcal{C}_{x,\sigma}$ .

In the sequel, we fix  $\gamma = \Omega(1/d^3)$  and say that a point is excellent if it is  $(\frac{\varepsilon}{2}, \gamma)$ -excellent. We now state the main lemma in this section.

**Lemma 6.3.2** (Local Structure). For  $\gamma = \Omega(\frac{1}{d^3})$ , let  $T$  be a cubes table that passes [Test 1](#) with probability larger than  $\varepsilon = \Omega(\frac{d^4}{\sqrt{q}})$ , then at least  $\frac{\varepsilon}{3}$  of the points  $x \in \mathbb{F}^m$  are excellent, and for each excellent  $x$  there exist a function  $f_x : \mathbb{F}^m \rightarrow \mathbb{F}$  such that

$$\Pr_{C \sim \mathcal{C}_x} [T(C) \stackrel{2\gamma}{\approx} f_x|_C] \geq \frac{\varepsilon}{4}.$$

We will consider the distribution  $\mathcal{D}$  on  $(x, \ell, C_1, C_2)$  obtained by choosing  $x$  uniformly, choosing  $\ell \in \mathcal{L}_x$  uniformly, and then choosing  $C_1, C_2 \in \mathcal{C}_\ell$  uniformly.

This distribution induces a distribution  $(x, T(C_1)(x))$  on pairs of point  $x$  and value  $\sigma \in \mathbb{F}$ .

**Claim 6.3.3.** *For every  $\gamma = \Omega(\frac{1}{d^3})$ ,*

$$\Pr_{(x, \sigma)} [(x, \sigma) \text{ is } (\frac{\varepsilon}{2}, \gamma) \text{ - excellent}] \geq \frac{\varepsilon}{3}.$$

*Proof.* We consider  $(x, \ell, C_1, C_2)$  chosen according to  $\mathcal{D}$ , and we note that the marginal distribution over all elements is uniform. We also write  $\sigma = T(C_1)(x)$ . We define the following events on  $(x, \ell, C_1, C_2)$ :

1.  $E$  : “ $\ell$  is confusing for  $x$ ”:  $T(C_1)(x) = T(C_2)(x)$ ,  $T(C_1)|_\ell \neq T(C_2)|_\ell$ .
2.  $H$  : “ $x, C_1$  is heavy”:  $\Pr_{C \sim \mathcal{C}_x}[T(C)(x) = T(C_1)(x)] \geq \frac{\varepsilon}{2}$

Since  $T(C_1)|_\ell, T(C_2)|_\ell$  are two degree  $d$  polynomials, and  $x$  is a random point in  $\ell$ ,

$$\Pr_{(x, \ell, C_1, C_2)} [E] \leq \frac{d}{q}.$$

Using the fact that  $\alpha_{\mathcal{C}_x \mathcal{C}}(T) \geq \varepsilon$ , and averaging, we get

$$\Pr_{(x, \ell, C_1, C_2)} [H] \geq \frac{\varepsilon}{2}. \tag{6.3.1}$$

Instead of picking  $C_1$  as a uniform cube containing  $x$ , we can choose it by the following process, pick  $\sigma$  proportional to its weight in  $\mathcal{C}_x$ , then pick  $C_1 \sim \mathcal{C}_{x, \sigma}$ . This process describes the same distribution.

Note that after deciding  $x, \sigma$ , the event  $H$  is already determined, so (6.3.1) becomes  $\Pr_{x, \sigma}[H] \geq \varepsilon/2$ . Also, notice that conditioned on  $x, \sigma$ , the distribution  $\mathcal{D}$  is choosing  $C_1$  uniformly from  $\mathcal{C}_{x, \sigma}$  and then  $\ell \subset C_1$  a random line containing  $x$  and then  $C_2$  a random cube containing  $\ell$  (and we do not require that  $T(C_2)(x) = \sigma$ ). The event  $H$  is already fixed by  $x, \sigma$ , but the event  $E$  will occur only if  $C_2 \in \mathcal{C}_{x, \sigma}$  and also  $T(C_1)|_\ell \neq T(C_2)|_\ell$ .

We want to bound the probability of  $x, \sigma$  such that  $H = 1$ , but  $\mathbf{E}_{C_1, \ell, C_2}[E|x, \sigma] \leq \gamma \cdot \frac{\varepsilon}{2}$ . We know that

$$\mathbf{E}_{x, \sigma} [\Pr[H \wedge E \mid x, \sigma]] = \Pr[H \wedge E] \leq \Pr[E] \leq \frac{d}{q}.$$

Therefore, by averaging, the probability over  $x, \sigma$  that we have  $\Pr[H \wedge E|x, \sigma] > \varepsilon\gamma/2$  is at most  $\frac{d/q}{\varepsilon\gamma/2}$ . So for at least  $\varepsilon/2 - \frac{d/q}{\varepsilon\gamma/2} \geq \varepsilon/3$  of the pairs  $x, \sigma$ , we have that both  $H$  occurs, and that  $\mathbf{E}_{C_1, \ell, C_2}[E|x, \sigma] \leq \varepsilon\gamma/2$ .

We end by showing that such  $x, \sigma$  are excellent. The first requirement follows by the fact that  $H$  occurs, for the second we need to show that for  $C_1 \in \mathcal{C}_{x, \sigma}$ , a uniform  $\ell \in \mathcal{C}_1$  and a uniform  $C_2 \in \mathcal{C}_{x, \sigma} \cap C_\ell$  the probability of  $T(C_1)|_\ell \neq T(C_2)|_\ell$  is lower than  $\gamma$ .

We notice that after fixing  $(x, \sigma)$ , the distribution  $\mathcal{D}$  chooses  $C_1 \in \mathcal{C}_{x, \sigma}$ , a uniform  $\ell \in \mathcal{C}_1$ , but then a uniform  $C_2 \in \mathcal{C}_\ell$ .

The event  $E$  can be written as  $E = E_1 \wedge E_2$  where  $E_1$  is the event “ $T(C_1)(x) = T(C_2)(x)$ ” and  $E_2$  is the event “ $T(C_1)|_\ell \neq T(C_2)|_\ell$ ”. In this notation

$$\begin{aligned} \mathbf{E}_{C_1, \ell, C_2}[E|x, \sigma] &= \mathbf{E}_{C_1, \ell, C_2}[E_1 \wedge E_2|x, \sigma] \\ &= \mathbf{E}_{C_1, \ell, C_2}[E_1|x, \sigma] \mathbf{E}_{C_1, \ell, C_2}[E_2|E_1, x, \sigma] \\ &\geq \frac{\varepsilon}{2} \cdot \mathbf{E}_{C_1, \ell, C_2}[E_2|E_1, x, \sigma]. \end{aligned} \quad (\text{since } H \text{ occurs})$$

We notice that if  $E_1$  occurs, then  $C_2 \in \mathcal{C}_{x, \sigma}$ , therefore

$$\mathbf{E}_{C_1, \ell, C_2}[T(C_1)|_\ell \neq T(C_2)|_\ell | C_2 \in \mathcal{C}_{x, \sigma}, x, \sigma] \leq \frac{2}{\varepsilon} \cdot \mathbf{E}_{C_1, \ell, C_2}[E|x, \sigma] \leq \frac{2}{\varepsilon} \frac{\varepsilon}{2} \gamma \leq \gamma,$$

which means that  $(x, \sigma)$  is  $(\frac{\varepsilon}{2}, \gamma)$  - excellent. ■

For each  $(x, \sigma)$  we define  $f_{x, \sigma}$  by plurality over all cubes  $C \in \mathcal{C}_{x, \sigma}$ .

**Definition 6.3.4.** For a pair  $(x, \sigma)$  define a function  $f_{x, \sigma} : \mathbb{F}^m \rightarrow \mathbb{F}$  as follows:

$$f_{x, \sigma}(y) = \operatorname{argmax}_{C \sim \mathcal{C}_y \cap \mathcal{C}_{x, \sigma}} \{T(C)(y)\}.$$

If  $\mathcal{C}_y \cap \mathcal{C}_{x, \sigma} = \emptyset$ , define  $f_{x, \sigma}(y)$  arbitrarily.

**Claim 6.3.5.** For an  $(\frac{\varepsilon}{2}, \gamma)$  excellent pair  $(x, \sigma)$ ,

$$\Pr_{C \sim \mathcal{C}_{x, \sigma}, y \sim C}[f_{x, \sigma}(y) = T(C)(y)] \geq 1 - \gamma.$$

*Proof.* Fix an  $(\frac{\varepsilon}{2}, \gamma)$  excellent pair  $(x, \sigma)$ , and denote  $f = f_{x, \sigma}$ . If we pick a uniform  $C_1 \in \mathcal{C}_{x, \sigma}$ , then  $y \in C_1$  such that  $y \neq x$ , and a uniform  $C_2 \in \mathcal{C}_{x, \sigma} \cap \mathcal{C}_y$ , then

$$\Pr_{C_1, y, C_2} [T(C_1)(y) \neq T(C_2)(y)] \leq \Pr_{C_1, y, C_2} [T(C_1)|_{\ell(x, y)} \neq T(C_2)|_{\ell(x, y)}] \leq \gamma,$$

since  $(x, \sigma)$  is  $(\frac{\varepsilon}{2}, \gamma)$  excellent.

For each  $y$ , denote  $\gamma_y = \Pr_{C_1, C_2 \sim \mathcal{C}_{x, \sigma} \cap \mathcal{C}_y} [T(C_1)(y) \neq T(C_2)(y)]$ . From the above we get that  $\mathbb{E}_y[\gamma_y] \leq \gamma$ , where  $y$  is distributed according to its weight in  $\mathcal{C}_{x, \sigma}$ . For each  $y$ ,

$$\begin{aligned} 1 - \gamma_y &= \sum_{\theta \in \mathbb{F}} \Pr_{C \sim \mathcal{C}_{x, \sigma} \cap \mathcal{C}_y} [T(C)(y) = \theta]^2 \\ &\leq \Pr_{C \sim \mathcal{C}_{x, \sigma} \cap \mathcal{C}_y} [T(C)(y) = f(y)] \sum_{\theta \in \mathbb{F}} \Pr_{C \sim \mathcal{C}_{x, \sigma} \cap \mathcal{C}_y} [T(C)(y) = \theta] \\ &\quad (f(y) \text{ is the most frequent value}) \\ &\leq \Pr_{C \sim \mathcal{C}_{x, \sigma} \cap \mathcal{C}_y} [T(C)(y) = f(y)]. \end{aligned}$$

Since it is true for each  $y$ , it is also true when taking expectation over  $y$ , for any distribution:

$$\Pr_{C \sim \mathcal{C}_{x, \sigma}, y \sim C} [f(y) = T(C)(y)] = \mathbf{E}_y \left[ \mathbf{E}_{C \sim \mathcal{C}_{x, \sigma} \cap \mathcal{C}_y} [\mathbb{I}(T(C)(y) = f(y))] \right] \geq \mathbf{E}_y [1 - \gamma_y] \geq 1 - \gamma.$$

In expectation, each  $y$  is chosen with probability proportional to its weight in  $\mathcal{C}_{x, \sigma}$ , as before. ■

### Proof of Lemma 6.3.2:

From Claim 6.3.3 we know that the probability of  $(x, \sigma)$  to be  $(\frac{\varepsilon}{2}, \gamma)$ -excellent is at least  $\frac{\varepsilon}{3}$ . Since  $x$  is chosen uniformly, it means that for at least  $\frac{\varepsilon}{3}$  of the inputs  $x \in \mathbb{F}^m$  there exists some  $\sigma \in \mathbb{F}$  such that  $(x, \sigma)$  is excellent. If there is more than one such  $\sigma$  choose one arbitrarily.

Fixing an excellent  $x$ , let  $\sigma$  be the value such that  $(x, \sigma)$  is excellent. For this  $\sigma$ ,  $\Pr_{C \in \mathcal{C}_x} [C \in \mathcal{C}_{x, \sigma}] \geq \frac{\varepsilon}{2}$ . From Claim 6.3.5,  $\Pr_{C \sim \mathcal{C}_{x, \sigma}, y \sim C} [f_{x, \sigma}(y) = T(C)(y)] \geq 1 - \gamma$ . By averaging, at least half of the cubes  $C \in \mathcal{C}_{x, \sigma}$  satisfy  $\Pr_{y \sim C} [f_{x, \sigma}(y) = T(C)(y)] \geq 1 - 2\gamma$ . For all these cubes  $T(C) \stackrel{2\gamma}{\approx} f_{x, \sigma}$ , and they are at least  $\frac{\varepsilon}{4}$  fraction of the cubes in  $\mathcal{C}_x$ . ■

### 6.3.2 Global Structure

In this section, we prove the following lemma:

**Lemma 6.3.6** (Global Structure). *Let  $T$  be a cubes table that passes [Test 1](#) with probability at least  $\varepsilon = \Omega(\frac{d^4}{\sqrt{q}})$ , then for every  $\gamma = \Omega(\frac{1}{d^3})$ , there exists an  $(\frac{\varepsilon}{2}, \gamma)$ -excellent  $x$  such that  $f = f_x : \mathbb{F}^m \rightarrow \mathbb{F}$  satisfies*

$$\Pr_C[T(C) \stackrel{32\gamma}{\approx} f|_C] \geq \frac{\varepsilon}{16}.$$

Let  $X^* \subseteq \mathbb{F}^m$  the set of  $(\frac{\varepsilon}{2}, \gamma)$  excellent points.

The main idea in the proof of the global structure, is showing that there exist many pairs of excellent points  $x, y \in X^*$ , such that for many cubes  $C$ , the  $T(C)$  is similar both to  $f_x$  and to  $f_y$  ([Claim 6.3.8](#)). If this is the case, then the functions  $f_x, f_y$  must be very similar ([Claim 6.3.9](#)). Finally, the lemma is proven by averaging and finding a single  $x$  such that  $f_x$  agrees simultaneously with many of the  $f_y$ 's and their supporting cubes.

**Definition 6.3.7** (Supporting cubes). *For any excellent  $x \in X^*$ , we denote by  $F_x$  the set of cubes “supporting”  $f_x$ ,*

$$F_x = \left\{ C \in \mathcal{C}_x \mid T(C) \stackrel{2\gamma}{\approx} f_{x|_C} \right\}.$$

**Claim 6.3.8.** *Let  $\mathcal{D}$  be the following process: choose  $x, y \in X^*$  independently and uniformly at random, let  $C$  be a random cube containing both  $x$  and  $y$ . Then*

$$\Pr_{x, y, C \sim \mathcal{D}}[C \in F_x \cap F_y] \geq \frac{\varepsilon^2}{26}.$$

*Proof.* Since each  $x \in X^*$  is excellent, we know from the local structure lemma, [Lemma 6.3.2](#), that  $\Pr_{C \sim \mathcal{C}_x}[C \in F_x] \geq \frac{\varepsilon}{4}$ . This is of course also true when taking a uniform  $x \in X^*$ , thus,  $\Pr_{x \sim X^*, C \sim \mathcal{C}_x}[C \in F_x] \geq \frac{\varepsilon}{4}$ .

From [Lemma 6.2.6\(4\)](#), the inclusion graph  $G = G(\mathbb{F}^m, \mathcal{C})$  has  $\lambda(G) = \lambda \leq (1 + o(1))\frac{1}{q^{3/2}}$ . Denote the measure of  $X^*$  by  $\mu$ , from [Lemma 6.3.2](#),  $\mu \geq \frac{\varepsilon}{3}$ . Hence, by the application of [Lemma 6.2.3](#) on the graph  $G$  with  $A = \mathcal{C}$ ,  $B = \mathbb{F}^m$  and  $B' = X^*$ , we get

$$\left| \Pr_{x \sim X^*, C \sim \mathcal{C}_x}[C \in F_x] - \Pr_{C \sim \mathcal{C}, x \sim C \cap X^*}[C \in F_x] \right| \leq \frac{\lambda}{\sqrt{\mu}} \leq \frac{2\lambda}{\sqrt{\varepsilon}}. \quad (6.3.2)$$

For each  $C \in \mathcal{C}$ , let  $p_C = \Pr_{x \sim C \cap X^\star}[C \in F_x]$ , this measures for every cube  $C$  how many points  $x \in C$  are such that  $f_{x|C} \stackrel{2\gamma}{\approx} T(C)$ . In this notation, (6.3.2) implies  $\mathbf{E}_C[p_C] \geq \frac{\varepsilon}{4} - \frac{2\lambda}{\sqrt{\varepsilon}} \geq \frac{\varepsilon}{5}$ . We can use this to bound the probability of the event  $C \in F_x \cap F_y$  by first choosing  $C$ , then two independent points in  $C \cap X^\star$ ,

$$\Pr_{\substack{C \sim \mathcal{C} \\ x, y \sim C \cap X^\star}}[C \in F_x \cap F_y] = \mathbf{E}_C[p_C^2] \geq \left(\mathbf{E}_C[p_C]\right)^2 \geq \frac{\varepsilon^2}{25}.$$

We observe that this distribution is very similar to the required distribution  $D$ . The only difference is that here we first pick  $C \in \mathcal{C}$  and then two excellent points in  $C$ , whereas in  $D$  we first pick two points in  $X^\star$  and then a common neighbor  $C$ . The graph  $G$  satisfies that every two distinct points  $x, y \in \mathbb{F}^m$  have exactly the same number of common neighbors. Therefore, we can use Lemma 6.2.5 on the graph  $G$  with  $A = \mathcal{C}, B = \mathbb{F}^m$  and  $B' = X^\star$  to get

$$\left| \Pr_{\substack{C \sim \mathcal{C} \\ x, y \sim C \cap X^\star}}[C \in F_x \cap F_y] - \Pr_{x, y, C \sim D}[C \in F_x \cap F_y] \right| \leq \frac{2\lambda}{\mu} + \frac{1}{\mu^2 d_A} + \frac{1}{\mu^2 |B|} \leq \frac{6\lambda}{\varepsilon} + \frac{9}{q^m \varepsilon^2} + \frac{9}{q^3 \varepsilon^2}.$$

Recall that  $\lambda \leq (1 + o(1)) \frac{1}{q^{3/2}}$  and since  $\varepsilon = \Omega(\frac{d^4}{\sqrt{q}})$ , we conclude that  $\Pr_{x, y, C \sim D}[C \in F_x \cap F_y] \geq \frac{\varepsilon^2}{25} - \frac{6\lambda}{\varepsilon} - \frac{9}{q^m \varepsilon^2} - \frac{9}{q^3 \varepsilon^2} \geq \frac{\varepsilon^2}{26}$ .  $\blacksquare$

**Claim 6.3.9.** *Let  $x \neq y \in X^\star$ , and let  $\ell$  be the line containing  $x$  and  $y$ , if  $\Pr_{C \sim \mathcal{C}_\ell}[C \in F_x \cap F_y] \geq \frac{\varepsilon^2}{100}$  then  $f_x \stackrel{5\gamma}{\approx} f_y$ .*

*Proof.* Consider the graph  $G = G(\mathbb{F}^m \setminus \ell, \mathcal{C}_\ell)$ . This is a bi-regular bipartite graph, and by Lemma 6.2.6(3) it has  $\lambda = \lambda(G) \leq (1 + o(1)) \frac{1}{\sqrt{q}}$ . Let  $F = F_x \cap F_y$ . By assumption,  $F$  has measure at least  $\frac{\varepsilon^2}{100}$  inside  $\mathcal{C}_\ell$ .

We denote by  $E' \subset E$  the edges of  $G$  that indicate agreement with both  $f_x$  and  $f_y$ ,

$$E' = \{(z, C) \mid T(C)(z) = f_x(z) = f_y(z)\}.$$

Every cube  $C \in F$  has  $1 - 2\gamma$  of the points  $z \in C$  satisfying  $T(C)(z) = f_x(z)$  and  $1 - 2\gamma$  of the points satisfying  $T(C)(z) = f_y(z)$ . By a union bound we get  $\Pr_{C \in F, z \in N(C)}[(z, C) \in E'] \geq 1 - 4\gamma$ . By Lemma 6.2.3 on  $G$  when  $A = \mathbb{F}^m \setminus \ell, B = \mathcal{C}_\ell, B' = F$ ,

$$\left| \Pr_{C \sim F, z \sim N(C)}[(z, C) \in E'] - \Pr_{z, C \sim N(z) \cap F}[(z, C) \in E'] \right| \leq \frac{20\lambda}{\varepsilon},$$

which means that  $\Pr_{z \sim \mathbb{F}^m, C \sim N(z) \cap F}[(z, C) \in E'] \geq 1 - 4\gamma - \frac{20\lambda}{\varepsilon} \geq 1 - 5\gamma$ . By the definition of  $E'$ , for each point  $z \in \mathbb{F}^m$  that has an adjacent edge in  $E'$ ,  $f_x(z) = f_y(z)$ . This means that

$$\Pr_z[f_x(z) = f_y(z)] \geq \Pr_z[\exists C \text{ s.t. } (z, C) \in E'] \geq \Pr_{z, C \sim N(z) \cap F}[(z, C) \in E'] \geq 1 - 5\gamma.$$

■

The above claim showed that if two functions have a large set of cubes on which they almost agree then these functions are similar. In order to prove the global structure, we also need to show that in this case, most of  $C \in F_y$  will also be close to  $f_x$ .

**Claim 6.3.10.** *Let  $x, y \in X^\star$  such that  $f_x \overset{5\gamma}{\approx} f_y$ , then*

$$\Pr_{C \sim F_y}[T(C) \overset{32\gamma}{\approx} f_{x|C}] \geq \frac{1}{2}.$$

Note that the function  $f_x$  may not be a low degree polynomial, so  $T(C) \overset{32\gamma}{\approx} f_{x|C}$  doesn't imply equality.

*Proof.* Let  $G = G(\mathbb{F}^m \setminus \{y\}, \mathcal{C}_y)$ , by [Claim 6.2.6\(5\)](#) it has  $\lambda = \lambda(G) \approx \frac{1}{q}$ . First, we denote by  $E'_y$  the following set of edges,

$$E'_y = \{(z, C) \mid T(C)(z) = f_y(z)\}.$$

For each  $C \in F_y$ , we know that  $\Pr_{z \in N(C)}[(z, C) \in E'_y] \geq 1 - 2\gamma$ . From [Lemma 6.2.3](#) on  $G$  when  $A = \mathbb{F}^m \setminus y, B = \mathcal{C}_y, B' = F_y$ , we know that

$$\left| \Pr_{C \sim F_y, z \sim N(C)}[(z, C) \in E'_y] - \Pr_{z, C \in N(z) \cap F_y}[(z, C) \in E'_y] \right| \leq \frac{4\lambda}{\varepsilon},$$

since the measure of  $F_y$  is at least  $\frac{\varepsilon}{4}$ . This implies that  $\Pr_{z, C \in N(z) \cap F_y}[(z, C) \in E'_y] \geq 1 - 3\gamma$ .

We define a second set of edges,  $E'_x$  to be the same only for  $f_x$ ,

$$E'_x = \{(z, C) \mid T(C)(z) = f_x(z)\}.$$

We notice that if  $z$  is a point such that  $f_x(z) = f_y(z)$ , then  $(z, C) \in E'_y \Rightarrow (z, C) \in E'_x$ .

$$\begin{aligned}
\Pr_{z, C \sim N(z) \cap F_y} [(z, C) \in E'_x] &\geq \Pr_z[f_x(z) = f_y(z)] \cdot \Pr_{z, C \sim N(z) \cap F_y} [(z, C) \in E'_y \mid f_x(z) = f_y(z)] \\
&\geq (1 - 5\gamma) \cdot \Pr_{z, C \sim N(z) \cap F_y} [(z, C) \in E'_y \mid f_x(z) = f_y(z)] \\
&\hspace{15em} (\text{since } f_x \overset{5\gamma}{\approx} f_y) \\
&\geq (1 - 5\gamma) \cdot \left( \Pr_{z, C \sim N(z) \cap F_y} [(z, C) \in E'_y] - 5\gamma \right) \\
&\geq 1 - 15\gamma.
\end{aligned}$$

Therefore, we can use [Lemma 6.2.3](#) again on the same graph  $G$  and set  $F_y$ , now with the edge set  $E'_x$ , to conclude that

$$\Pr_{C \sim F_y, z \sim N(C)} [(z, C) \in E'_x] \geq \Pr_{z, C \sim N(z) \cap F_y} [(z, C) \in E'_x] - \frac{4\lambda}{\varepsilon} \geq 1 - 16\gamma,$$

By averaging, at least half of  $C \in F_y$  satisfies  $T(C) \overset{32\gamma}{\approx} f_{x|C}$ . ■

We are now ready to prove the global structure.

### Proof of [Lemma 6.3.6](#):

Let  $T$  be the cubes table that passes [Test 1](#) with probability at least  $\varepsilon = \Omega(\frac{d^4}{\sqrt{q}})$ . From the local structure, [Lemma 6.3.2](#), we know that there exists a set  $X^*$  of excellent points, such that each  $x \in X^*$  has a function  $f_x$ , and  $|F_x| \geq \frac{\varepsilon}{4} |\mathcal{C}_x|$ .

From [Claim 6.3.8](#), we know that  $\Pr_{x, y, C \sim D}[C \in F_x \cap F_y] \geq \frac{\varepsilon^2}{26}$ , when  $x, y$  are chosen uniformly from  $X^*$  and  $C$  is a common neighbor. Therefore, there must be  $x \in X^*$  such that  $\Pr_{y \sim X^*, C \sim N(x) \cap N(y)}[C \in F_x \cap F_y] \geq \frac{\varepsilon^2}{26}$ .

Fix such  $x \in X^*$ , and let  $X'$  be the set of  $y \in X^*$  such that  $|F_x \cap F_y| \geq \frac{\varepsilon^2}{100} |\mathcal{C}_y|$ . By averaging,  $|X'| \geq \frac{\varepsilon^2}{100} |X^*| \geq \frac{\varepsilon^3}{400} |\mathbb{F}|^m$ .

By [Claim 6.3.9](#), for all  $y \in X'$ ,  $f_y \overset{5\gamma}{\approx} f_x$ . For each  $y \in X'$ , let

$$F'_y = \{C \in F_y \mid T(C) \overset{32\gamma}{\approx} f_{x|C}\}.$$

At this point we have a large collection of  $y$ 's and for each one a large collection of cubes  $F'_y$  such that all of these support the same function  $f_x$ . It is immediate that  $f_x$  is

supported by some  $\text{poly}(\varepsilon)$  fraction of all of the cubes. Since we are aiming for a better quantitative bound of  $\Omega(\varepsilon)$  fraction of  $\mathcal{C}$ , we will rely on the expansion once more.

In order to finish the proof, we need to show that  $|\cup_{y \in X'} F'_y| \geq \frac{\varepsilon}{16} |\mathcal{C}|$ .

Let  $G = G(\mathbb{F}^m, \mathcal{C})$ , by [Lemma 6.2.6\(4\)](#)  $\lambda(G) \leq q^{-\frac{3}{2}}$ . We use  $X'$  as the set of vertices, and define

$$E' = \{(y, C) \mid T(C) \stackrel{32\gamma}{\approx} f_{x|_C}\}.$$

By [Lemma 6.2.3](#) on  $G$  with  $A = \mathcal{C}, B = \mathbb{F}^m, B' = X'$ ,

$$\left| \Pr_{y \sim X', C \sim N(y)}[(y, C) \in E'] - \Pr_{C \sim \mathcal{C}, y \sim N(C) \cap X'}[(y, C) \in E'] \right| \leq \frac{20\lambda}{\sqrt{\varepsilon^3}} \leq \frac{20q^{-\frac{3}{2}}}{q^{-\frac{3}{4}}} \leq 20q^{-\frac{3}{4}} \leq \frac{\varepsilon}{16},$$

where we used the fact that  $\varepsilon \geq \frac{1}{\sqrt{q}}$ .

[Claim 6.3.10](#) lets us bound the first term on the left, since for each  $y \in X'$ ,  $\Pr_{C \sim N(y)}[C \in F'_y] \geq \frac{1}{2} \Pr_{C \sim N(y)}[C \in F_y] \geq \frac{\varepsilon}{8}$ . Thus,

$$\Pr_{C \sim \mathcal{C}, y \sim N(C) \cap X'}[(y, C) \in E'] \geq \frac{\varepsilon}{8} - \frac{\varepsilon}{16} = \frac{\varepsilon}{16}.$$

We notice that a cube with even a single adjacent edge in  $E'$  satisfies  $T(C) \stackrel{32\gamma}{\approx} f_{x|_C}$ , so we are done. ■

### 6.3.3 Low Degree

The last step is to prove that the global function discovered in the previous section can be modified to make it a low degree function, while still maintaining large support for it among the cubes.

**Theorem 6.3.11** (Theorem [6.1.2](#) restated). *For every  $d$  and large enough prime power  $q$  and every  $m \geq 3$  the following holds. Let  $T$  be a cubes table that passes [Test 1](#) with probability at least  $\varepsilon = \Omega(\frac{d^4}{\sqrt{q}})$ , then there exist a degree  $d$  polynomial  $g : \mathbb{F}^m \rightarrow \mathbb{F}$  such that  $T(C) = g|_C$  on an  $\Omega(\varepsilon)$  fraction of the cubes.*

From [Lemma 6.3.6](#), we get a function  $f$  such that  $\Omega(\varepsilon)$  of the cubes have  $T(C) \approx f|_C$ . In this section, we will show that this function  $f$  is close to a degree  $d$  polynomial  $g$ . Afterwards, we also need to show that  $\Omega(\varepsilon)$  of the cubes satisfies  $T(C) = g|_C$ .

To show the first part, we will use a robust characterization of low degree polynomials given by Rubinfeld and Sudan.

**Theorem 6.3.12** ([RS96, Theorem 4.1]). *Let  $f : \mathbb{F}^m \rightarrow \mathbb{F}$  be a function, and let  $N_{y,h} = \{y + i(h - y) \mid i \in \{0, \dots, d+1\}\}$ , if  $f$  satisfies*

$$\Pr_{y,h \in \mathbb{F}^m} [\exists \deg d \text{ polynomial } p \text{ s.t. } p|_{N_{y,h}} = f|_{N_{y,h}}] \geq 1 - \delta,$$

*for  $\delta \leq \frac{1}{2(d+2)^2}$ , then there exists a degree  $d$  polynomial  $g$  such that  $f \approx^{2\delta} g$ .*

For completeness, we present proof of the above theorem in [Section 6.7](#).

**Claim 6.3.13.** *Fix any  $\gamma \leq \frac{1}{100(d+2)^3}$ , let  $f : \mathbb{F}^m \rightarrow \mathbb{F}$  and  $x \in \mathbb{F}^m$  such that*

$$\Pr_{C \in \mathcal{C}_x} [T(C) \overset{32\gamma}{\approx} f|_C] \geq \frac{\varepsilon}{4},$$

*then exists a degree  $d$  polynomial  $g$  such that  $f \overset{84d\gamma}{\approx} g$ .*

*Proof.* Denote by  $F \subseteq \mathcal{C}_x$  the following set

$$F = \{C \in \mathcal{C}_x \mid T(C) \overset{32\gamma}{\approx} f|_C\}.$$

Our first goal is to show that for nearly all lines,  $f$  agrees with a low degree function on almost all of the points of the line.

Fix  $C \in F$ , if we pick a uniform  $\ell \subset C$  we expect that  $T(C)_\ell \overset{O(\gamma)}{\approx} f|_\ell$ . Using the spectral properties we show that almost all lines satisfy this property. Let  $G_C = G(A \cup B, E)$  be the following bipartite inclusion graph where  $A$  is all the points in  $C$ , and  $B$  is all the affine lines in  $C$ . Let  $A' \subset A$  be  $A' = \{y \in A \mid T(C)(y) \neq f(y)\}$ , and  $B' \subset B$  be  $B' = \{\ell \in B \mid |N(\ell) \cap A'| \geq 40\gamma |N(\ell)|\}$ . From [Lemma 6.2.6\(6\)](#) with  $m = 3$  (we apply the lemma where " $\mathbb{F}^m$ " is the cube  $C$ ),  $\lambda_C = \lambda(G_C) \leq \frac{2}{\sqrt{q}}$ . We apply [Lemma 6.2.3](#) on  $G_C$  and the set  $B'$ , where the set of edges is all the edges adjacent to  $A'$ :

$$\left| \Pr_{y \in A, \ell \in N(y) \cap B'} [y \in A'] - \Pr_{\ell \in B', y \in N(\ell)} [y \in A'] \right| \leq \frac{\lambda_C}{\sqrt{\frac{|B'|}{|B|}}}.$$

We notice that  $\Pr_{y \in A} [y \in A'] \leq 32\gamma$ . By the definition of  $B'$ ,  $\Pr_{\ell \in B', y \in N(\ell)} [y \in A'] \geq 40\gamma$ . Therefore  $|B'| \leq \left(\frac{\lambda_C}{8\gamma}\right)^2 |B| < \gamma |B|$ .

We have shown that for every cube  $C \in F$ , almost all lines in it satisfy  $T(C)_\ell \stackrel{40\gamma}{\approx} f|_\ell$ . Now we need to show that the set  $F$  is large enough to cover  $(1 - O(\gamma))$  of all the lines in  $\mathcal{L}$ . The inclusion graph  $G = G(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$  has  $\lambda = \lambda(G) \leq \frac{1}{\sqrt{q}}$ , by Lemma 6.2.6(1). We denote by  $E'$  the set of edges  $(\ell, C)$  such that  $T(C)|_\ell \stackrel{40\gamma}{\approx} f|_\ell$ . As we've seen above, for every  $C \in F$ ,  $\Pr_{\ell \in N(C)}[(\ell, C) \in E'] \geq 1 - \gamma$ .

By Lemma 6.2.3 on  $G$ , with  $A = \mathcal{L} \setminus \mathcal{L}_x, B = \mathcal{C}_x, B' = F$ ,

$$\left| \Pr_{\ell, C \sim N(\ell) \cap F}[(\ell, C) \in E'] - \Pr_{C \sim F, \ell \sim C}[(\ell, C) \in E'] \right| \leq \frac{\lambda}{\sqrt{\varepsilon}} \leq \gamma,$$

which means that

$$\Pr_\ell[\exists C \text{ s.t. } (\ell, C) \in E'] \geq \Pr_{\ell, C \sim N(\ell) \cap F}[(\ell, C) \in E'] \geq 1 - 2\gamma.$$

This means that for  $1 - 2\gamma$  of the lines in  $\mathcal{L}$ ,  $f$  agrees with a degree  $d$  function on  $1 - 40\gamma$  fraction of the points of each line.

We are very close to being able to apply the low degree test of Rubinfeld and Sudan [RS96], that works in the high soundness regime. For this, we need to move to neighborhoods. For  $y, h \in \mathbb{F}^m$ , we define the neighborhood of  $y, h$ ,

$$N_{y,h} = \{y + i(h - y) \mid 0 \leq i \leq d + 1\}.$$

Notice that  $N_{y,h} \subset \ell(y, h)$ . We show that on almost all of the neighborhoods  $N_{y,h}$ , the function  $f|_{N_{y,h}}$  equals a degree  $d$  polynomial, by showing that for almost all  $N_{y,h}$ , there exists some cube  $C$  such that  $f|_{N_{y,h}} = T(C)|_{N_{y,h}}$  ( $T(C)$  is a degree  $d$  polynomial).

Picking a random neighborhood  $N_{y,h}$  is equivalent to picking a random line  $\ell \in \mathcal{L}$  and then uniform  $y, h \in \ell$ . We have already showed that almost all lines  $\ell \in \mathcal{L}$ , there exists a cube  $C$  such that  $T(C)_\ell \stackrel{\Omega(\gamma)}{\approx} f|_\ell$ .

Now we can bound the same probability over neighborhoods

$$\begin{aligned}
& \Pr_{y, h \sim \mathbb{F}^m} [\exists C \text{ s.t. } f(N_{y,h}) = T(C)(N_{y,h})] \\
& \geq \Pr_{\ell} [\exists C \text{ s.t. } (\ell, C) \in E'] \cdot \Pr_{\ell, y, h \sim \ell} [f(N_{y,h}) = T(C)(N_{y,h}) \mid \exists C \text{ s.t. } (\ell, C) \in E'] \\
& \geq (1 - 2\gamma) \Pr_{\ell, y, h \sim \ell} [f(N_{y,h}) = T(C)(N_{y,h}) \mid \exists C \text{ s.t. } (\ell, C) \in E'] \\
& \geq (1 - 2\gamma)(1 - (d + 2) \cdot 40\gamma), \\
& \geq 1 - 42d\gamma,
\end{aligned} \tag{6.3.3}$$

where (6.3.3) is due to union bound on the neighborhoods inside  $\ell$ . Therefore, the function  $f$  equals a degree  $d$  polynomial on  $(1 - 42d\gamma)$  of the neighborhoods. Since  $\gamma \leq 100(d + 2)^{-3}$ , by [Theorem 6.3.12](#), we get that there exists a degree  $d$  polynomial  $g$ , such that  $f \stackrel{84d\gamma}{\approx} g$ .  $\blacksquare$

#### Proof of [Theorem 6.3.11](#):

Fix the cubes table  $T$ , and let  $f : \mathbb{F}^m \rightarrow \mathbb{F}$  be the function promised from [Lemma 6.3.6](#). This function satisfies the conditions of [Claim 6.3.13](#), so there exists a degree  $d$  polynomial  $g$  such that  $f \stackrel{84d\gamma}{\approx} g$ .

Since  $g$  is a degree  $d$  polynomial, for every cube  $C$  either  $T(C) = g|_C$ , or else they are very different. Let  $G$  be the inclusion graph  $G = G(\mathbb{F}^m, \mathcal{C})$ , and let

$$F = \{C \in \mathcal{C} \mid T(C) \stackrel{32\gamma}{\approx} f|_C\}$$

From [Lemma 6.3.6](#), the measure of  $F$  is at least  $\frac{\varepsilon}{16}$ , let  $A'$  be the set of points on which  $f \neq g$ . By [Lemma 6.2.6\(4\)](#),  $\lambda(G) \leq q^{-\frac{3}{2}}$ . We use [Lemma 6.2.3](#) on  $G$  with  $A = \mathbb{F}^m, B = \mathcal{C}, B' = F$ ,

$$\left| \Pr_{C \in F, y \in N(C)} [y \in A'] - \Pr_{y, C \in N(y) \cap F} [y \in A'] \right| \leq \frac{q^{-\frac{3}{2}}}{\varepsilon} \leq \gamma$$

We know that  $\Pr_{y, C \in N(y) \cap F} [y \in A'] \leq \Pr_y [y \in A'] \leq 84d\gamma$ , which implies that

$$\Pr_{C \in F, y \in N(C)} [y \in A'] \leq 85d\gamma.$$

By averaging, for at least half of the cubes  $C \in F$ ,  $\Pr_{y \in C} [y \in A'] \leq 200d\gamma \leq \frac{1}{2}$ . For all these cubes  $T(C) = g|_C$ , because  $\Pr_{y \in C} [T(C)(y) = g(y)] \geq \Pr_{y \in C} [T(C)(y) =$

$f(y), y \notin A'] \geq 1 - 32\gamma - \frac{1}{2} > d/q$ , and since  $g|_C, T(C)$  are both degree  $d$  polynomials, they must be equal.  $\blacksquare$

**Remark 6.3.14.** *Instead of Theorem 6.3.12, we can use another similar characterization from [RS96], where the neighborhood is defined as  $N_{y,h} = \{y + i(h - y) \mid i \in \{0, \dots, 10d\}\}$ . The advantage of using this new neighborhood is that we can conclude  $f^{(1+o(1))\delta} \approx g$  as long as  $\delta = O(1/d)$ . This will help in reducing the exponent of  $d$  by 1 in our main theorem. We chose to use Theorem 6.3.12 for a self contained proof.*

## 6.4 Comparing between different tests and their agreement parameter

There are many variants for the low degree test, in this section we look into equivalences between similar low degree agreement tests. We first prove the equivalence in a more general setting and as a corollary we get some interesting results.

Throughout this section, we will work over  $\mathbb{F}^m$  where  $\mathbb{F}$  is a field of size  $q$  and let  $s \leq m/2$  be fixed. Also, let  $T$  denotes a table which maps every  $s$  dimensional affine subspace in  $\mathbb{F}^m$  to a degree  $d$  polynomial. Let  $\mathcal{A}^s$  denote the set of all  $s$  dimensional affine subspaces in  $\mathbb{F}^m$ . For  $r < s$  and for  $R \in \mathcal{A}^r$  let  $\mathcal{A}_R^s \subseteq \mathcal{A}^s$  denote all subspaces in  $\mathcal{A}^s$  which contain a particular subspace  $R$ ,

$$\mathcal{A}_R^s = \{S \subset \mathbb{F}^m \mid \dim(S) = s, R \subseteq S\}.$$

For parameters  $s > k \geq r$  consider the following test:

---

**Test 3** Subspace agreement test :  $\alpha_{sks(r)}$

---

1. Select  $K \in \mathcal{A}^k$  u.a.r.
2. Pick  $S_1, S_2 \in \mathcal{A}_K^s$  u.a.r.
3. Pick a  $r$  dimensional subspace  $R \subseteq K$  u.a.r.
4. Accept iff  $T(S_1)|_R = T(S_2)|_R$ .

Let  $\alpha_{sks(r)}(T)$  be the *agreement* of the table  $T = (f_S)_{S \in \mathcal{A}^s}$ , i.e. the probability of acceptance of the test.

---

When  $r = k$  we simply denote the agreement as  $\alpha_{sks}(T)$ . With these notations, the success probability of [Test 1](#) is denoted by  $\alpha_{3,0,3}(T)$ , and of [Test 2](#) by  $\alpha_{2,1,2}(T)$ .

In this section, we prove the following main lemma.

**Lemma 6.4.1.** *Let  $0 \leq r < k < s \leq \frac{m}{2}$ , we have*

$$\alpha_{srs}(T) \left(1 - \left(\frac{d}{q}\right)^{r+1}\right) \leq \alpha_{sks}(T) \leq \alpha_{srs}(T) + (1 + o(1))q^{-(s-2k+r+1)},$$

From [Lemma 6.4.1](#), we can deduce the following corollary,

**Corollary 6.4.2.** *Let  $\alpha_{c\ell c}(T) = \alpha_{3,1,3}(T)$  be the success probability of [Test 3](#) with  $s = 3, k = r = 1$ , i.e checking consistency of two cubes that intersect on a line. Then for every cubes table  $T$ ,*

$$\alpha_{cxc}(T) \left(1 - \frac{d}{q}\right) \leq \alpha_{c\ell c}(T) \leq \alpha_{cxc}(T) + \frac{1}{q^2}(1 + o(1)).$$

The corollary implies that [Theorem 6.1.2](#) holds if we modify the test as selecting two cubes u.a.r from a pair of cubes intersecting in a line and checking consistency on the whole line.

Using [Lemma 6.4.1](#), we can also compare the Raz-Safra Plane vs. Plane agreement tests where planes intersect at a point and on a line. Recall that  $\alpha_{\ell\mathcal{P}}(T)$  is the acceptance probability of [Test 2](#). Invoking [Lemma 6.4.1](#) with  $s = 2, k = 1$  and  $r = 0$ , we get the following corollary.

**Corollary 6.4.3** ([Lemma 6.1.4](#) restated). *Let  $T$  be a planes table, and let  $\alpha_{x\mathcal{P}}(T)$  be the success probability of [Test 3](#) with  $s = 2, k = r = 0$ , i.e two planes that intersects on a point. Let  $\alpha_{\ell\mathcal{P}}(T)$  be the success probability of [Test 2](#) from the introduction (two planes that intersects on a line), then*

$$\alpha_{x\mathcal{P}}(T) \left(1 - \frac{d}{q}\right) \leq \alpha_{\ell\mathcal{P}}(T) \leq \alpha_{x\mathcal{P}}(T) + \frac{1}{q}(1 + o(1)).$$

#### 6.4.1 Proof of [Lemma 6.4.1](#)

We prove a few claims that together with the observation  $\alpha_{sks(r)}(T) \geq \alpha_{sks}(T)$ , prove the lemma.

The following claim shows that two distinct low degree polynomials agree on a random subspace of fixed dimension with very small probability.

**Claim 6.4.4.** *Let  $P_1, P_2 : \mathbb{F}^t \rightarrow \mathbb{F}$  be two distinct degree  $d$  polynomials. For  $r \leq t$*

$$\Pr_{R \in \mathcal{A}^r} [(P_1)|_R \equiv (P_2)|_R] \leq \left(\frac{d}{q}\right)^{r+1}.$$

*Proof.* Consider the following way of choosing an  $r$  dimensional *affine* subspace from  $\mathcal{A}^r$  uniformly at random: Pick  $x_0, x_1, x_2, \dots, x_r$  from  $\mathbb{F}_q^t$  independently and u.a.r. Then pick a  $r$  dimensional affine subspace  $R$  containing  $\{x_0 + \text{span}(x_1, x_2, \dots, x_r)\}$  u.a.r ( $R$  is determined by  $x_0, x_1, x_2, \dots, x_r$ , unless  $\dim \text{span}(x_1, x_2, \dots, x_r) < r$ ). It is easy to see that  $R$  is distributed uniformly in  $\mathcal{A}^r$ . Now,  $P_1$  and  $P_2$  agreeing on the whole subspace  $R$  implies that they agree on the points  $\{x_0, x_0 + x_1, x_0 + x_2, \dots, x_0 + x_r\}$  as all these points are contained in  $R$ . Therefore,

$$\begin{aligned} \Pr_{R \in \mathcal{A}^r} [(P_1)|_R \equiv (P_2)|_R] &\leq \Pr_{x_0, x_1, \dots, x_r \sim \mathbb{F}^t} [P_1(x_0) = P_2(x_0) \wedge_{i=1}^r P_1(x_0 + x_i) = P_2(x_0 + x_i)] \\ &= \left( \Pr_{x \in \mathbb{F}_q^t} [P_1(x) = P_2(x)] \right)^{r+1} \leq \left(\frac{d}{q}\right)^{r+1}, \end{aligned}$$

where the last inequality is because two different degree  $d$  polynomial agree on at most  $\frac{d}{q}$  fraction of the points (Schwartz-Zippel lemma). ■

**Claim 6.4.5.** *Let  $M_{m \times n}$  be the adjacency matrix of a bi regular bipartite graph  $G$ , and let  $f$  be a  $n$ -dimensional  $\{0, 1\}$  vector such that  $\mathbf{E}[f] = \mu$ . Then*

$$\langle Mf, Mf \rangle \leq \mu^2 + \lambda(G)^2 \mu.$$

*Proof.* Let  $\mathbf{1}$  be the unit vector. We write  $f$  as  $f = f_1 + f_1^\perp$  where  $f_1$  is in the direction of  $\mathbf{1}$ , the singular vector with the maximal singular value, and  $f_1^\perp$  is its orthogonal component. We note that  $f_1 = \mu \mathbf{1}$ , and hence  $\langle f_1, f_1 \rangle = \mu^2$ . Also,

$$\mu = \langle f, f \rangle = \langle f_1 + f_1^\perp, f_1 + f_1^\perp \rangle = \langle f_1, f_1 \rangle + \langle f_1^\perp, f_1^\perp \rangle \geq \langle f_1^\perp, f_1^\perp \rangle.$$

Using this we can bound:

$$\begin{aligned}
\langle Mf, Mf \rangle &= \langle Mf_1 + Mf_1^\perp, Mf_1 + Mf_1^\perp \rangle \\
&= \langle f_1, f_1 \rangle + \langle Mf_1^\perp, Mf_1^\perp \rangle \\
&\leq \mu^2 + \lambda(G)^2 \langle f_1^\perp, f_1^\perp \rangle \\
&\leq \mu^2 + \lambda(G)^2 \mu.
\end{aligned}$$

■

**Claim 6.4.6.**  $\alpha_{sks(r)}(T) \geq \alpha_{srs}(T)$ .

*Proof.* We start by fixing  $R \in \mathcal{A}^r, \sigma \in \mathbb{F}^{q^r}$ . For each  $k$  dimensional subspace  $K \in \mathcal{A}_R^k$ , denote by  $p_K$  the following probability  $p_K = \Pr_{S \sim \mathcal{A}_K^s} [T(S)|_R \equiv \sigma]$ . In this notation

$$\begin{aligned}
\Pr_{\substack{K \sim \mathcal{A}_R^k \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \equiv T(S_2)|_R \equiv \sigma] &= \mathbf{E}_K [p_K^2] \\
&\geq \left( \mathbf{E}_K [p_K] \right)^2 = \Pr_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)|_R \equiv T(S_2)|_R \equiv \sigma].
\end{aligned} \tag{6.4.1}$$

Now, we average over  $R, \sigma$  to get  $\alpha_{srs}(T)$  and  $\alpha_{sks(r)}(T)$ :

$$\begin{aligned}
\alpha_{srs}(T) &= \Pr_{\substack{R \sim \mathcal{A}^r \\ S_1, S_2 \sim \mathcal{A}_R^s}} [T(S_1)|_R \equiv T(S_2)|_R] \\
&= \mathbf{E}_{R \sim \mathcal{A}^r} \left[ \sum_{\sigma \in \mathbb{F}^{q^r}} \Pr_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)|_R \equiv T(S_2)|_R \equiv \sigma] \right].
\end{aligned} \tag{6.4.2}$$

Picking a uniform  $R \in \mathcal{A}^r$  then  $K \in \mathcal{A}_R^k$  is the same as picking  $K \in \mathcal{A}^k$  and then a random  $r$  dimensional subspace  $R$  in  $K$ , so by definition

$$\begin{aligned}
\alpha_{sks(r)}(T) &= \Pr_{\substack{R \sim \mathcal{A}^r, K \sim \mathcal{A}_R^k \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \equiv T(S_2)|_R] \\
&= \mathbf{E}_{R \sim \mathcal{A}^r} \left[ \sum_{\sigma \in \mathbb{F}^{q^r}} \Pr_{\substack{K \sim \mathcal{A}_R^k \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \equiv T(S_2)|_R \equiv \sigma] \right].
\end{aligned} \tag{6.4.3}$$

Using (6.4.1), (6.4.2) and (6.4.3), we get  $\alpha_{sks(r)}(T) \geq \alpha_{srs}(T)$ . ■

**Claim 6.4.7.**  $\alpha_{sks}(T) \geq \alpha_{sks(r)}(T) \left( 1 - \left( \frac{d}{q} \right)^{r+1} \right)$ .

*Proof.* By the definition of the agreement,

$$\alpha_{sks}(T) = 1 - \mathbf{E}_{K \sim \mathcal{A}^k} \left[ \Pr_{S_1, S_2 \sim \mathcal{A}_K^s} [T(S_1)|_K \neq T(S_2)|_K] \right],$$

and

$$\alpha_{sks(r)}(T) = 1 - \mathbf{E}_{K \sim \mathcal{A}^k} \left[ \Pr_{\substack{R \sim K, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \neq T(S_2)|_R] \right],$$

where we use  $R \sim K$  to denote a random  $r$  dimensional subspace in  $K$ . For every subspace  $K \in \mathcal{A}^k$ ,  $R \subseteq K$  is uniform and is independent of  $S_1, S_2$ .

$$\begin{aligned} \Pr_{\substack{R \sim K, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \neq T(S_2)|_R] &= \Pr_{\substack{R \sim K, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_K \neq T(S_2)|_K, T(S_1)|_R \neq T(S_2)|_R] \\ &= \Pr_{S_1, S_2 \sim \mathcal{A}_K^s} [T(S_1)|_K \neq T(S_2)|_K] \cdot \\ &\quad \Pr_{\substack{R \sim K, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \neq T(S_2)|_R \mid T(S_1)|_K \neq T(S_2)|_K] \\ &\geq \Pr_{S_1, S_2 \sim \mathcal{A}_K^s} [T(S_1)|_K \neq T(S_2)|_K] \cdot \left( 1 - \left( \frac{d}{q} \right)^{r+1} \right). \end{aligned}$$

The lower bound on the probability in the last inequality is as follows: the event  $T(S_1)|_K \neq T(S_2)|_K$  implies that the degree  $d$  polynomials corresponding to  $T(S_1)|_K$  and  $T(S_2)|_K$  are distinct. Thus, using [Claim 6.4.4](#)  $\Pr_{R \sim K} [T(S_1)|_R \equiv T(S_2)|_R] \leq (d/q)^{r+1}$ . Therefore, for a  $k$  dimensional subspace  $K \in \mathcal{A}^k$ ,

$$\Pr_{\substack{R \sim K, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \neq T(S_2)|_R] \geq \Pr_{S_1, S_2 \sim \mathcal{A}_K^s} [T(S_1)|_K \neq T(S_2)|_K] \left( 1 - \left( \frac{d}{q} \right)^{r+1} \right).$$

Finally, taking the expectation of the inequality over  $K$  finishes the proof.  $\blacksquare$

We first state a lemma about an expansion of the kind of inclusion graphs which we will be dealing with in analyzing the [Test 3](#), the proof of which appears in [Section 6.5](#).

**Lemma 6.4.8.** *Let  $r \leq k < s \leq \frac{m}{2}$  be integers, and let  $G$  be the inclusion graph  $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$  for a  $r$  dimensional subspace  $R$ , where  $R \neq \emptyset$ . Then,*

$$\lambda(G)^2 \leq (1 + o(1)) \cdot q^{-(s-2k+r+1)}.$$

**Claim 6.4.9.**  $\alpha_{sks(r)}(T) \leq \alpha_{srs(r)}(T) + \lambda(G)^2$  where  $G$  is the inclusion graph  $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$  for an  $r$  dimensional subspace  $R$ .

*Proof.* Fix an  $r$  dimensional affine subspace  $R \in \mathcal{A}^r$ . We prove the following inequality:

$$\Pr_{\substack{K \sim \mathcal{A}_R^k, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \equiv T(S_2)|_R] \leq \Pr_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)|_R \equiv T(S_2)|_R] + \lambda(G)^2, \quad (6.4.4)$$

Note that this implies the claim if we take expectation over  $R \in \mathcal{A}^r$ . Towards proving (6.4.4), for each value  $\sigma \in \mathbb{F}^{q^k}$ , denote by  $A_\sigma \subseteq \mathcal{A}_R^s$  the following set

$$A_\sigma = \{S \in \mathcal{A}_R^s \mid T(S)|_R \equiv \sigma\},$$

and  $\mu_\sigma = \frac{|A_\sigma|}{|\mathcal{A}_R^s|}$ . Let  $f_\sigma$  be the indicator function for  $A_\sigma$ , for  $S \in A_\sigma$ ,  $f_\sigma(S) = 1$ . By definition

$$\Pr_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)|_R \equiv T(S_2)|_R] = \sum_{\sigma} \mu_\sigma^2. \quad (6.4.5)$$

Let  $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$  be the inclusion graph, and denote by  $M \in \mathbb{R}^{|\mathcal{A}_R^k| \times |\mathcal{A}_R^s|}$  the normalized adjacency matrix, such that each entry is either 0 or  $\frac{1}{\deg(K)}$  where  $K \in \mathcal{A}_R^k$ .

For each  $k$  dimensional subspace  $K \in \mathcal{A}_R^k$ , the value  $(Mf_\sigma)_K$  is the fraction of  $K$ 's neighbors in  $A_\sigma$ ,  $(Mf_\sigma)_K = \Pr_{S \sim \mathcal{A}_K^s} [S \in A_\sigma]$ . Therefore, the inner product gives us the expected value:

$$\langle Mf_\sigma, Mf_\sigma \rangle = \mathbf{E}_{K \in \mathcal{A}_R^k} \left[ \mathbf{E}_{S \in \mathcal{A}_K^s} [S \in A_\sigma]^2 \right] = \mathbf{E}_{K \in \mathcal{A}_R^k} \left[ \mathbf{E}_{S_1, S_2 \in \mathcal{A}_K^s} [S_1, S_2 \in A_\sigma] \right].$$

Therefore

$$\begin{aligned} \Pr_{\substack{K \sim \mathcal{A}_R^k, \\ S_1, S_2 \sim \mathcal{A}_K^s}} [T(S_1)|_R \equiv T(S_2)|_R] &= \sum_{\sigma} \langle Mf_\sigma, Mf_\sigma \rangle \\ &\leq \sum_{\sigma} \mu_\sigma^2 + \lambda(G)^2 \mu_\sigma \quad (\text{using Claim 6.4.5}) \\ &= \Pr_{S_1, S_2 \sim \mathcal{A}_R^s} [T(S_1)|_R \equiv T(S_2)|_R] + \lambda(G)^2. \quad (\text{from (6.4.5)}) \end{aligned}$$

which proves (6.4.4). ■

Claim 6.4.9 together with Lemma 6.4.8 gives us  $\alpha_{sks}(T) \leq \alpha_{srs}(T) + (1+o(1))q^{-2(s-2k+r+1)}$ . Claim 6.4.6 and Claim 6.4.7 prove the other inequality,  $\alpha_{srs}(T) \left(1 - \left(\frac{d}{q}\right)^{r+1}\right) \leq \alpha_{sks}(T)$ .

## 6.5 Spectral properties of Certain Inclusion Graphs

Let  $G_{s,k}$  be the intersection graph where the vertex set is all *linear subspaces* of dimension  $s$  in  $\mathbb{F}_q^m$  and  $U \sim U'$  iff  $\dim(U \cap U') = k$ . We will use the  $T_{s,k}$  to denote the *Markov operator* associated with a random walk on this graph. We will need following fact about eigenvalues of  $T_{k,k-1}$ .

**Definition 6.5.1.**  $k$ -th  $q$ -ary Gaussian binomial coefficient  $\begin{bmatrix} m \\ k \end{bmatrix}_q$  is given by

$$\begin{bmatrix} m \\ k \end{bmatrix}_q := \prod_{i=0}^{k-1} \frac{q^m - q^i}{q^k - q^i}.$$

As  $q$  is fixed throughout the article, we will omit the subscript from now on.

**Fact 6.5.2.** ([BCN89, Theorem 9.3.3]) Suppose  $1 \leq k \leq \frac{m}{2}$ ,

1. The number of  $k$  dimensional linear subspaces in  $\mathbb{F}_q^m$  is exactly  $\begin{bmatrix} m \\ k \end{bmatrix}$ .
2. The degree of  $G_{k,k-1}$  is  $q \begin{bmatrix} k \\ 1 \end{bmatrix} \begin{bmatrix} m-k \\ 1 \end{bmatrix}$ .
3. The eigen values of  $T_{k,k-1}$  are

$$\lambda_j(T_{k,k-1}) = \frac{q^{j+1} \begin{bmatrix} k-j \\ 1 \end{bmatrix} \begin{bmatrix} m-k-j \\ 1 \end{bmatrix} - \begin{bmatrix} j \\ 1 \end{bmatrix}}{q \begin{bmatrix} k \\ 1 \end{bmatrix} \begin{bmatrix} m-k \\ 1 \end{bmatrix}},$$

with multiplicities  $\begin{bmatrix} m \\ j \end{bmatrix} - \begin{bmatrix} m \\ j-1 \end{bmatrix}$  for  $j = 0, 1, \dots, k$ . Asymptotically,  $\lambda_j(T_{k,k-1}) = \Theta(q^{-j})$ .

**Claim 6.5.3.** For any  $1 \leq k \leq \frac{m}{2}$  and , we have  $|\lambda_1(T_{k,k-2}) - \lambda_1(T_{k,k-1})^2| = (1 + o(1)) \frac{1}{q^k}$ .

*Proof.* Consider a two-step random walk on the graph  $G_{k,k-1}$ . We will show that with very high probability, a two-step random walk on  $G_{k,k-1}$  corresponds to a single step random walk on  $G_{k,k-2}$ . Let  $U_1, U_2, U_3$  be the vertices from a two-step random walk on  $G_{k,k-1}$ . Note that conditioned on the event  $\dim(U_1 \cap U_3) = k-2$ , the distribution of  $(U_1, U_3)$  is exactly same as a single step random walk on  $G_{k,k-2}$ . We will upper bound the probability of the event  $\dim(U_1 \cap U_3) \neq k-2$ .

Let  $w_1 = U_1 \cap U_2$  and  $w_2 = U_2 \cap U_3$ , we can describe the distribution of the two-step random walk as follows:

1. Choose a uniform  $k$  dimensional subspace  $U_2$ .
2. Choose two random  $k - 1$  dimensional subspaces,  $w_1, w_2 \subset U_2$ .
3. Choose a point  $x_1 \in \mathbb{F}^m \setminus U_2$ , and set  $U_1 = \text{span}(w_1, x_1)$ .
4. Choose a point  $x_2 \in \mathbb{F}^m \setminus U_2$ , and set  $U_3 = \text{span}(w_2, x_2)$ .

By definition,  $U_2$  has  $\binom{k}{k-1}$  subspaces of size  $k - 1$ , therefore  $\Pr_{w_1, w_2}[w_1 = w_2] = \frac{1}{\binom{k}{k-1}}$ . In order to satisfy  $\dim(U_1 \cap U_3) \neq k - 2$  given that  $w_1 \neq w_2$ , the point  $x_2$  should be in  $U_1$ . There are  $q^k - q^{k-1}$  points in  $U_1 \setminus U_2$ , and therefore this probability equals  $\frac{|U_1 \setminus U_2|}{|\mathbb{F}^m \setminus U_2|} = \frac{q^k - q^{k-1}}{q^m - q^k}$ .

$$\begin{aligned}
\Pr[\dim(U_1 \cap U_3) \neq k - 2] &= \Pr[w_1 = w_2] + \Pr[\dim(U_1 \cap U_3) \neq k - 2 \wedge w_1 \neq w_2] \\
&= \frac{1}{\binom{k}{k-1}} + \left(1 - \frac{1}{\binom{k}{k-1}}\right) \Pr[\dim(U_1 \cap U_3) \neq k - 2 \mid w_1 \neq w_2] \\
&= \frac{1}{\binom{k}{k-1}} + \left(1 - \frac{1}{\binom{k}{k-1}}\right) \cdot \frac{q^k - q^{k-1}}{q^m - q^k} =: \beta.
\end{aligned}$$

Thus, we have

$$T_{k,k-1}^2 = \beta \mathcal{N} + (1 - \beta) T_{k,k-2},$$

where  $\mathcal{N}$  is a Markov operator corresponding to the two-step random walk on  $G_{k,k-1}$ , conditioning on  $\dim(U_1 \cap U_3) \neq k - 2$ . The claim follows as  $\beta = (1 + o(1))1/q^k$ . ■

Following fact follows from the definition of  $\lambda(G)$ .

**Fact 6.5.4.** *For a bi-regular bipartite graph  $G(A, B)$ , if  $T$  is a Markov operator associated with a random walk of length two starting from  $A$  (or  $B$ ) then  $\lambda(G)^2 = \lambda(T)$ .*

We now prove [Lemma 6.2.6](#).

**Lemma 6.5.5** (Restatement of [Lemma 6.2.6](#)). *We have for every  $m \geq 6$ ,*

1. For  $G_1(\mathcal{L} \setminus \mathcal{L}_x, \mathcal{C}_x)$ ,  $\lambda(G_1) \approx \frac{1}{\sqrt{q}}$ .
2. For  $G_2(\mathcal{L}_x, \mathcal{C}_x)$ ,  $\lambda(G_2) \approx \frac{1}{q}$ .
3. For  $G_3(\mathbb{F}^m \setminus \ell, \mathcal{C}_\ell)$ ,  $\lambda(G_3) \approx \frac{1}{\sqrt{q}}$ .

4. For  $G_4(\mathbb{F}^m, \mathcal{C})$ ,  $\lambda(G_4) \approx \frac{1}{q^{3/2}}$ .

5. For  $G_5(\mathbb{F}^m \setminus \{x\}, \mathcal{C}_x)$ ,  $\lambda(G_5) \approx \frac{1}{q}$ .

And for every  $m \geq 3$

6. For  $G_6(\mathbb{F}^m, \mathcal{L})$ ,  $\lambda(G_6) \approx \frac{1}{\sqrt{q}}$ .

where  $\approx$  denotes equality up to a multiplicative factor of  $1 \pm o(1)$ .

*Proof.* Suppose  $T$  is an  $n \times n$  Markov operator which is a convex combination of a bunch of other Markov operators:  $T = \sum_{i=1}^k \alpha_i T_i$  where  $\alpha_i \geq 0$  and  $\sum_{i=1}^k \alpha_i = 1$ , and that both  $T$  and  $T_i$ 's are regular. As the row sum of each Markov operator is 1, the largest eigenvalue is 1, since both  $T$  and  $T_i$ 's are regular, the eigenvector of the largest eigenvalue is the all 1 vector. The second largest eigenvalue of  $T$  can be upper bounded by

$$\begin{aligned} \lambda(T) &:= \max_{\substack{v \in \mathbb{R}^n, \|v\|=1, \\ v \perp \mathbf{1}}} \|Tv\| \\ &= \max_{\substack{v \in \mathbb{R}^n, \|v\|=1, \\ v \perp \mathbf{1}}} \left\| \sum_{i=1}^k \alpha_i T_i \right\| \\ &\leq \sum_{i=1}^k \max_{\substack{v \in \mathbb{R}^n, \|v\|=1, \\ v \perp \mathbf{1}}} \|\alpha_i T_i\| = \sum_{i=1}^k \alpha_i \lambda(T_i). \end{aligned}$$

In proving the lemma, we repeatedly use the above simple fact to upper bound the eigenvalue.

1. Without loss of generality, we can assume  $x = \mathbf{0}$ . Let  $d_L$  and  $d_R$  denote the left and right degree of  $G_1$  respectively. Fix a line  $\ell$ ,  $d_L$  is the number of cubes containing  $\ell$  and not passing through  $\mathbf{0}$ . Every point  $x \notin \text{span}(\ell, \mathbf{0})$  defines a cube  $C = \text{span}(x, \mathbf{0}, \ell)$ . Thus, the number of linear cubes containing  $\ell$  equals  $d_L = \frac{q^m - q^2}{q^3 - q^2}$ , where the denominator is the overcounting factor, the number of points that give the same cube.

Fix a linear cube  $C$ . The right degree is the number of lines in  $C$  not passing through the origin which is  $\frac{\binom{q^3}{2}}{\binom{q}{2}} - \frac{q^3 - 1}{q - 1}$ , where the first term counts all possible lines

in  $C$  (each two different points define a line, we divide by the double counting) and the second term counts all the lines in  $C$  that pass through the origin.

Let  $T_1$  be the Markov operator associated with a two-step random walk in  $G_1$  starting from  $C_x$ . Using [Fact 6.5.4](#), in order to bound  $\lambda(G_1)$  it is enough to bound the second largest eigenvalue of  $T_1$ . Since  $G_1$  is bi-regular, the first eigenvector of  $T_1$  is the all ones vector. For every cube  $C$ , the number of two-step walks starting from  $C$  is  $d_L \cdot d_R$ .

If  $\dim\{C_1 \cap C_2\} = 1$ , then the two cubes intersection is only on a line. Since both cubes are linear, it means that this line goes through the origin, therefore it doesn't correspond to a vertex on the left side, and there is no walk  $C_1 \rightarrow \ell \rightarrow C_2$ , so  $(T_1)_{C_1, C_2} = 0$ . Of course, the same holds if  $\dim\{C_1 \cap C_2\} = 0$ .

If  $\dim\{C_1 \cap C_2\} = 2$ , there there is a plane going through the origin in both  $C_1, C_2$ . The number of walks  $C_1 \rightarrow \ell \rightarrow C_2$  equals the number of lines in this plane that don't contain the origin,  $\mathbf{0}$ . Each pair of distinct points on the plane correspond to a line, and we divide by the double counting. Therefore the number of lines in a plane equals  $\frac{\binom{q^2}{2}}{\binom{q}{2}}$ . We subtract from it the number of lines in a plane that contains  $\mathbf{0}$ , resulting in  $\frac{\binom{q^2}{2}}{\binom{q}{2}} - \frac{q^2-1}{q-1} =: \beta$ .

If  $C_1 = C_2$ , then exists a path  $C_1 \rightarrow \ell \rightarrow C_2$  for every line  $\ell$  adjacent to  $C_1$ , and there are  $d_R$  such lines.

Since  $T_1$  is a Markov operator, we need to normalize the number of paths between  $C_1, C_2$  by dividing in the total number of outgoing paths from  $C_1$ , which equals  $d_R \cdot d_L$ . Therefore,

$$(T_1)_{C_i, C_j} = \begin{cases} \frac{d_R}{d_R \cdot d_L}, & \text{if } C_i = C_j \\ \frac{\beta}{d_R \cdot d_L}, & \text{if } \dim\{C_1 \cap C_2\} = 2 \\ 0, & \text{otherwise} \end{cases} \quad (6.5.1)$$

Thus, we can write  $T_1$  as:

$$T_1 = \frac{1}{d_L} I + \frac{\beta}{d_R d_L} \cdot G_{3,2} = \frac{1}{d_L} I + \frac{\beta d'}{d_R d_L} \cdot T_{3,2},$$

where  $d'$  is the degree of a vertex in  $G_{3,2}$ . One can verify that  $T_1$  is indeed a convex combination of two Markov operators  $I$  and  $T_{3,2}$ . Since  $G_{3,2}$  is a regular graph, the second eigenvector of  $T_{3,2}$  is also orthogonal to  $\mathbf{1}$ . Hence,

$$\begin{aligned}\lambda(G_1)^2 = \lambda(T_1) &= \max_{\substack{v \in \mathbb{R}^{|\mathcal{C}_x|}, v \perp \mathbf{1} \\ \|v\|=1}} \|T_1 v\| = \max_{\substack{v \in \mathbb{R}^{|\mathcal{C}_x|}, v \perp \mathbf{1} \\ \|v\|=1}} \left\| \left( \frac{1}{d_L} I + \frac{\beta d'}{d_R d_L} \cdot T_{3,2} \right) v \right\| \\ &= \frac{1}{d_L} + \frac{\beta d'}{d_R d_L} \cdot \lambda_1(T_{3,2}).\end{aligned}\tag{6.5.2}$$

We now just need to plug in the values of  $\beta, d'$  and  $\lambda_1(T_{3,2})$ . Using [Fact 6.5.2](#),  $\lambda_1(T_{3,2})$  is given by the following expression,

$$\lambda_1(T_{3,2}) = \frac{q^2 \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} m-4 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}}{q \begin{bmatrix} 2 \\ 1 \end{bmatrix} \begin{bmatrix} m-3 \\ 1 \end{bmatrix}} = (1 + o(1)) \frac{1}{q}.$$

As we have seen before,  $d_R = \frac{\binom{q^3}{2}}{\binom{q}{2}} - \frac{q^3-1}{q-1} = (1 + o(1))q^4$ ,  $d_L = \frac{q^m - q^2}{q^3 - q^2} = (1 + o(1))q^{m-3}$  and  $\beta = \frac{\binom{q^2}{2}}{\binom{q}{2}} - \frac{q^2-1}{q-1} = (1 + o(1))q^2$ . From [Fact 6.5.2](#),  $d' = (1 + o(1))q^{m-1}$ . Thus,

$$\frac{1}{d_L} = (1 + o(1)) \frac{1}{q^{m-3}}, \quad \frac{\beta d'}{d_R d_L} \lambda_1(T_{3,2}) = (1 + o(1)) \frac{1}{q}$$

Plugging these values in [\(6.5.2\)](#) gives  $\lambda(G_1) = (1 + o(1)) \frac{1}{\sqrt{q}}$  as required.

2. This bound is implied from a more general [Lemma 6.4.8](#) we prove below with  $s = 3, k = 1$  and  $r = 0$ .
3. In this case, it will be easier to bound the eigenvalue of the Markov operator associated with a random walk of length two starting from  $\mathbb{F}^m \setminus \ell$ . Let  $T_3$  be the Markov operator. Now, the path of length two starting from  $x$  looks like  $x \rightarrow C \rightarrow y$ . Thus, the cube  $C$  contains all points from the affine plane spanned by  $x$  and  $\ell$ . Let  $p(x, \ell)$  be the affine plane spanned by  $x$  and  $\ell$ . We have  $\Pr[y \in p(x, \ell)] = \frac{q^2 - q}{q^3 - q} \approx \frac{1}{q}$ . If  $y \notin p(x, \ell)$  then the distribution of  $y$  is uniform in  $\mathbb{F}^m \setminus p(x, \ell)$ . Thus, we have

$$T_3 = (1 - o(1)) \left( 1 - \frac{1}{q} \right) J + (1 + o(1)) \frac{1}{q} \mathcal{N},$$

where  $J$  is a Markov operator associated with a complete graph on  $\mathbb{F}^m \setminus \ell$ , with self loops and  $\mathcal{N}$  is an appropriate Markov operator. Thus, we have bound  $\lambda(T_3) = (1 + o(1)) \frac{1}{q}$ . Since  $\lambda(G_3)^2 = \lambda(T_3)$ , the bound follows.

4. Proof of this is along the same lines as (3). The Markov operator here (starting a walk from the left side) can be written as

$$T_4 = (1 \pm o(1)) \frac{1}{q^3} I + \left( (1 \pm o(1)) \left(1 - \frac{1}{q^3}\right) \right) J,$$

where  $I$  is an identity matrix. Thus  $\lambda(T_4) = (1 \pm o(1)) \frac{1}{q^3} = \lambda(G_4)^2$ .

5. The proof of this item is also similar to (3), we look on the path of length 2 starting from the left side, i.e  $y \rightarrow C \rightarrow z$ , and let  $T_5$  be the Markov operator. Let  $\ell(x, y)$  be the line spanned by  $x, y$  (where  $x$  is the fixed point,  $G_5(\mathbb{F}^m \setminus \{x\}, \mathcal{C}_x)$ ), then  $\Pr[z \in \ell(x, y)] = \frac{|\ell(x, y) \setminus \{x\}|}{|C \setminus \{x\}|} = \frac{q-1}{q^3-1} \approx \frac{1}{q^2}$ , let  $\mathcal{N}$  be the appropriate Markov operator of the event that  $x, y, z$  are colinear, then

$$T_5 = (1 - o(1)) \left(1 - \frac{1}{q^2}\right) J + (1 + o(1)) \frac{1}{q^2} \mathcal{N}.$$

Here  $J$  is the Markov operator of the complete graph on  $\mathbb{F}^m \setminus \{x\}$ . Thus  $\lambda(G_5)^2 \approx \frac{1}{q^2}$ .

6. Consider a two-step random walk in  $G_6$ ,  $x \rightarrow \ell \rightarrow y$ . If we sample a random line through  $x$  then conditioned on  $y \neq x$ ,  $y$  is uniformly distributed in  $\mathbb{F}^m$ . Thus, we can write the Markov operator  $T$  associated with this process as:

$$T = \frac{1}{q} I + \left(1 - \frac{1}{q}\right) T',$$

where  $T'$  is a Markov operator associated with a random walk on a complete graph on  $A$ , without self loops and  $I$  is an identity matrix. As  $T' = \frac{1}{|A|-1} J - \frac{1}{|A|-1} I$ ,  $\lambda(T') = \frac{1}{q^3-1}$ . Thus,  $\left| \lambda(T) - \frac{1}{q} \right| \leq \frac{1}{q^3-1}$ . The claim follows as  $\lambda(G_6)^2 = \lambda(T)$ . ■

Next, we prove [Lemma 6.4.8](#). Recall that  $\mathcal{A}^s$  denotes set of all  $s$  dimensional affine subspaces in  $\mathbb{F}^m$ . Also, for  $r < s$  and for  $R \in \mathcal{A}^r$ ,  $\mathcal{A}_R^s \subseteq \mathcal{A}^s$  denotes all those subspaces in  $\mathcal{A}^s$  which contains a particular subspace  $R$ .

**Lemma 6.5.6** (Restatement of [Lemma 6.4.8](#)). *Let  $r \leq k < s \leq \frac{m}{2}$  be integers, and let  $G$  be the inclusion graph  $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$  for an  $r$  dimensional subspace  $R$ , where  $R \neq \emptyset$ . Then,*

$$\lambda(G)^2 \leq (1 + o(1)) \cdot q^{-(s-2k+r+1)}.$$

*Proof.* Fix an  $r$  dimensional subspace  $R \subseteq \mathbb{F}^m, R \neq \emptyset$  and recall that

$$\mathcal{A}_R^k = \{K \subset \mathbb{F}^m \mid \dim(K) = k, R \subset K\}.$$

Let  $G = G(\mathcal{A}_R^k, \mathcal{A}_R^s)$  be the biregular bipartite inclusion graph and let  $d_k$  (resp.  $d_s$ ) denote the degree of vertex in  $\mathcal{A}_R^k$  (resp.  $\mathcal{A}_R^s$ ).

For every  $n, t, j \in \mathbb{N}$ , let  $h(n, t, j)$  be the number of  $t$  dimensional subspaces in  $\mathbb{F}^n$  that contain a specific  $j$  subspace,

$$h(n, t, j) = \frac{(q^n - q^j) \cdots (q^n - q^{t-1})}{(q^t - q^j) \cdots (q^t - q^{t-1})} \approx q^{(n-t)(t-j)}, \quad (6.5.3)$$

where  $\approx$  denotes equality up to a multiplicative factor  $(1 \pm o(1))$ , as before. For any fixed  $j$  dimensional subspace  $X$ , the numerator equals the number of  $t - j$  linearly independent points  $y_1, y_2, \dots, y_{t-j}$  in  $\mathbb{F}^n$  such that  $\dim(\text{span}(X, y_1, y_2, \dots, y_{t-j})) = t$ , whereas for every  $t$  dimensional subspace  $Z$ , the denominator equals the double counting of  $Z$ , i.e the number of  $t - j$  linearly independent points  $y_1, y_2, \dots, y_{t-j}$  such that  $\text{span}(X, y_1, y_2, \dots, y_{t-j}) = Z$ . We can now bound the number of vertices and the left and right degree in  $G$ .

$$\begin{aligned} |\mathcal{A}_R^k| &= h(m, k, r), & |\mathcal{A}_R^s| &= h(m, s, r), \\ d_k &= h(m, s, k), & d_s &= h(s, k, r). \end{aligned}$$

Let  $T$  be the two-step Markov operator on the bipartite graph  $G$ , starting from  $\mathcal{A}_R^k$ , we want to calculate the entries of  $T$ . Let  $K_1, K_2 \in \mathcal{A}_R^k$ , by definition  $(T)_{K_1, K_2}$  is the probability that a two-step random walk will end at  $K_2$ , conditioned on it starting from  $K_1$ .

Let  $r' = \dim(K_1 \cap K_2) \geq r$ , in this notation  $\dim(K_1 \cup K_2) = 2k - r'$ . Any 2 step random walk from  $K_1$  to  $K_2$  looks like  $K_1 \rightarrow S' \rightarrow K_2$  where  $S'$  is an  $s$  dimensional subspace containing both  $K_1$  and  $K_2$ . The number of such  $S'$  is exactly  $h(m, s, 2k - r')$ . Thus,  $(T)_{K_1, K_2}$  equals

$$\begin{aligned} (T)_{K_1, K_2} &= \Pr[\text{R.W ends at } K_2 \mid \text{R.W starts at } K_1] \\ &= \frac{h(m, s, 2k - r')}{d_k \cdot d_s} = \frac{h(m, s, 2k - r')}{h(m, s, k) \cdot h(s, k, r)}. \end{aligned} \quad (6.5.4)$$

This probability is the same for every  $K_1, K_2 \in \mathcal{A}_R^k$  such that  $\dim(K_1 \cap K_2) = r'$ , so we can denote this value by  $p_{r'} = (T)_{K_1, K_2}$ . Notice that  $p_{r'} \geq p_r$  for every  $r' \geq r$ .

Let  $G_{r'}$  be the graph with vertex set  $\mathcal{A}_R^k$ , where  $K_1, K_2$  are connected by an edge if  $\dim(K_1 \cap K_2) = r'$ . We also denote the 0/1 adjacency matrix of graph  $G_{r'}$  by  $G_{r'}$ . With these notations, the 2 step Markov operator  $T$  equals

$$T = \sum_{r'=r}^k p_{r'} G_{r'}.$$

Notice that this is not a convex combination,  $\sum_{r'} p_{r'} \neq 1$ , but rather  $p_{r'}$  are the entries of  $T$ , and  $G_{r'}$  are 0/1 matrices.

Let  $J$  be the all 1 matrix, we know that  $J = \sum_{r'=r}^k G_{r'}$ . The first matrix in the sum  $G_r$  is the only non sparse matrix, since for every subspace  $K_1 \in \mathcal{A}_R^k$ , almost all other subspaces intersects with  $K_1$  only in  $R$ . Therefore we can write  $G_r = J - \sum_{r'=r+1}^k G_{r'}$ , and get

$$T = p_r J + \sum_{r'=r+1}^k (p_{r'} - p_r) G_{r'}.$$

Since  $T$  is a Markov operator of a regular graph, the all  $\mathbf{1}$  vector is the vector with the maximal eigenvalue, which equals 1. Since  $G_{r'}$  are also regular graphs,  $\mathbf{1}$  is the vector with the maximal eigenvalue, which equals  $\deg(G_{r'})$ , which is the number of  $K' \in \mathcal{A}_R^k$  such that  $\dim(K \cap K') = r'$  (as the adjacency matrices are not normalized).

$$\begin{aligned} \deg(G_{r'}) &= h(k, r', r) \cdot \frac{(q^m - q^k) \cdots (q^m - q^{2k-r'-1})}{(q^k - q^{r'}) \cdots (q^k - q^{k-1})} \\ &\approx q^{(k-r')(r'-r)} \cdot q^{(m-k)(k-r')} = q^{(k-r')(m-k+r'-r)} \end{aligned}$$

For every  $K \in \mathcal{A}_R^k$ , the factor  $h(k, r', r)$  is the number of  $r'$  dimensional subspace in  $K$  that contain  $R$ , the second factor is the number of  $k$  dimensional subspaces that intersect with  $K$  only in a specific  $r'$  dimensional subspace.

Let  $v$  be the normalized eigenvector of the second eigenvalue of  $T$ , this means that  $v \perp \mathbf{1}$  and  $\|v\| = 1$ . Since  $J$  is the all 1 matrix,  $Jv = 0$ . We also know that for every

$r' > r$ ,  $\|G_{r'}v\| \leq \deg(G_{r'})$ , as it is true for every vector  $v$ .

$$\begin{aligned}
\|Tv\| &= \left\| \sum_{r'=r+1}^k (p_{r'} - p_r) G_{r'}v \right\| \\
&\leq \sum_{r'=r+1}^k (p_{r'} - p_r) \|G_{r'}v\| && \text{(triangle inequality)} \\
&\leq \sum_{r'=r+1}^k p_{r'} \deg(G_{r'})
\end{aligned}$$

For every  $r'$ , by using the expression for  $p_{r'}$  from (6.5.4) and bounds on  $h$  from (6.5.3) we get that

$$p_{r'} \deg(G_{r'}) \approx p_{r'} q^{(k-r')(m-s+r'-r)} \approx q^{-(r'-r)(s-2k+r')}.$$

Since  $r' > r$ ,  $(r' - r)(s - 2k + r')$  is minimized when  $r' = r + 1$  and hence

$$\lambda(T) = \|Tv\| \leq (1 + o(1)) \sum_{r'=r+1}^k \frac{1}{q^{s-2k+r'}} \leq (1 + o(1)) \cdot \frac{1}{q^{s-2k+r+1}}.$$

The lemma statement now follows from the [Fact 6.5.4](#). ■

## 6.6 Spectral Expansion Properties Proofs

**Lemma 6.6.1** (Restatement of [Lemma 6.2.3](#)). *Let  $D_1, D_2$  as defined in [Definition 6.2.2](#).*

*Let  $G = (A \cup B, E)$  be a bi-regular bipartite graph, then for every subset  $B' \subset B$  of measure  $\mu > 0$  and every  $E' \subset E$*

$$\left| \Pr_{(a,b) \sim D_1} [(a,b) \in E'] - \Pr_{(a,b) \sim D_2} [(a,b) \in E'] \right| \leq \frac{\lambda(G)}{\sqrt{\mu}}.$$

*Where is  $D_2$  returned  $\perp$ , we treat it as it is not in  $E'$ .*

*Proof.* In the proof we represent both probabilities as an inner product, and then use  $\lambda(G)$  to bound the difference. Let  $M \in \mathbb{R}^{A \times B}$  the adjacency matrix of the graph  $G$ , normalized such that  $M\mathbf{1} = \mathbf{1}$  (where the first  $\mathbf{1}$  is of dimension  $|B|$  and the second of dimension  $|A|$ ). We define the matrix  $M'$  representing the subset of edges  $E'$ ,  $M'_{a,b} = M_{a,b} \cdot (\mathbf{1}_{E'})_{a,b}$ .

Starting with the probability of  $(a, b) \sim D_1$ , the vector  $M'\mathbf{1}_{B'}$  satisfies that for every  $a \in A$ ,  $(M'\mathbf{1}_{B'})_a = \Pr_{b \in N(a)}[(a, b) \in E', b \in B']$ .

$$\begin{aligned}
\langle \mathbf{1}, M'\mathbf{1}_{B'} \rangle &= \mathbf{E}_{a \sim A} [E_{b \sim N(a)}[\mathbb{I}((a, b) \in E', b \in B')]] \\
&= \Pr_{a \sim A, b \sim N(a)} [(a, b) \in E', b \in B'] \quad (\text{using bi-regularity of } G) \\
&= \Pr_{b \sim B, a \sim N(b)} [(a, b) \in E', b \in B'] \\
&= \Pr_{b \sim B} [b \in B'] \cdot \Pr_{b \sim B, a \sim N(b)} [(a, b) \in E' \mid b \in B'] \\
&= \mu \cdot \Pr_{(a, b) \sim D_1} [(a, b) \in E'].
\end{aligned}$$

We now want to represent the second probability as an inner product. We define the vector  $P \in [0, 1]^A$  as follows, for each  $a \in A$ :

1. If  $N(a) \cap B' = \emptyset$ , then  $P_a = 0$ .
2. Else,  $P_a = \Pr_{b \in N(a)}[(a, b) \in E' \mid b \in B']$ .

In this notation  $\Pr_{(a, b) \sim D_2}[(a, b) \in E'] = \langle \mathbf{1}, P \rangle$ .

We now want to find a connection between the inner products. If  $P_a \neq 0$ , then it defined as the conditional probability, and

$$\Pr_{b \sim N(a)} [b \in B', (a, b) \in E'] = \Pr_{b \sim N(a)} [b \in B'] \Pr_{b \sim N(a)} [(a, b) \in E' \mid b \in B'] = \Pr_{b \sim N(a)} [b \in B'] P_a.$$

If  $P_a = 0$  then also  $\Pr_{b \sim N(a)} [b \in B', (a, b) \in E'] = 0$ , and the above equality still holds.

We notice that  $(M'\mathbf{1}_{B'})_a = \Pr_{b \in N(a)}[(a, b) \in E', b \in B']$  and  $(M\mathbf{1}_{B'})_a = \Pr_{b \in N(a)}[b \in B']$ , which means that for every  $a \in A$ ,  $(M'\mathbf{1}_{B'})_a = (M\mathbf{1}_{B'})_a P_a$  and

$$\langle M\mathbf{1}_{B'}, P \rangle = \langle \mathbf{1}, M'\mathbf{1}_{B'} \rangle.$$

Therefore we can express the difference between the two probabilities as

$$\begin{aligned}
\left| \Pr_{(a,b) \sim D_1} [(a,b) \in E'] - \Pr_{(a,b) \sim D_2} [(a,b) \in E'] \right| &= \left| \frac{1}{\mu} \langle \mathbf{1}, M' \mathbf{1}_{B'} \rangle - \langle \mathbf{1}, P \rangle \right| \quad (6.6.1) \\
&= \left| \frac{1}{\mu} \langle M \mathbf{1}_{B'}, P \rangle - \langle \mathbf{1}, P \rangle \right| \\
&= \frac{1}{\mu} |\langle M \mathbf{1}_{B'} - \mu \mathbf{1}, P \rangle| \\
&\leq \frac{1}{\mu} \|M \mathbf{1}_{B'} - \mu \mathbf{1}\| \|P\|
\end{aligned}$$

(By Cauchy Swartz)

Since  $P$  is a vector in  $[0, 1]$  and the inner product we use is expectation,  $\|P\| \leq 1$ . In order to finish the proof we need to bound the size of the vector

$$M \mathbf{1}_{B'} - \mu \mathbf{1} = M \mathbf{1}_{B'} - \mu M \mathbf{1} = M(\mathbf{1}_{B'} - \mu \mathbf{1}).$$

We notice that  $\mathbf{1}_{B'}$  is a  $\{0, 1\}$  vector of measure  $\mu$ , so  $\langle \mathbf{1}_{B'}, \mathbf{1} \rangle = \langle \mathbf{1}_{B'}, \mathbf{1}_{B'} \rangle = \mu$ , and  $(\mathbf{1}_{B'} - \mu \mathbf{1}) \perp \mathbf{1}_B$ . By the definition of  $\lambda(G)$ , this means that

$$\|M(\mathbf{1}_{B'} - \mu \mathbf{1})\| \leq \lambda(G) \|\mathbf{1}_{B'} - \mu \mathbf{1}\| \leq \lambda \sqrt{\mu}.$$

We substitute the norm of the vector in equation (6.6.1) and we are done.  $\blacksquare$

**Lemma 6.6.2** (Restatement of Lemma 6.2.5). *Let  $D_3, D_4$  as defined in Definition 6.2.4. Let  $G = (A \cup B, E)$  be a bi-regular bipartite graph, such that every two distinct  $b_1, b_2 \in B$  have exactly the same number of common neighbors (i.e for all distinct  $b_1, b_2 \in B$ ,  $|N(b_1) \cap N(b_2)|$  is the same), and this number is non-zero. Then for every subset  $B' \subset B$  of measure  $\mu > 0$  and every  $E' \subset E$*

$$\left| \Pr_{a, b_1, b_2 \sim D_3} [(a, b_1)(a, b_2) \in E'] - \Pr_{a, b_1, b_2 \sim D_4} [(a, b_1)(a, b_2) \in E'] \right| \leq \frac{2\lambda(G)}{\mu} + \frac{1}{\mu^2 d_A} + \frac{1}{\mu^2 |B|}$$

Where is  $D_4$  returned  $\perp$ , we treat it as it is not in  $E'$  and  $d_A$  is the degree on  $A$  side.

*Proof.* This proof is similar in spirit to the proof of Lemma 6.2.3, with more complication since the event contains two edges instead of a single one.

Let  $M \in \mathbb{R}^{A \times B}$  the adjacency matrix of the graph  $G$ , normalized such that  $M \mathbf{1} = \mathbf{1}$ . We denote by  $M'$  the matrix that represents the edges in  $E'$ , i.e for each  $a \in A, b \in B$ ,  $M'_{a,b} = M_{a,b} \cdot (\mathbf{1}_{E'})_{a,b}$ .

Starting from  $D_3$ , we first write the conditional probability

$$\begin{aligned}
& \Pr_{\substack{b_1, b_2 \\ a \sim N(b_1) \cap N(b_2)}} [b_1, b_2 \in B', (a, b_1), (a, b_2) \in E'] \\
&= \Pr_{b_1, b_2} [b_1, b_2 \in B'] \Pr_{a, b_1, b_2 \sim D_3} [(a, b_1), (a, b_2) \in E'] \quad (6.6.2) \\
&= \mu^2 \Pr_{a, b_1, b_2 \sim D_3} [(a, b_1), (a, b_2) \in E'].
\end{aligned}$$

We want to express the left side as an inner product, we notice that for each  $a \in A$ :

$$(M' \mathbf{1}_{B'})_a = \mathbf{E}_{b \sim N(a)} [\mathbb{I}(b \in B', (a, b) \in E')].$$

Therefore the inner product satisfies

$$\begin{aligned}
\langle M' \mathbf{1}_{B'}, M' \mathbf{1}_{B'} \rangle &= \mathbf{E}_{a \sim A} \left[ \mathbf{E}_{b_1, b_2 \sim N(a)} [\mathbb{I}(b_1, b_2 \in B', (a, b_1), (a, b_2) \in E')] \right] \quad (6.6.3) \\
&= \Pr_{a \sim A, b_1, b_2 \sim N(a)} [b_1, b_2 \in B', (a, b_1), (a, b_2) \in E']
\end{aligned}$$

Since each two  $b_1, b_2 \in B$  has the same number of neighbors,

$$\Pr_{\substack{a \sim A \\ b_1 \neq b_2 \sim N(a)}} [b_1, b_2 \in B', (a, b_1), (a, b_2) \in E'] = \Pr_{\substack{b_1 \neq b_2 \sim B \\ a \sim N(b_1) \cap N(b_2)}} [b_1, b_2 \in B', (a, b_1), (a, b_2) \in E'].$$

We want to switch the expression in (6.6.3) by the one in (6.6.2), we know that they are equal when  $b_1 \neq b_2$ . But the probability of  $b_1 = b_2$  is different between the two cases, it is  $\frac{1}{d_A}$  if we pick neighbors of  $a$  and  $\frac{1}{|B|}$  if we pick two random vertices in  $B$ . If we add the probability of  $b_1 = b_2$  as an error, we get that

$$\left| \mu^2 \Pr_{a, b_1, b_2 \sim D_3} [(a, b_1), (a, b_2) \in E'] - \langle M' \mathbf{1}_{B'}, M' \mathbf{1}_{B'} \rangle \right| \leq \frac{1}{d_A} + \frac{1}{|B|} \quad (6.6.4)$$

Now we want to express the probability of  $a, b_1, b_2 \sim D_4$  as an inner product. In order to do that, we define the vector  $P$ , for every  $a \in A$

1. If  $N(a) \cap B' = \emptyset$ , then  $P_a = 0$ .
2. Else,  $P_a = \Pr_{b_1, b_2 \sim N(a)} [(a, b_1), (a, b_2) \in E' \mid b_1, b_2 \in B']$ .

The vector  $P$  is defined such that

$$\Pr_{a, b_1, b_2 \sim D_4} [(a, b_1), (a, b_2) \in E'] = \mathbf{E}_a [P_a] = \langle \mathbf{1}, P \rangle.$$

We want to find a connection between this expression and the expression representing the probability  $\Pr_{a,b_1,b_2 \sim D_3}[(a,b_1)(a,b_2) \in E']$ .

We use (6.6.4) and the triangle inequality to bound the difference between the two target probabilities

$$\begin{aligned} & \left| \Pr_{a,b_1,b_2 \sim D_3}[(a,b_1)(a,b_2) \in E'] - \Pr_{a,b_1,b_2 \sim D_4}[(a,b_1)(a,b_2) \in E'] \right| \\ & \leq \left| \frac{1}{\mu^2} \langle M' \mathbf{1}_{B'}, M' \mathbf{1}_{B'} \rangle - \langle \mathbf{1}, P \rangle \right| + \frac{1}{\mu^2 d_A} + \frac{1}{\mu^2 |B|} \end{aligned} \quad (6.6.5)$$

We now need to bound the expression in (6.6.5), in order to do that, we will first show that

$$\langle M' \mathbf{1}_{B'}, M' \mathbf{1}_{B'} \rangle = \Pr_{a \sim A, b_1, b_2 \sim N(a)}[(a_1, b)(a_2, b) \in E', b_1, b_2 \in B'] = \mathbf{E}_a [P_a (M \mathbf{1}_{B'})_a^2]. \quad (6.6.6)$$

We notice that for  $a$  such that  $P_a > 0$ , it equals the conditional probability and

$$\Pr_{b_1, b_2 \sim N(a)}[(a_1, b)(a_2, b) \in E', b_1, b_2 \in B'] = \Pr_{b_1, b_2 \sim N(a)}[b_1, b_2 \in B'] P_a.$$

If  $a$  is such that  $P_a = 0$ , then  $\Pr_{b_1, b_2 \sim N(a)}[(a_1, b)(a_2, b) \in E', b_1, b_2 \in B'] = 0$  and the above equality still holds. We further notice that

$$(M \mathbf{1}_{B'})_a = \mathbf{E}_{b \sim N(a)}[\mathbb{I}(b \in B')].$$

If we substitute  $\Pr_{b_1, b_2 \sim N(a)}[b_1, b_2 \in B']$  in  $(M \mathbf{1}_{B'})_a^2$ , we get (6.6.6).

In order to finish the proof, we upper bound

$$\left| \frac{1}{\mu^2} \langle M' \mathbf{1}_{B'}, M' \mathbf{1}_{B'} \rangle - \langle \mathbf{1}, P \rangle \right| = \left| \mathbf{E}_a \left[ \frac{1}{\mu^2} P_a (M \mathbf{1}_{B'})_a^2 - P_a \right] \right| = \frac{1}{\mu^2} \left| \mathbf{E}_a [P_a ((M \mathbf{1}_{B'})_a^2 - \mu^2)] \right|.$$

We now upper bound the expectation as follows,

$$\begin{aligned} \mathbf{E}_a [P_a ((M \mathbf{1}_{B'})_a^2 - \mu^2)] &= \mathbf{E}_a [P_a ((M \mathbf{1}_{B'})_a - \mu)((M \mathbf{1}_{B'})_a + \mu)] \\ &\leq \max_a \{ |P_a| \} \mathbf{E}_a [ |((M \mathbf{1}_{B'})_a - \mu)((M \mathbf{1}_{B'})_a + \mu)| ] \\ &\leq \|M \mathbf{1}_{B'} - \mu \mathbf{1}\| \|M \mathbf{1}_{B'} + \mu \mathbf{1}\| \end{aligned} \quad (6.6.7)$$

$$\leq \lambda \sqrt{\mu} \sqrt{4\mu}, \quad (6.6.8)$$

where (6.6.7) is due to Cauchy-Schwarz inequality and using  $|P_a| \leq 1$ . In (6.6.8), we bound

$\|M\mathbf{1}_{B'} - \mu\mathbf{1}\|$  like in the previous proof,

$$\|M\mathbf{1}_{B'} - \mu\mathbf{1}\| = \|M\mathbf{1}_{B'} - \mu M\mathbf{1}\| = \|M(\mathbf{1}_{B'} - \mu\mathbf{1})\| \leq \lambda \|\mathbf{1}_{B'}\| \leq \lambda\sqrt{\mu}.$$

Finally, we bound  $\|M\mathbf{1}_{B'} + \mu\mathbf{1}\|$ :

$$\begin{aligned} \|M\mathbf{1}_{B'} + \mu\mathbf{1}\|^2 &= \langle M\mathbf{1}_{B'} + \mu\mathbf{1}, M\mathbf{1}_{B'} + \mu\mathbf{1} \rangle \\ &= \langle M\mathbf{1}_{B'}, M\mathbf{1}_{B'} \rangle + 2\langle M\mathbf{1}_{B'}, \mu\mathbf{1} \rangle + \langle \mu\mathbf{1}, \mu\mathbf{1} \rangle \\ &\leq \|\mathbf{1}_{B'}\|^2 + 2\mu + \mu^2 \|\mathbf{1}\|^2 \\ &\leq \mu + 2\mu + \mu^2 \leq 4\mu. \end{aligned}$$

■

## 6.7 Rubinfeld-Sudan Characterization

In this section, we present a proof of [Theorem 6.3.12](#). The proof uses the following fact from [\[vdWANB49\]](#):

**Fact 6.7.1.** *Let  $f : \mathbb{F}^m \rightarrow \mathbb{F}$  be a function, and let  $N_{y,h} = \{y + ih \mid i \in \{0, \dots, d+1\}\}$ .  $f$  is degree  $d$  iff it satisfies the following identity for all  $y$  and  $h$ :*

$$\sum_{i=0}^{d+1} \alpha_i f(y + ih) = 0,$$

where  $\alpha_i = \binom{d+1}{i} (-1)^{i+1}$ .

Throughout this section we let  $\alpha_i = \binom{d+1}{i} (-1)^{i+1}$  as in the above fact.

**Theorem 6.7.2** (Restatement of [Theorem 6.3.12](#)). *Let  $f : \mathbb{F}^m \rightarrow \mathbb{F}$  be a function, and let  $N_{y,h} = \{y + ih \mid i \in \{0, \dots, d+1\}\}$ , if  $f$  satisfies*

$$\Pr_{y,h \in \mathbb{F}^m} [\exists \text{ deg } d \text{ polynomial } p \text{ s.t. } p|_{N_{y,h}} = f|_{N_{y,h}}] \geq 1 - \delta, \quad (6.7.1)$$

for  $\delta \leq \frac{1}{2(d+2)^2}$ , then there exists a degree  $d$  polynomial  $g$  such that  $f \approx^{2\delta} g$ .

*Proof.* Define a function  $g : \mathbb{F}^m \rightarrow \mathbb{F}$  to be  $g(y) = \text{maj}_{h \in \mathbb{F}^m} \{\sum_{i=1}^{d+1} \alpha_i f(y + ih)\}$  breaking the ties arbitrarily. Next we argue that  $g$  is very close to  $f$  and  $g$  itself is a degree  $d$  function.

To see that  $g$  is  $(1 - 2\delta)$  close to  $f$ , consider the set of all  $y$  for which  $\Pr_h[f(y) = \sum_{i=1}^{d+1} \alpha_i f(y + ih)] > 1/2$ . For all these  $y$ ,  $f(y) = g(y)$  as  $g$  was the majority vote. It is easy to see that fraction of  $y$  for which the probability is at most  $1/2$  is at most  $2\delta$  as otherwise it will contradict the hypothesis (6.7.1). The rest of the proof will be proving the following two claims.

**Claim 6.7.3.** For all  $y \in \mathbb{F}^m$ ,  $\Pr_h[g(y) = \sum_{i=1}^{d+1} \alpha_i f(y + ih)] \geq 1 - 2(d+1)\delta$ .

**Claim 6.7.4.** For all  $y$  and  $h$  in  $\mathbb{F}^m$ , we have  $\sum_{i=0}^{d+1} \alpha_i g(y + ih) = 0$ .

Claim 6.7.4 and Fact 6.7.1 imply that  $g$  is in fact a degree  $d$  function and hence the theorem follows. We now proceed with proving these two claims.

**Proof of Claim 6.7.3:** We will show that for all  $y \in \mathbb{F}^m$ ,

$$\Pr_{h_1, h_2} \left[ \sum_{i=1}^{d+1} \alpha_i f(y + ih_1) = \sum_{j=1}^{d+1} \alpha_j f(y + jh_2) \right] \geq 1 - 2(d+1)\delta. \quad (6.7.2)$$

Note that this is enough to prove the claim. To see this, let  $p_a = \Pr_h[\sum_{i=1}^{d+1} \alpha_i f(y + ih) = a]$  for  $a \in \mathbb{F}$ . Then (6.7.2) becomes  $\sum_{a \in \mathbb{F}} p_a^2 \geq 1 - 2(d+1)\delta$ . Since  $g(y)$  was the majority vote, we have  $\Pr_h[g(y) = \sum_{i=1}^{d+1} \alpha_i f(y + ih)] = \max_{a \in \mathbb{F}} p_a \geq \sum_{a \in \mathbb{F}} p_a^2 \geq 1 - 2(d+1)\delta$ .

To prove (6.7.2), consider the following  $(d+2) \times (d+2)$  matrix  $Z$  with  $(i, j)^{th}$  entry  $Z_{i,j} = \alpha_i \alpha_j f(y + ih_1 + jh_2)$ , for  $i, j \in \{0, \dots, d+1\}$ .

$$Z = \begin{bmatrix} f(y) & \dots & \alpha_0 \alpha_j f(y + jh_2) & \dots \\ \vdots & \ddots & \vdots & \ddots \\ \alpha_i \alpha_0 f(y + ih_1) & \dots & \alpha_i \alpha_j f(y + ih_1 + jh_2) & \dots \\ \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

If  $h_1 \in \mathbb{F}^m$  u.a.r then for any  $i \in \{1, 2, \dots, d+1\}$ ,  $ih_1$  is distributed uniformly in  $\mathbb{F}^m$ . Same is true for  $h_2$  and  $jh_2$ . Consider the following events:

- For every  $i \in \{1, 2, \dots, d+1\}$ ,  $R_i$  be the event that the sum of the  $i$ 'th row is zero, i.e  $\sum_{j=0}^{d+1} Z_{i,j} = 0$ .

- For every  $j \in \{1, 2, \dots, d+1\}$ ,  $C_j$  be the event that sum of the  $j$ 'th column is *zero*, i.e  $\sum_{i=0}^{d+1} Z_{i,j} = 0$ .

Note that  $R_i, C_j$  are not defined for the first row and column ( $i = 0$  and  $j = 0$ ). Using the hypothesis (6.7.1) of the theorem and Fact 6.7.1, we have

$$\begin{aligned} \Pr_{h_1, h_2} [R_i] &\geq 1 - \delta, & \forall i \in \{1, 2, \dots, d+1\} \\ \Pr_{h_1, h_2} [C_j] &\geq 1 - \delta, & \forall j \in \{1, 2, \dots, d+1\} \end{aligned}$$

The event in (6.7.2) is same as  $\sum_{i=1}^{d+1} Z_{i,0} = \sum_{j=1}^{d+1} Z_{0,j}$  (note that the sums don't include the first element,  $Z_{0,0}$ ). If all the above events  $R_i, C_j$  happen then  $\sum_{i=1}^{d+1} Z_{i,0} = \sum_{j=1}^{d+1} Z_{0,j} = -\sum_{i,j=1}^{d+1} Z_{i,j}$ . By using union bound we get  $\Pr[\bigwedge_{i=1}^{d+1} R_i \wedge \bigwedge_{j=1}^{d+1} C_j] \geq 1 - 2(d+1)\delta$  which implies (6.7.2).

**Proof of Claim 6.7.4:** In this case, consider the following  $(d+2) \times (d+2)$  matrix  $Y$  whose  $(i, j)^{th}$  entry is  $Y_{i,j} = \alpha_i \alpha_j f(y + ih + j(h_1 + ih_2))$  except when  $j = 0$ . When  $j = 0$ ,  $Y_{i,0} = \alpha_i \alpha_0 g(y + ih)$ .

$$Y = \begin{bmatrix} \alpha_0 \alpha_0 g(y) & \dots & \alpha_0 \alpha_j f(y + jh_1) & \dots \\ \vdots & \ddots & \vdots & \ddots \\ \alpha_i \alpha_0 g(y + ih) & \dots & \alpha_i \alpha_j f(y + ih + j(h_1 + ih_2)) & \dots \\ \vdots & \ddots & \vdots & \ddots \end{bmatrix}$$

Define the following set of events:

- For  $i \in \{0, 1, \dots, d+1\}$ ,  $R_i$  be the event that the sum of all elements from row  $i$  is *zero*, i.e  $\sum_{i=0}^{d+1} Y_{i,j} = 0$ .
- For  $j \in \{0, 1, \dots, d+1\}$ ,  $C_j$  be the event that the sum of all elements from column  $j$  is *zero*, i.e  $\sum_{j=0}^{d+1} Y_{i,j} = 0$ .

Let  $h_1, h_2$  are independent and distributed u.a.r in  $\mathbb{F}^m$ . As the event  $C_0$  is independent of  $h_1$  and  $h_2$ , in order to prove the claim it is enough to show that  $\Pr_{h_1, h_2}[C_0] > 0$ .

For each row  $i \in \{0, 1, 2, \dots, d+1\}$  we apply Claim 6.7.3 with  $y' = y + ih$  and  $h' = h_1 + ih_2$ , and get  $\Pr_{h_1, h_2}[\neg R_i] \leq 2(d+1)\delta$  (note that  $\alpha_0 = -1$ ). If  $h_1, h_2$  are independent and distributed u.a.r in  $\mathbb{F}^m$  then so are  $(y + jh_1)$  and  $(h + h_2)$ . Therefore,

using the hypothesis (6.7.1) of the theorem and Fact 6.7.1, we have for all columns except  $j = 0$ ,  $\Pr_{h_1, h_2}[\neg C_j] \leq \delta$ . Using union bound, we get

$$\Pr_{h_1, h_2} \left[ \bigwedge_{i=0}^{d+1} R_i \bigwedge_{j=1}^{d+1} C_j \right] \geq 1 - 2(d+1)(d+2)\delta + (d+1)\delta > 0.$$

The claim now follows using the observation that the event  $C_0$  is implied by the event  $\bigwedge_{i=0}^{d+1} R_i \bigwedge_{j=1}^{d+1} C_j$ . To see this, the event  $\bigwedge_{i=0}^{d+1} R_i$  implies that the sum of all entries in  $Y$  is *zero* whereas  $\bigwedge_{j=1}^{d+1} C_j$  implies that the sum of all elements from the submatrix  $(Y_{i,j})_{j=1}^{d+1}$  is *zero*. Hence, if both these events happen then the sum of all elements from column 0 must be *zero*. ■

## Chapter 7

### $k$ -bit Dictatorship Test

#### 7.1 Introduction

In this chapter, we study  $k$  query dictatorship test with perfect completeness. Boolean functions are the most basic objects in the field of theoretical computer science. Studying different properties of Boolean functions has found applications in many areas including hardness of approximation, communication complexity, circuit complexity etc. In this chapter, we are interested in studying Boolean functions from a property testing point of view.

In *property testing*, one has given access to a function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  and the task is to decide if a given function has a particular property or whether it is *far* from it. One natural notion of farness is what fraction of  $f$ 's output we need to change so that the modified function has the required property. A verifier can have an access to random bits. This task of property testing seems trivial if we do not have restrictions on how many queries one can make and also on the computation. One of the main questions in this area is can we still decide if  $f$  is very far from having the property by looking at a very few locations with high probability.

There are few different parameters which are of interests while designing such tests including the amount of randomness, the number of locations queried, the amount of computation the verifier is allowed to do etc. The test can either be *adaptive* or *non-adaptive*. In an adaptive test, the verifier is allowed to query a function at a few locations and based on the answers that it gets, the verifier can decide the next locations to query whereas a non-adaptive verifier queries the function in one shot and once the answers are received makes a decision whether the function has the given property. In terms of how good the prediction is we want the test to satisfy the following two

properties:

- **Completeness:** If a given function has the property then the test should accept with high probability
- **Soundness:** If the function is far from the property then the test should accept with very tiny probability.

A test is said to have *perfect completeness* if in the completeness case the test always accepts. A test with *imperfect completeness* (or almost perfect completeness) accepts a dictator function with probability arbitrarily close to 1. Let us define the soundness parameter of the test as how small we can make the acceptance probability in the soundness case.

A function is called a *dictator* if it depends on exactly one variable i.e  $f(x_1, x_2, \dots, x_n) = x_i$  for some  $i \in [n]$ . In this work, we are interested in a non-adaptive test with perfect completeness which decides whether a given function is a dictator or far from it. This was first studied in [BGS98, PRS02] under the name of Dictatorship test and Long Code test. Apart from a natural property, dictatorship test has been used extensively in the construction of probabilistically checkable proofs (PCPs) and hardness of approximation.

An instance of a *Label Cover* is a bipartite graph  $G((A, B), E)$  where each edge  $e \in E$  is labeled by a projection constraint  $\pi_e : [L] \rightarrow [R]$ . The goal is to assign labels from  $[L]$  and  $[R]$  to vertices in  $A$  and  $B$  respectively so that the number of edge constraints satisfied is maximized. Let  $\text{GapLC}(1, \varepsilon)$  is a promise gap problem where the task is to distinguish between the case when all the edges can be satisfied and at most  $\varepsilon$  fraction of edges are satisfied by any assignment. As a consequence of the PCP Theorem [ALM<sup>+</sup>98, AS98] and the Parallel Repetition Theorem [Raz98],  $\text{GapLC}(1, \varepsilon)$  is NP-hard for any constant  $\varepsilon > 0$ . In [Hås01], Håstad used various dictatorship tests along with the hardness of Label Cover to prove optimal inapproximability results for many constraint satisfaction problems. Since then dictatorship test has been central in proving hardness of approximation.

A dictatorship test with  $k$  queries and  $P$  as an accepting predicate is usually useful in showing hardness of approximating  $\text{Max-}P$  problem. Although this is true for many CSPs, there is no black-box reduction from such dictatorship test to getting inapproximability result. One of the main obstacles in converting dictatorship test to NP-hardness result is that the constraints in Label Cover are  $d$ -to-1 where the parameter  $d$  depends on  $\varepsilon$  in  $\text{GapLC}(1, \varepsilon)$ . To remedy this, Khot in [Kho02a] conjectured that a Label Cover where the constraints are 1-to-1, called *Unique Games*, is also hard to approximate within any constant. More specifically, Khot conjectured that  $\text{GapUG}(1 - \varepsilon, \varepsilon)$ , an analogous promise problem for Unique Games, is NP-hard for any constant  $\varepsilon > 0$ . One of the significance of this conjecture is that many dictatorship tests can be composed easily with  $\text{GapUG}(1 - \varepsilon, \varepsilon)$  to get inapproximability results. However, since the Unique Games problem lacks perfect completeness it cannot be used to show hardness of approximating *satisfying* instances.

From the PCP point of view, in order to get  $k$ -bit PCP with perfect completeness, the first step is to analyze  $k$ -query dictatorship test with perfect completeness. For its application to construction PCPs there are two important things we need to study about the dictatorship test. First one is how to compose the dictatorship test with the known PCPs and second is how sound we can make the dictatorship test. In this work, we make a progress in understanding the answer to the later question. To make a remark on the first question, there is a dictatorship test with perfect completeness and soundness  $\frac{2^{\tilde{O}(k^{1/3})}}{2^k}$  and also a way to compose it with  $\text{GapLC}(1, \varepsilon)$  to get a  $k$ -bit PCP with perfect completeness and the same soundness that of the dictatorship test. This was done in [Hua13] and is currently the best known  $k$ -bit non-adaptive PCP with perfect completeness.

### Distance from a dictator function:

There are multiple notion of closeness to a dictator function. One natural definition is the minimum fraction of values we need to change such that the function becomes a dictator. There are other relaxed notions such as how close the function is to *juntas* - functions that depend on constantly many variables. Since our main motivation is the

use of dictatorship test in the construction of PCP, we can work with even more relaxed notion which we describe next: For a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  an influence of  $i^{th}$  variable is the probability that for a random input  $x \in \{0, 1\}^n$  flipping the  $i^{th}$  coordinate flips the value of the function. Note that a dictator function has a variable whose influence is 1. The influence of  $i^{th}$  variable can be expressed in terms of the fourier coefficients of  $f$  as  $\inf_i[f] = \sum_{S \subseteq [n] | i \in S} \hat{f}(S)^2$ . Using this, a degree  $d$  influence of  $f$  is  $\inf_i^{\leq d}[f] = \sum_{S \subseteq [n] | i \in S, |S| \leq d} \hat{f}(S)^2$ . We say that  $f$  is far from any dictator if for a constant  $d$  all its degree  $d$  influences are upper bounded by some small constant.

In this chapter, we investigate the trade-off between the number of queries and the soundness parameter of a dictatorship test with perfect completeness w.r.t to the above defined distance to a dictator function. A random function is far from any dictator but still it passes any (non-trivial)  $k$ -query test with probability at least  $1/2^k$ . Thus, we cannot expect the test to have soundness parameter less than  $1/2^k$ . The main theorem in this chapter is to show there exists a dictatorship test with perfect completeness and soundness at most  $\frac{2k+1}{2^k}$ .

**Theorem 7.1.1.** *Given a Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , for every  $k$  of the form  $2^m - 1$  for any  $m > 2$ , there is a  $k$  query dictatorship test with perfect completeness and soundness  $\frac{2k+1}{2^k}$ .*

Our theorem improves a result of Tamaki-Yoshida[TY15] which had a soundness of  $\frac{2k+3}{2^k}$ .

**Remark 7.1.2.** *Tamaki-Yoshida [TY15] studied a  $k$  functions test where if a given set of  $k$  functions are all the same dictator then the test accepts with probability 1. They use low degree cross influence (Definition 2.4 in [TY15]) as a criteria to decide closeness to a dictator function. Our whole analysis also goes through under the same setting as that of [TY15], but we stick to single function version for a cleaner presentation.*

### 7.1.1 Previous Work

The notion of Dictatorship Test was introduced by Bellare et al. [BGS98] in the context of Probabilistically Checkable Proofs and also studied by Parnas et al. [PRS02]. As our focus is on non-adaptive test, for an adaptive  $k$ -bit dictatorship test, we refer interested readers to [ST09, HW03, HK05, EH08]. Throughout this section, we use  $k$  to denote the number of queries and  $\varepsilon > 0$  an arbitrary small constant.

Getting the soundness parameter for a specific values of  $k$  had been studied earlier. For instance, for  $k = 3$  Håstad [Hås01] gave a 3-bit PCP with completeness  $1 - \varepsilon$  and soundness  $1/2 + \varepsilon$ . It was earlier shown by Zwick [Zwi97] that any 3-bit dictator test with perfect completeness must have soundness at least  $5/8$ . For a 3-bit dictatorship test with perfect completeness, Khot-Saket [KS06] achieved a soundness parameter  $20/27$  and they were also able to compose their test with Label Cover towards getting 3-bit PCP with similar completeness and soundness parameters. The dictatorship test of Khot-Saket [KS06] was later improved by O'Donnell-Wu [OW09a] to the optimal value of  $5/8$ . The dictatorship test of O'Donnell-Wu [OW09a] was used in O'Donnell-Wu [OW09b] to get a conditional (based on Khot's  $d$ -to-1 conjecture) 3-bit PCP with perfect completeness and soundness  $5/8$  which was later made unconditional by Håstad [Hås14].

For a general  $k$ , Samorodensky-Trevisan [ST00] constructed a  $k$ -bit PCP with imperfect completeness and soundness  $2^{2\sqrt{k}}/2^k$ . This was improved later by Engebretsen and Holmerin [EH08] to  $2^{\sqrt{2k}}/2^k$  and by Håstad-Khot [HK05] to  $2^{4\sqrt{k}}/2^k$  with perfect completeness. To break the  $2^{O(\sqrt{k})}/2^k$  Samorodensky-Trevisan [ST09] introduced the relaxed notion of soundness (based on the low degree influences) and gave a dictatorship test (called Hypergraph dictatorship test) with almost perfect completeness and soundness  $2k/2^k$  for every  $k$  and also  $(k+1)/2^k$  for infinitely many  $k$ . They combined this test with Khot's Unique Games Conjecture [Kho02a] to get a conditional  $k$ -bit PCP with similar completeness and soundness guarantees. This result was improved by Austrin-Mossel [AM09] and they achieved  $k + o(k)/2^k$  soundness.

For any  $k$ -bit CSP for which there is an instance with an integrality gap of  $c/s$  for

a certain SDP, using a result of Raghavendra [Rag08] one can get a dictatorship test with completeness  $c - \varepsilon$  and soundness  $s + \varepsilon$ . Getting the explicit values of  $c$  and  $s$  for a given value of  $k$  is not clear from this result and also it cannot be used to get a dictatorship test with perfect completeness. Similarly, using the characterization of strong approximation resistance of Khot et. al [KTW14] one can get a dictatorship test but it also lacks perfect completeness. Recently, Chan [Cha13] significantly improved the parameters for a  $k$ -bit PCP which achieves soundness  $2k/2^k$  albeit losing perfect completeness. Later Huang [Hua13] gave a  $k$ -bit PCP with perfect completeness and soundness  $2^{\tilde{O}(k^{1/3})}/2^k$ .

As noted earlier, the previously best known result for a  $k$ -bit dictatorship test with perfect completeness is by Tamaki-Yoshida [TY15]. They gave a test with soundness  $\frac{2k+3}{2^k}$  for infinitely many  $k$ .

## 7.2 Proof Overview

Let  $f : \{-1, +1\}^n \rightarrow \{-1, +1\}$  be a given balanced Boolean function<sup>1</sup>. Any non-adaptive  $k$ -query dictatorship test queries the function  $f$  at  $k$  locations and receives  $k$  bits which are the function output on these queries inputs. The verifier then applies some predicate, let's call it  $\mathcal{P} : \{0, 1\}^k \rightarrow \{0, 1\}$ , to the received bits and based on the outcome decides whether the function is a dictator or far from it. Since we are interested in a test with perfect completeness this puts some restriction on the set of  $k$  queried locations. If we denote  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k$  as the set of queried locations then the  $i^{th}$  bit from  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$  should satisfy the predicate  $\mathcal{P}$ . This is because, the test should always accept no matter which dictator  $f$  is.

Let  $\mu$  denotes a distribution on  $\mathcal{P}^{-1}(1)$ . One natural way to sample  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$  such that the test has a perfect completeness guarantee is for each coordinate  $i \in [n]$  independently sample  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)_i$  from distribution  $\mu$ . This is what we do in our dictatorship test for a specific distribution  $\mu$  supported on  $\mathcal{P}^{-1}(1)$ . It is now easy to see that the test accepts with probability 1 of  $f$  is an  $i^{th}$  dictator for any  $i \in [n]$ .

---

<sup>1</sup>Here we switch from 0/1 to +1/-1 for convenience. With this notation switch, balanced function means  $\mathbf{E}[f(\mathbf{x})] = 0$

Analyzing the soundness of a test is the main technical task. First note that the soundness parameter of the test depends on  $\mathcal{P}^{-1}(1)$  as it can be easily verified that if  $f$  is a random function, which is far from any dictator function, then the test accepts with probability at least  $\frac{|\mathcal{P}^{-1}(1)|}{2^k}$ . Thus, for a better soundness guarantee we want  $P$  to have as small support as possible. The acceptance probability of the test is given by the following expression:

$$\begin{aligned} \Pr[\text{Test accepts } f] &= \mathbf{E}[\mathcal{P}(f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_k))] \\ &= \frac{|\mathcal{P}^{-1}(1)|}{2^k} + \mathbf{E} \left[ \sum_{S \subseteq [k], S \neq \emptyset} \hat{\mathcal{P}}(S) \prod_{i \in S} f(\mathbf{x}_i) \right] \end{aligned}$$

Thus, in order to show that the test accepts with probability at most  $\frac{|\mathcal{P}^{-1}(1)|}{2^k} + \varepsilon$  it is enough to show that all the expectations  $E_S := |\mathbf{E}[\prod_{i \in S} f(\mathbf{x}_i)]|$  are small if  $f$  is far from any dictator function. Recall that at this point, we can have any predicate  $\mathcal{P}$  on  $k$  bits which the verifier uses. As we will see later, for the soundness analysis we need the predicate  $\mathcal{P}$  to satisfy certain properties.

For the rest of the section, assume that the given function  $f$  is such that the low degree influence of every variable  $i \in [n]$  is very small constant  $\tau$ . If  $f$  is a constant degree function (independent of  $n$ ) then the usual analysis goes by invoking invariance principle to claim that the quantity  $E_S$  does not change by much if we replace the distribution  $\mu$  to a distribution  $\xi$  over Gaussian random variable with the same first and second moments. An advantage of moving to a Gaussian distribution is that if  $\mu$  was a uniform and pairwise independent distribution then so is  $\xi$  and using the fact that a pairwise independence implies a total independence in the Gaussian setting, we have  $E_S \approx |\prod_{i \in S} \mathbf{E}[f(\mathbf{g}_i)]|$ . Since we assumed that  $f$  was a balanced function we have  $\mathbf{E}[f(\mathbf{g}_i)] = 0$  and hence we can say that the quantity  $E_S$  is very small.

There are two main things we need to take care in the above argument. 1) We assumed that  $f$  is a low degree function and in general it may not be true. 2) The argument crucially needed  $\mu$  to satisfy pairwise independence condition and hence it puts some restriction on the size of  $\mathcal{P}^{-1}(1)$  (Ideally, we would like  $|\mathcal{P}^{-1}(1)|$  to be as small as possible for a better soundness guarantee). We take care of (1), as in the

previous works [TY15, OW09a, AM09] etc., by requiring the distribution  $\mu$  to have *correlation* bounded away from 1. This can be achieved by making sure the support of  $\mu$  is *connected* - for every coordinate  $i \in [k]$  there exists  $a, b \in \mathcal{P}^{-1}(1)$  which differ at the  $i^{\text{th}}$  location. For such distribution, we can add independent *noise* to each co-ordinate without changing the quantity  $E_S$  by much. Adding independent noise has the effect that it damps the higher order fourier coefficients of  $f$  and the function behaves as a low degree function. We can now apply invariance principle to claim that  $E_S \approx 0$ . This was the approach in [TY15] and they could find a distribution  $\mu$  whose support size is  $2k + 3$  which is connected and pairwise independent.

In order to get an improvement in the soundness guarantee, our main technical contribution is that we can still get the overall soundness analysis to go through even if  $\mu$  does not support pairwise independence condition. To this end, we start with a distribution  $\mu$  whose support size is  $2k + 1$  and has the property that it is *almost* pairwise independent. Since we lack pairwise independence, it introduces few obstacles in the above mentioned analysis. First, the *amount* of noise we can add to each co-ordinate has some limitations. Second, because of the limited amount of independent noise, we can no longer say that the function  $f$  behaves as a low degree function after adding the noise. With the limited amount of noise, we can say that  $f$  behaves as a low degree function as long as it does not have a large fourier mass in some interval i.e the fourier mass corresponding to  $\hat{f}(T)^2$  such that  $|T| \in (s, S)$  for some constant sized interval  $(s, S)$  independent of  $n$ . We handle this obstacle by designing a family of distributions  $\mu_1, \mu_2, \dots, \mu_r$  for large enough  $r$  such that the intervals that we cannot handle for different  $\mu_i$ 's are disjoint. Also, each  $\mu_i$  has the same support and is almost pairwise independent. We then let our final test distribution as first selecting  $i \in [r]$  u.a.r and then doing the test with the corresponding distribution  $\mu_i$ . Since the total fourier mass of a  $-1/+1$  function is bounded by 1 and  $f$  was fixed before running the test it is very unlikely that  $f$  has a large fourier mass in the interval corresponding to the selected distribution  $\mu_i$ . Hence, we can conclude that for this overall distribution,  $f$  behaves as a low degree function. We note that this approach of using family of distributions was used in [Hås14] to construct a 3-bit PCP with perfect completeness.

There it was used in the composition step.

To finish the soundness analysis, let  $\tilde{f}$  be the low degree part of  $f$ . The argument in the previous paragraph concludes that  $E_S \approx |\mathbf{E}[\prod_{i \in S} \tilde{f}(\mathbf{x}_i)]|$ . As in the previous work, we can now apply invariance principle to claim that  $E_S \approx |\mathbf{E}[\prod_{i \in S} \tilde{f}(\mathbf{g}_i)]|$  where the  $i^{\text{th}}$  coordinate  $(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k)_i$  is distributed according to  $\xi$  which is almost pairwise independent. We can no longer bring the expectation inside as our distribution lacks independence. To our rescue, we have that the degree of  $\tilde{f}$  is bounded by some constant independent of  $n$ . We then prove that low degree functions are robust w.r.t slight perturbation in the inputs on average. This lets us conclude  $\mathbf{E}[\prod_{i \in S} \tilde{f}(\mathbf{g}_i)] \approx \mathbf{E}[\prod_{i \in S} \tilde{f}(\mathbf{h}_i)]$  where  $(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k)_i$  is pairwise independent. We now use the property of independence of Gaussian distribution and bring the expectation inside to conclude that  $E_S \approx |\mathbf{E}[\prod_{i \in S} \tilde{f}(\mathbf{h}_i)]| = |\prod_{i \in S} \mathbf{E}[\tilde{f}(\mathbf{h}_i)]| = 0$ .

### 7.3 Invariance Principle

Let  $\mu$  be any distribution on  $\{-1, +1\}^k$ . Consider the following distribution on  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k \in \{-1, +1\}^n$  such that independently for each  $i \in [n]$ ,  $((\mathbf{x}_1)_i, (\mathbf{x}_2)_i, \dots, (\mathbf{x}_k)_i)$  is sampled from  $\mu$ . We will denote this distribution as  $\mu^{\otimes n}$ . We are interested in evaluation of a multilinear polynomial  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  on  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$  sampled as above.

Invariance principle shows the closeness between two different distributions w.r.t some quantity of interest. We are now ready to state the version of the invariance principle from [Mos10] that we need.

**Theorem 7.3.1** ([Mos10]). *For any  $\alpha > 0, \varepsilon > 0, k \in \mathbf{N}^+$  there are  $d, \tau > 0$  such that the following holds: Let  $\mu$  be the distribution on  $\{+1, -1\}^k$  satisfying*

1.  $\mathbf{E}_{x \sim \mu}[x_i] = 0$  for every  $i \in [k]$
2.  $\mu(x) \geq \alpha$  for every  $x \in \{-1, +1\}^k$  such that  $\mu(x) \neq 0$

*Let  $\nu$  be a distribution on standard jointly distributed Gaussian variables with the same covariance matrix as distribution  $\mu$ . Then, for every set of  $k$   $(d, \tau)$ -quasirandom multilinear polynomials  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$ , and suppose  $\text{Var}[f_i^{>d}] \leq (1 - \gamma)^{2d}$  for  $0 < \gamma < 1$  it*

holds that

$$\left| \mathbf{E}_{(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k) \sim \mu^{\otimes n}} \left[ \prod_{i=1}^k f_i(\mathbf{x}_i) \right] - \mathbf{E}_{(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k) \sim \nu^{\otimes n}} \left[ \prod_{i=1}^k f_i(\mathbf{g}_i) \right] \right| \leq \varepsilon$$

(Note: one can take  $d = \frac{\log(1/\tau)}{\log(1/\alpha)}$  and  $\tau$  such that  $\varepsilon = \tau^{\Omega(\gamma/\log(1/\alpha))}$ , where  $\Omega(\cdot)$  hides constant depending only on  $k$ .)

## 7.4 Query efficient Dictatorship Test

We are now ready to describe our dictatorship test. The test queries a function at  $k$  locations and based on the  $k$  bits received decides if the function is a dictator or far from it. The check on the received  $k$  bits is based on a predicate with few accepting inputs which we describe next.

### 7.4.1 The Predicate

Let  $k = 2^m - 1$  for some  $m > 2$ . Let the coordinates of the predicate is indexed by elements of  $\mathbb{F}_2^m \setminus \mathbf{0} =: \{w_1, w_2, \dots, w_{2^m-1}\}$ . The Hadamard predicate  $H_k$  has following satisfying assignments:

$$H_k = \{x \in \{0, 1\}^k \mid \exists a \in \mathbb{F}_2^m \setminus \mathbf{0} \text{ s.t. } \forall i \in [k], x_i = a \cdot w_i\}$$

We will identify the set of satisfying assignments in  $H_k$  with the variables  $h_1, h_2, \dots, h_k$ .

Our final predicate  $\mathcal{P}_k$  is the above predicate along with few more satisfying assignments. More precisely, we add all the assignments which are at a hamming distance at most 1 from  $0^k$  i.e.  $\mathcal{P}_k = H_k \cup_{i=1}^k e_i \cup 0^k$ .

### 7.4.2 The Distribution $\mathcal{D}_{k,\varepsilon}$

For  $0 < \varepsilon \leq \frac{1}{k^2}$ , consider the following distribution  $\mathcal{D}_{k,\varepsilon}$  on the set of satisfying assignments of  $\mathcal{P}_k$  where  $\alpha := (k-1)\varepsilon$ .

Probabilities	Assignments
	$\mathcal{D}_{k,\varepsilon} \leftarrow \left\{ \begin{array}{cccc} x_1 & x_2 & \cdots & x_k \end{array} \right.$
$\frac{1}{1-\alpha} \left( \frac{1}{k+1} - \alpha \right)$	$\leftarrow \left\{ \begin{array}{cccc} 0 & 0 & \cdots & 0 \end{array} \right.$
$\frac{1}{1-\alpha} \left( \frac{1}{k+1} - \varepsilon \right)$	$\leftarrow \left\{ \begin{array}{c} h_1 \\ h_2 \\ \vdots \\ h_k \end{array} \right.$
$\frac{\varepsilon}{1-\alpha}$	$\leftarrow \left\{ \begin{array}{cccc} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ & & \ddots & \\ 0 & 0 & \cdots & 1, \end{array} \right.$

where each  $h_i$  gets a probability mass  $\frac{1}{1-\alpha}(\frac{1}{k+1} - \varepsilon)$  and each  $e_i$  gets weight  $\frac{\varepsilon}{1-\alpha}$ . The reasoning behind choosing this distribution is as follows: An uniform distribution on  $H_k \cup 0^k$  has a property that it is uniform on every single co-ordinate and also pairwise independent. These two properties are very useful proving the soundness guarantee. One more property which we require is that the distribution has to be *connected*. In order to achieve this, we add  $k$  extra assignment  $\{e_1, e_2, \dots, e_k\}$  and force the distribution to be supported on all  $H_k \cup_{i=1}^k e_i \cup 0^k$ . Even though by adding extra assignments, we loose the pairwise independent property we make sure that the final distribution is *almost* pairwise independent.

We now list down the properties of this distribution which we will use in analyzing the dictatorship test.

**Observation 7.4.1.** *The distribution  $\mathcal{D}_{k,\varepsilon}$  above has the following properties:*

1.  $\mathcal{D}_{k,\varepsilon}$  is supported on  $\mathcal{P}_k$ .

2. Marginal on every single coordinate is uniform.
3. For  $i \neq j$ , covariance of two variables  $x_i, x_j$  sampled from above distribution is:  

$$\text{Cov}[x_i, x_j] = -\frac{\varepsilon}{2(1-\alpha)}.$$
4. If we view  $\mathcal{D}_{k,\varepsilon}$  as a joint distribution on space  $\prod_{i=1}^k \mathcal{X}^{(i)}$  where each  $\mathcal{X}^{(i)} = \{0, 1\}$ , then for all  $i \in [k]$ ,  $\rho\left(\mathcal{X}^{(i)}, \prod_{j \in [k] \setminus \{i\}} \mathcal{X}^{(j)}; \mathcal{D}_{k,\varepsilon}\right) \leq 1 - \frac{\varepsilon^2}{2(1-\alpha)^2}.$

*Proof.* We prove each of the observations about the distribution. The first property is straight-forward. To prove (2), we compute  $\mathbf{E}[x_i]$  as follows.

$$\begin{aligned} \mathbf{E}[x_i] &= (k+1) \cdot \frac{1}{1-\alpha} \left( \frac{1}{k+1} - \varepsilon \right) \cdot \frac{1}{2} + \frac{\varepsilon}{1-\alpha} \\ &= \frac{1 - \varepsilon(k+1) + 2\varepsilon}{2(1-\alpha)} \\ &= \frac{1}{2} \end{aligned}$$

Consider the quantity  $\mathbf{E}_{\mathcal{D}_{k,\varepsilon}}[x_i x_j]$ . If  $x$  is sampled from 0's or  $e_i$ 's, the value is 0. Moreover, we know that if it is sampled uniformly from  $H_k \cup 0^k$ , it is 1/4 because of pairwise independence and the above fact. Therefore, we can write

$$\mathbf{E}_{\mathcal{D}_{k,\varepsilon}}[x_i x_j] = (k+1) \frac{1}{1-\alpha} \left( \frac{1}{k+1} - \varepsilon \right) \frac{1}{4}$$

We know that  $\mathbf{E}_{\mathcal{D}_{k,\varepsilon}}[x_i] = \mathbf{E}_{\mathcal{D}_{k,\varepsilon}}[x_j] = 1/2$ . Therefore,

$$\begin{aligned} \text{Cov}[x_i, x_j] &= \mathbf{E}_{\mathcal{D}_{k,\varepsilon}}[x_i x_j] - \mathbf{E}_{\mathcal{D}_{k,\varepsilon}}[x_i] \mathbf{E}_{\mathcal{D}_{k,\varepsilon}}[x_j] \\ &= \frac{1}{4(1-\alpha)} - \frac{\varepsilon(k+1)}{4(1-\alpha)} - \frac{1}{4} \\ &= \frac{-\varepsilon}{2(1-\alpha)} \end{aligned}$$

To prove the last item, we first show that the bi-partite graph  $G\left(\mathcal{X}^{(i)}, \prod_{j \in [k] \setminus \{i\}} \mathcal{X}^{(j)}, E\right)$  where  $(a, b) \in \mathcal{X}^{(i)} \times \prod_{j \in [k] \setminus \{i\}} \mathcal{X}^{(j)}$  is an edge iff  $\Pr(a, b) > 0$ , is connected. To see that the graph is connected, note that for both 0 and 1 on the left hand side,  $0^{k-1}$  is a neighbor on the right hand side as the distribution's support includes  $e_i$  for all  $i$ , and  $0^k$ . From the distribution, we see that the smallest atom is at least  $\frac{\varepsilon}{1-\alpha}$ , since  $\varepsilon \leq 1/k^2$ . We now use [Lemma 2.2.2](#) to get the required result. ■

**Test  $\mathcal{T}_{k,\delta}$** 

1. Sample  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k \in \{-1, +1\}^n$  as follows:
  - (a) For each  $i \in [n]$ , independently sample  $((\mathbf{x}_1)_i, (\mathbf{x}_2)_i, \dots, (\mathbf{x}_k)_i)$  according to the distribution  $\mathcal{D}_{k,\delta}$ .
2. Check if  $(f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_k)) \in \mathcal{P}_k$ .

**Test  $\mathcal{T}'_{k,\varepsilon}$** 

1. Set  $r = \left(\frac{k}{\text{err}}\right)^2$
2. Select  $j$  from  $\{1, 2, \dots, r\}$  uniformly at random.
3. Set  $\delta = \varepsilon_j$
4. Run test  $\mathcal{T}_{k,\delta}$ .

**7.4.3 Dictatorship Test**

We will switch the notations from  $\{0, 1\}$  to  $\{+1, -1\}$  where we identify  $+1$  as 0 and  $-1$  as 1. Let  $f : \{-1, +1\}^n \rightarrow \{-1, +1\}$  be a given boolean function. We also assume that  $f$  is folded i.e. for every  $\mathbf{x} \in \{-1, +1\}^n$ ,  $f(\mathbf{x}) = -f(-\mathbf{x})$ . We think of  $\mathcal{P}_k$  as a function  $\mathcal{P}_k : \{-1, +1\}^k \rightarrow \{0, 1\}$  such that  $P_k(z) = 1$  iff  $z \in \mathcal{P}_k$ . Consider the following dictatorship test:

The final test distribution is basically the above test where the parameter  $\delta$  is chosen from an appropriate distribution. For a given  $\frac{1}{k^2} \geq \varepsilon > 0$ , let  $\text{err} = \frac{\varepsilon/5}{2^k}$  and define the following quantities :  $\varepsilon_0 = \varepsilon$  and for  $j \geq 0$ ,  $\varepsilon_{j+1} = \text{err} \cdot 2^{-\left(\frac{k^{10}}{\text{err}^3 \varepsilon_j}\right)^k}$ .

We would like to make a remark that this particular setting of  $\varepsilon_{j+1}$  is not very important. For our analysis, we need a sequence of  $\varepsilon_j$ 's such that each subsequent  $\varepsilon_j$  is sufficiently small compared to  $\varepsilon_{j-1}$ .

## 7.5 Analysis of the Dictatorship Test

### Notation:

We can view  $f : \{-1, +1\}^n \rightarrow \{-1, +1\}$  as a function over  $n$ -fold product set  $\mathcal{X}_1 \times \mathcal{X}_2 \times \cdots \times \mathcal{X}_n$  where each  $\mathcal{X}_i = \{-1, +1\}^{\{i\}}$ . In the test distribution  $\mathcal{T}_{k,\delta}$ , we can think of  $\mathbf{x}_i$  sampled from the product distribution on  $\mathcal{X}_1^{(i)} \times \mathcal{X}_2^{(i)} \times \cdots \times \mathcal{X}_n^{(i)}$ . With these notations in hand, the overall distribution on  $(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k)$ , from the test  $\mathcal{T}_{k,\delta}$ , is a  $n$ -fold product distribution from the space

$$\prod_{j=1}^n \left( \prod_{i=1}^k \mathcal{X}_j^{(i)} \right).$$

where we think of  $\prod_{i=1}^k \mathcal{X}_j^{(i)}$  as correlated space. We define the parameters for the sake of notational convenience:

1.  $\beta_j := \frac{\varepsilon_j}{1-(k-1)\varepsilon_j}$  be the minimum probability of an atom in the distribution  $\mathcal{D}_{k,\varepsilon_j}$ .
2.  $s_{j+1} := \log(\frac{k}{\text{err}}) \frac{1}{\varepsilon_j^2}$  and  $S_j = s_{j+1}$  for  $0 \leq j \leq r$ .
3.  $\alpha_j := (k-1)\varepsilon_j$  for  $j \in [r]$ ,

### 7.5.1 Completeness

Completeness is trivial, if  $f$  is say  $i$ th dictator then the test will be checking the following condition

$$((\mathbf{x}_1)_i, (\mathbf{x}_2)_i, \dots, (\mathbf{x}_k)_i) \in \mathcal{P}_k$$

Using [Observation 7.4.1\(1\)](#), the distribution is supported on only strings in  $\mathcal{P}_k$ . Therefore, the test accepts with probability 1.

### 7.5.2 Soundness

**Lemma 7.5.1.** *For every  $\frac{1}{k^2} \geq \varepsilon > 0$  there exists  $0 < \tau < 1, d \in \mathbf{N}^+$  such that the following holds: Suppose  $f$  is such that for all  $i \in [n]$ ,  $\inf_i^{\leq d}(f) \leq \tau$ , then the test  $\mathcal{T}'_{k,\varepsilon}$  accepts with probability at most  $\frac{2k+1}{2^k} + \varepsilon$ . (Note: One can take  $\tau$  such that  $\tau^{\Omega_k(\text{err}/10s_r \log(1/\beta_r))} \leq \text{err}$  and  $d = \frac{\log(1/\tau)}{\log(1/\beta_r)}.$ )*

*Proof.* The acceptance probability of the test is given by the following expression:

$$\Pr[\text{Test accepts } f] = \mathbf{E}_{\mathcal{T}'_{k,\varepsilon}} [\mathcal{P}_k(f(\mathbf{x}_1), f(\mathbf{x}_2), \dots, f(\mathbf{x}_k))]$$

After expanding  $\mathcal{P}_k$  in terms of its Fourier expansion, we get

$$\begin{aligned} \Pr[\text{Test accepts } f] &= \frac{2k+1}{2^k} + \mathbf{E}_{\mathcal{T}'_{k,\varepsilon}} \left[ \sum_{S \subseteq [k], S \neq \emptyset} \hat{\mathcal{P}}_k(S) \prod_{i \in S} f(\mathbf{x}_i) \right] \\ &= \frac{2k+1}{2^k} + \sum_{S \subseteq [k], S \neq \emptyset} \hat{\mathcal{P}}_k(S) \mathbf{E}_{\mathcal{T}'_{k,\varepsilon}} \left[ \prod_{i \in S} f(\mathbf{x}_i) \right] \\ &\leq \frac{2k+1}{2^k} + \sum_{S \subseteq [k], S \neq \emptyset} \left| \mathbf{E}_{\mathcal{T}'_{k,\varepsilon}} \left[ \prod_{i \in S} f(\mathbf{x}_i) \right] \right| \quad (|\hat{\mathcal{P}}_k(S)| \leq 1) \\ &= \frac{2k+1}{2^k} + \sum_{S \subseteq [k], |S| \geq 2} \left| \mathbf{E}_{\mathcal{T}'_{k,\varepsilon}} \left[ \prod_{i \in S} f(\mathbf{x}_i) \right] \right|. \end{aligned}$$

In the last equality, we used the fact that each  $\mathbf{x}_i$  is distributed uniformly in  $\{-1, +1\}^n$  and hence when  $S = \{i\}$ ,  $\mathbf{E}[f(\mathbf{x}_i)] = \hat{f}(\emptyset) = 0$ . Thus, to prove the lemma it is enough to show that for all  $S \subseteq [k]$  such that  $|S| \geq 2$ ,  $\mathbf{E} \left[ \prod_{i \in S} f(\mathbf{x}_i) \right] \leq \frac{\varepsilon}{2^k}$ . This follows from [Lemma 7.5.2](#).  $\blacksquare$

**Lemma 7.5.2.** *For any  $S \subseteq [k]$  such that  $|S| \geq 2$ ,*

$$\left| \mathbf{E}_{j \in [r]} \left[ \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i \in S} f(\mathbf{x}_i) \right] \right] \right| \leq \frac{\varepsilon}{2^k}$$

The proof of this follows from the following Lemmas [7.5.3](#), [7.5.4](#), [7.5.5](#).

**Lemma 7.5.3.** *For any  $j \in [r]$  and for any  $S \subseteq [k]$ ,  $|S| \geq 2$  such that  $S = \{\ell_1, \ell_2, \dots, \ell_t\}$ ,*

$$\left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i \in S} f(\mathbf{x}_{\ell_i}) \right] - \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{x}_{\ell_i}) \right] \right| \leq 2 \cdot \text{err} + k \sqrt{\sum_{s_j \leq |T| \leq S_j} \hat{f}(T)^2}.$$

where  $\gamma_j = \frac{\text{err}}{ks_j}$  and  $d_{j,i}$  is a sequence given by  $d_{j,1} = \frac{2k^2 \cdot s_j}{\text{err}} \log\left(\frac{k}{\text{err}}\right)$  and  $d_{j,i} = (d_{j,1})^i$  for  $1 < i \leq t$ .

**Lemma 7.5.4.** *Let  $j \in [r]$  and  $\nu_j$  be a distribution on jointly distributed standard Gaussian variables with same covariance matrix as that of  $\mathcal{D}_{k,\varepsilon_j}$ . Then for any  $S \subseteq [k]$ ,  $|S| \geq 2$  such that  $S = \{\ell_1, \ell_2, \dots, \ell_t\}$ ,*

$$\left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{x}_{\ell_i}) \right] - \mathbf{E}_{(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k) \sim \nu_j^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{g}_i) \right] \right| \leq \text{err}_2$$

where  $d_{j,i}$  from [Lemma 7.5.3](#) and  $\text{err}_2 = \tau^{\Omega_k(\gamma_j/\log(1/\beta_j))}$  (Note:  $\Omega(\cdot)$  hides a constant depending on  $k$ ).

**Lemma 7.5.5.** *Let  $k \geq 2$  and  $S \subseteq [k]$  such that  $|S| \geq 2$  and let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a multilinear polynomial of degree  $D \geq 1$  such that  $\|f\|_2 \leq 1$ . If  $\mathcal{G}$  be a joint distribution on  $k$  standard gaussian random variable with a covariance matrix  $(1 + \delta)\mathbf{I} - \delta\mathbf{J}$  and  $\mathcal{H}$  be a distribution on  $k$  independent standard gaussian then it holds that*

$$\left| \mathbf{E}_{\mathcal{G}^{\otimes n}} \left[ \prod_{i \in S} f(\mathbf{g}_i) \right] - \mathbf{E}_{\mathcal{H}^{\otimes n}} \left[ \prod_{i \in S} f(\mathbf{h}_i) \right] \right| \leq \delta \cdot (2k)^{2kD}$$

Proofs of [Lemma 7.5.3](#), [7.5.4](#), [7.5.5](#) appear in [Section 7.5.3](#). We now prove [Lemma 7.5.2](#) using the above three claims.

**Proof of Lemma 7.5.2:** Let  $S = \{\ell_1, \ell_2, \dots, \ell_t\}$ . We are interested in getting an upper bound for the following expectation:

$$\left| \mathbf{E}_{j \in [r]} \left[ \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] \right] \right| \leq \mathbf{E}_{j \in [r]} \left[ \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] \right| \right].$$

Let us look at the inner expectation first. Let  $\gamma_j = \frac{\text{err}}{ks_j}$  and the sequence  $d_{j,i}$  be from [Lemma 7.5.3](#). We can upper bound the inner expectation as follows:

$$\begin{aligned} \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] \right| &\leq \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{x}_{\ell_i}) \right] \right| + 2 \cdot \text{err} + k \sqrt{\sum_{s_j \leq |T| \leq S_j} \hat{f}(T)^2} \\ &\quad \text{(by [Lemma 7.5.3](#))} \\ &\stackrel{\text{(by [Lemma 7.5.4](#))}}{\leq} \left| \mathbf{E}_{(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k) \sim \nu_j^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{g}_i) \right] \right| + \text{err}_2 + 2 \cdot \text{err} + \\ &\quad k \sqrt{\sum_{s_j \leq |T| \leq S_j} \hat{f}(T)^2}, \end{aligned} \tag{7.5.1}$$

where  $\text{err}_2 = \tau^{\Omega_k(\gamma_j/\log(1/\beta_j))}$  and  $\nu_j$  has the same covariance matrix as  $\mathcal{D}_{k,\varepsilon_j}$ . If we let  $\delta_j = \frac{2\varepsilon_j}{1-\alpha_j}$  then using [Observation 7.4.1\(3\)](#), the covariance matrix is precisely  $(1 +$

$\delta_j \mathbf{I} - \delta_j \mathbf{J}$  (note that we switched from 0/1 to  $-1/+1$  which changes the covariance by a factor of 4). Each of the functions  $(T_{1-\gamma_j} f)^{\leq d_{j,i}}$  has  $\ell_2$  norm upper bounded by 1 and degree at most  $d_{j,t}$ . We can now apply [Lemma 7.5.5](#) to conclude that

$$\left| \mathbf{E}_{(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k) \sim \nu_j^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{g}_i) \right] \right| \leq \left| \mathbf{E}_{(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k)} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{h}_i) \right] \right| + \delta_j \cdot (2k)^{2kd_{j,t}}, \quad (7.5.2)$$

where  $\mathbf{h}_i$ 's are independent and each  $\mathbf{h}_i$  is distributed according to  $\mathcal{N}(0, 1)^n$ . Thus,

$$\begin{aligned} \mathbf{E}_{(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k)} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{h}_i) \right] &= \prod_{\ell_i \in S} \mathbf{E}_{\mathbf{h}_i} \left[ (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{h}_i) \right] \\ &= \left( \widehat{(T_{1-\gamma_j} f)^{\leq d_{j,i}}(\emptyset)} \right)^t = (\hat{f}(\emptyset))^t = 0, \end{aligned} \quad (7.5.3)$$

where we used the fact that  $f$  is a folded function in the last step. Combining (7.5.1), (7.5.2) and (7.5.3), we get

$$\left| \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] \right| \leq \left( \delta_j \cdot (2k)^{2kd_{j,t}} \right) + \left( \tau^{\Omega_k(\gamma_j / \log(1/\beta_j))} \right) + 2 \cdot \text{err} + k \sqrt{\sum_{s_j \leq |T| \leq S_j} \hat{f}(T)^2} \quad (7.5.4)$$

We now upper bound the first term. For this, we use a very generous upper bounds

$$d_{j,1} \leq \frac{k^5}{\text{err}^3 \varepsilon_{j-1}^2} \text{ and } \delta_j \leq 4\varepsilon_j.$$

$$\begin{aligned} \delta_j \cdot (2k)^{2kd_{j,t}} &\leq \left( 4\varepsilon_j \cdot (2k)^{2d_{j,k}} \right) \\ &\leq \varepsilon_j \cdot 2^{\left( \frac{k^{10}}{\text{err}^3 \varepsilon_{j-1}} \right)^k} \\ &\leq \text{err}. \end{aligned} \quad \left( \text{using } \varepsilon_j = \text{err} \cdot 2^{-\left( \frac{k^{10}}{\text{err}^3 \varepsilon_{j-1}} \right)^k} \right)$$

The second term in (7.5.4) can also be upper bounded by  $\text{err}$  by choosing small enough  $\tau$ .

$$\max_j \left\{ \left( \tau^{\Omega_k(\gamma_j / \log(1/\beta_j))} \right) \right\} \leq \left( \tau^{\Omega_k(\gamma_r / \log(1/\beta_r))} \right) \leq \text{err}.$$

Finally, taking the outer expectation of (7.5.4), we get

$$\mathbf{E}_{j \in [r]} \left[ \left| \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] \right| \right] \leq 4 \cdot \text{err} + k \mathbf{E}_{j \in r} \left[ \sqrt{\sum_{s_j \leq |T| \leq S_j} \hat{f}(T)^2} \right].$$

Using Cauchy-Schwartz inequality,

$$\mathbf{E}_{j \in [r]} \left[ \sqrt{\sum_{s_j < |T| < S_j} \hat{f}(T)^2} \right] \leq \sqrt{\mathbf{E}_{j \in [r]} \left[ \sum_{s_j < |T| < S_j} \hat{f}(T)^2 \right]} \leq \frac{1}{\sqrt{r}},$$

where the last inequality uses the fact that the intervals  $(s_j, S_j)$  are disjoint for  $j \in [r]$  and  $\|f\|_2^2 = \sum_T \hat{f}(T)^2 \leq 1$ . The final bound we get is

$$\left| \mathbf{E}_{j \in [r]} \left[ \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] \right] \right| \leq \mathbf{E}_{j \in [r]} \left[ \left| \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] \right| \right] \leq 4 \cdot \text{err} + \frac{k}{\sqrt{r}} \leq 5 \cdot \text{err} \leq \frac{\varepsilon}{2^k},$$

as required.  $\blacksquare$

### 7.5.3 Proofs of Lemma 7.5.3 , 7.5.4 & 7.5.5

In this section, we provide proofs of three crucial lemmas which we used in proving the soundness analysis of our dictatorship test.

#### 7.5.4 Moving to a low degree function

The following lemma, at a very high level, says that if change  $f$  to its low degree *noisy version* then the loss we incur in the expected quantity is small.

**Lemma 7.5.6** (Restatement of Lemma 7.5.3). *For any  $j \in [r]$  and for any  $S \subseteq [k]$ ,  $|S| \geq 2$  such that  $S = \{\ell_1, \ell_2, \dots, \ell_t\}$ ,*

$$\left| \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] - \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{x}_{\ell_i}) \right] \right| \leq 2 \cdot \text{err} + k \sqrt{\sum_{s_j \leq |T| \leq S_j} \hat{f}(T)^2}.$$

where  $\gamma_j = \frac{\text{err}}{k s_j}$  and  $d_{j,i}$  is a sequence given by  $d_{j,1} = \frac{2k^2 \cdot s_j}{\text{err}} \log\left(\frac{k}{\text{err}}\right)$  and  $d_{j,i} = (d_{j,1})^i$  for  $1 < i \leq t$ .

*Proof.* The proof is presented in two parts. We first prove an upper bound on

$$\Gamma_1 := \left| \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} f(\mathbf{x}_{\ell_i}) \right] - \mathbf{E}_{\mathcal{D}_{k, \varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \right] \right| \leq \text{err} + k \sqrt{\sum_{s_j \leq |T| \leq S_j} \hat{f}(T)^2} \quad (7.5.5)$$

and then an upper bound on

$$\Gamma_2 := \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \right] - \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{x}_{\ell_i}) \right] \right| \leq \text{err}. \quad (7.5.6)$$

Note that both these upper bounds are enough to prove the lemma.

**Upper Bounding  $\Gamma_1$ :** The following analysis is very similar to the one in [TY15], we reproduce it here for the sake of completeness. The first upper bound is obtained by getting the upper bound for the following, for every  $a \in [t]$ .

$$\Gamma_{1,a} := \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i \geq a} f(\mathbf{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \right] - \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i > a} f(\mathbf{x}_{\ell_i}) \prod_{i \leq a} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \right] \right| \quad (7.5.7)$$

Note that by triangle inequality,  $\Gamma_1 \leq \sum_{a \in [t]} \Gamma_{1,a}$ .

$$\begin{aligned} (7.5.7) &= \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ (f(\mathbf{x}_{\ell_a}) - T_{1-\gamma_j} f(\mathbf{x}_{\ell_a})) \prod_{i > a} f(\mathbf{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \right] \right| \\ &= \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ (id - T_{1-\gamma_j}) f(\mathbf{x}_{\ell_a}) \prod_{i > a} f(\mathbf{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \right] \right| \\ &= \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ U((id - T_{1-\gamma_j}) f)(\mathbf{x}_{\{\ell_i: i \in [t] \setminus \{a\}\}}) \prod_{i > a} f(\mathbf{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \right] \right| \quad (7.5.8) \end{aligned}$$

where  $U$  is the Markov operator for the correlated probability space which maps functions from the space  $\mathcal{X}^{(\ell_a)}$  to the space  $\prod_{i \in [t] \setminus \{a\}} \mathcal{X}^{(\ell_i)}$ . We can look at the above expression as a product of two functions,  $F = \prod_{i > a} f \prod_{i < a} (T_{1-\gamma_j} f)$  and  $G = U(id - T_{1-\gamma_j}) f$ . From [Observation 7.4.1](#)(4), the correlation between spaces  $(\mathcal{X}^{(\ell_a)}, \prod_{i \in [t] \setminus \{a\}} \mathcal{X}^{(\ell_i)})$  is upper bounded by  $1 - \left(\frac{\varepsilon_j}{1-\alpha_j}\right)^2 \leq 1 - \varepsilon_j^2 =: \rho_j$ . Taking the Efron-Stein decomposition with respect to the product distribution, we have the following because of orthogonality of the Efron-Stein decomposition,

$$\begin{aligned} (7.5.8) &= \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} [G \times F] \right| = \left| \sum_{T \subseteq [n]} \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} [G_T \times F_T] \right| \\ &\quad (\text{by Cauchy-Schwartz}) \leq \sqrt{\sum_{T \subseteq [n]} \|F_T\|_2^2} \sqrt{\sum_{T \subseteq [n]} \|G_T\|_2^2} \quad (7.5.9) \end{aligned}$$

where the norms are with respect to  $\mathcal{D}_{k,\varepsilon_j}^{\otimes n}$ 's marginal distribution on the product distribution  $\prod_{i \in [t] \setminus \{a\}} \mathcal{X}^{(\ell_i)}$ . By orthogonality, the quantity  $\sqrt{\sum_{T \subseteq [n]} \|F_T\|_2^2}$  is just  $\|F\|_2$ . As  $F$  is product of function whose range is  $[-1, +1]$ , range of  $F$  is also  $[-1, +1]$  and hence  $\|F\|_2$  is at most 1. Therefore,

$$(7.5.9) \leq \sqrt{\sum_{T \subseteq [n]} \|G_T\|_2^2} \quad (7.5.10)$$

We have  $G_T = (UG')_T$ , where  $G' = (id - T_{1-\gamma_j})f$ . In  $G'_T$ , the Efron-Stein decomposition is with respect to the marginal distribution of  $\mathcal{D}_{k,\varepsilon_j}^{\otimes n}$  on  $\mathcal{X}^{(\ell_a)}$ , which is just uniform (by [Observation 7.4.1\(2\)](#)). Using [Proposition 2.2.5](#), we have  $G_T = UG'_T = U(id - T_{1-\gamma_j})f_T$ . Substituting in (7.5.10), we get

$$(7.5.10) = \sqrt{\sum_{T \subseteq [n]} \|U(id - T_{1-\gamma_j})f_T\|_2^2} \quad (7.5.11)$$

We also have that the correlation is upper bounded by  $\rho_j$ . We can therefore apply [Proposition 2.2.6](#), and conclude that for each  $T \subseteq [n]$ ,

$$\|U(id - T_{1-\gamma_j})f_T\|_2 \leq \rho_j^{|T|} \|(id - T_{1-\gamma_j})f_T\|_2$$

where the norm on the right is with respect to the uniform distribution. Observe that

$$\|(id - T_{1-\gamma_j})f_T\|_2^2 = (1 - (1 - \gamma_j)^{|T|})^2 \hat{f}(T)^2$$

Substituting back into (7.5.11), we get

$$(7.5.11) \leq \sqrt{\sum_{T \subseteq [n]} \underbrace{\rho_j^{2|T|} (1 - (1 - \gamma_j)^{|T|})^2 \hat{f}(T)^2}_{\text{Term}(\varepsilon_j, \gamma_j, T)}} \quad (7.5.12)$$

We will now break the above summation into three different parts and bound each part separately.

$$\begin{aligned} \Theta_1 &:= \sum_{\substack{T \subseteq [n], \\ |T| \leq s_j}} \text{Term}(\varepsilon_j, \gamma_j, T) & \Theta_2 &:= \sum_{\substack{T \subseteq [n], \\ s_j < |T| < S_j}} \text{Term}(\varepsilon_j, \gamma_j, T) \\ \Theta_3 &:= \sum_{\substack{T \subseteq [n], \\ |T| \geq S_j}} \text{Term}(\varepsilon_j, \gamma_j, T) \end{aligned}$$

• **Upper bounding  $\Theta_1$ :**

$$\begin{aligned}\Theta_1 &= \sum_{\substack{T \subseteq [n], \\ |T| \leq s_j}} \text{Term}(\varepsilon_j, \gamma_j, T) = \sum_{\substack{T \subseteq [n], \\ |T| \leq s_j}} \rho_j^{2|T|} (1 - (1 - \gamma_j)^{|T|})^2 \hat{f}(T)^2 \\ &\leq \sum_{\substack{T \subseteq [n], \\ |T| \leq s_j}} (1 - (1 - \gamma_j)^{|T|})^2 \hat{f}(T)^2.\end{aligned}$$

For every  $|T| \leq s_j$  we have  $1 - (1 - \gamma_j)^{|T|} \leq \text{err}_1/k$ . Thus,

$$\Theta_1 \leq \left(\frac{\text{err}_1}{k}\right)^2 \sum_{\substack{T \subseteq [n], \\ |T| \leq s_j}} \hat{f}(T)^2.$$

• **Upper bounding  $\Theta_3$ :**

$$\Theta_3 = \sum_{\substack{T \subseteq [n], \\ |T| \geq S_j}} \text{Term}(\varepsilon_j, \gamma_j, T) = \sum_{\substack{T \subseteq [n], \\ |T| \geq S_j}} \rho_j^{2|T|} (1 - (1 - \gamma_j)^{|T|})^2 \hat{f}(T)^2 \leq \sum_{\substack{T \subseteq [n], \\ |T| \geq S_j}} \rho_j^{2|T|} \hat{f}(T)^2.$$

For every  $|T| \geq S_j$  we have  $\rho_j^{|T|} \leq (1 - \varepsilon_j^2)^{|T|} \leq \text{err}_1/k$ . Thus,

$$\Theta_3 \leq \left(\frac{\text{err}_1}{k}\right)^2 \sum_{\substack{T \subseteq [n], \\ |T| \geq S_j}} \hat{f}(T)^2.$$

Substituting these upper bounds in (7.5.12),

$$\begin{aligned}\Gamma_{1,a} &\leq \sqrt{\left(\frac{\text{err}_1}{k}\right)^2 \sum_{\substack{T \subseteq [n], \\ |T| \leq s_j \text{ or } |T| \geq S_j}} \hat{f}(T)^2 + \sum_{\substack{T \subseteq [n], \\ s_j < |T| < S_j}} \hat{f}(T)^2} \\ &\leq \sqrt{\left(\frac{\text{err}_1}{k}\right)^2 + \sum_{s_j < |T| < S_j} \hat{f}(T)^2} \quad (\text{since } \sum_T \hat{f}(T)^2 \leq 1) \\ &\leq \frac{\text{err}_1}{k} + \sqrt{\sum_{s_j < |T| < S_j} \hat{f}(T)^2}. \quad (\text{using concavity})\end{aligned}$$

The required upper bound on  $\Gamma_1$  follows by using  $\Gamma_1 \leq \sum_{a \in [t]} \Gamma_{1,a}$  and the above bound.

**Upper Bounding  $\Gamma_2$ :** We will now show an upper bound on  $\Gamma_2$ . The approach is similar to the previous case, we upper bound the following quantity for every  $a \in [t]$

$$\Gamma_{2,a} := \left| \begin{aligned} &\mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i \geq a} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \prod_{i < a} (T_{1-\gamma_j} f^{\leq d_{j,i}})(\mathbf{x}_{\ell_i}) \right] \\ &\mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{i > a} (T_{1-\gamma_j} f)(\mathbf{x}_{\ell_i}) \prod_{i \leq a} (T_{1-\gamma_j} f^{\leq d_{j,i}})(\mathbf{x}_{\ell_i}) \right] \end{aligned} \right|$$

$$\begin{aligned}
&= \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \left( T_{1-\gamma_j} f(\mathbf{x}_{\ell_a}) - T_{1-\gamma_j} f^{\leq d_{j,a}}(\mathbf{x}_{\ell_a}) \right) \prod_{i>a} T_{1-\gamma_j} f(\mathbf{x}_{\ell_i}) \prod_{i<a} (T_{1-\gamma_j} f^{\leq d_{j,i}})(\mathbf{x}_{\ell_i}) \right] \right| \\
&= \left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \left( T_{1-\gamma_j} f^{> d_{j,a}}(\mathbf{x}_{\ell_a}) \right) \prod_{i>a} T_{1-\gamma_j} f(\mathbf{x}_{\ell_i}) \prod_{i<a} (T_{1-\gamma_j} f^{\leq d_{j,i}})(\mathbf{x}_{\ell_i}) \right] \right| \quad (7.5.13)
\end{aligned}$$

By using Holder's inequality we can upper bound (7.5.13) as:

$$(7.5.13) \leq \|T_{1-\gamma_j} f^{> d_{j,a}}\|_2 \prod_{i>a} \|T_{1-\gamma_j} f\|_{2(t-1)} \prod_{i<a} \|T_{1-\gamma_j} f^{\leq d_{j,i}}\|_{2(t-1)}, \quad (7.5.14)$$

where each norm is w.r.t the uniform distribution as marginal of each  $\mathbf{x}_{\ell_i}$  is uniform in  $\{+1, -1\}^n$ . Now,  $\|T_{1-\gamma_j} f\|_{2(t-1)} \leq 1$  as the range of  $T_{1-\gamma_j} f$  is in  $[-1, +1]$ . To upper bound  $\|T_{1-\gamma_j} f^{\leq d_{j,i}}\|_{2(t-1)}$ , we use [Proposition 2.3.3](#) and using the fact that  $\{-1, +1\}$  uniform random variable is  $(2, q, 1/\sqrt{q-1})$  hypercontractive ([Theorem 2.3.2](#)) to get

$$\|T_{1-\gamma_j} f^{\leq d_{j,i}}\|_{2(t-1)} \leq (2t-3)^{d_{j,i}} \|T_{1-\gamma_j} f^{\leq d_{j,i}}\|_2 \leq (2t)^{d_{j,i}}.$$

Plugging this in (7.5.14), we get

$$\begin{aligned}
(7.5.14) &\leq \|T_{1-\gamma_j} f^{> d_{j,a}}\|_2 \prod_{i>a} (2t)^{d_{j,i}} \leq (1-\gamma_j)^{d_{j,a}} \cdot \prod_{i<a} (2t)^{d_{j,i}} \\
&\leq e^{-\gamma_j d_{j,a}} \cdot (2k)^{k \cdot d_{j,a-1}} \\
&\leq e^{-\frac{\text{err}}{ks_j} \cdot d_{j,a}} \cdot (2k)^{k \cdot d_{j,a-1}} \quad (7.5.15)
\end{aligned}$$

Now,

$$\begin{aligned}
d_{j,1} \cdot d_{j,a-1} &= d_{j,a} \\
\frac{2k^2 \cdot s_j}{\text{err}} \log\left(\frac{k}{\text{err}}\right) \cdot d_{j,a-1} &= d_{j,a} \\
\frac{k^2 \cdot s_j}{\text{err}} \log\left(\frac{k}{\text{err}}\right) + \frac{k^2 \cdot s_j}{\text{err}} \log\left(\frac{k}{\text{err}}\right) \cdot d_{j,a-1} &\leq d_{j,a} \\
\frac{k \cdot s_j}{\text{err}} \log\left(\frac{k}{\text{err}}\right) + \frac{k^2 \cdot s_j}{\text{err}} \cdot \log(2k) \cdot d_{j,a-1} &\leq d_{j,a} \\
\frac{k \cdot s_j}{\text{err}} \cdot \left( \log\left(\frac{k}{\text{err}}\right) + k \cdot d_{j,a-1} \log(2k) \right) &= d_{j,a} \\
\frac{k \cdot s_j}{\text{err}} \cdot \log\left(\frac{k}{\text{err}} (2k)^{k \cdot d_{j,a-1}}\right) &= d_{j,a}
\end{aligned}$$

This implies

$$\begin{aligned}
\log \left( \frac{k}{\text{err}} (2k)^{k \cdot d_{j,a-1}} \right) &= \frac{\text{err}}{k s_j} \cdot d_{j,a} \\
\Rightarrow \frac{k}{\text{err}} (2k)^{k \cdot d_{j,a-1}} &= e^{\frac{\text{err}}{k s_j} \cdot d_{j,a}} \\
\Rightarrow e^{-\frac{\text{err}}{k s_j} \cdot d_{j,a}} \cdot (2k)^{k \cdot d_{j,a-1}} &= \frac{\text{err}}{k}.
\end{aligned}$$

Thus from (7.5.15), we have  $\Gamma_{2,a} \leq \frac{\text{err}}{k}$ . To conclude the proof, by triangle inequality we have  $\Gamma_2 \leq \sum_{a \in [t]} \Gamma_{2,a} \leq \text{err}$ .  $\blacksquare$

### 7.5.5 Moving to the Gaussian setting

We are now in the setting of *low degree* polynomials because of Lemma 7.5.3. The following lemma let us switch from our test distribution to a Gaussian distribution with the same first two moments.

**Lemma 7.5.7** (Restatement of Lemma 7.5.4). *Let  $j \in [r]$  and  $\nu_j$  be a distribution on jointly distributed standard Gaussian variables with same covariance matrix as that of  $\mathcal{D}_{k,\varepsilon_j}$ . Then for any  $S \subseteq [k]$ ,  $|S| \geq 2$  such that  $S = \{\ell_1, \ell_2, \dots, \ell_t\}$ ,*

$$\left| \mathbf{E}_{\mathcal{D}_{k,\varepsilon_j}^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{x}_{\ell_i}) \right] - \mathbf{E}_{(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k) \sim \nu_j^{\otimes n}} \left[ \prod_{\ell_i \in S} (T_{1-\gamma_j} f)^{\leq d_{j,i}}(\mathbf{g}_i) \right] \right| \leq \text{err}_2$$

where  $d_{j,i}$  from Lemma 7.5.3 and  $\text{err}_2 = \tau^{\Omega_k(\gamma_j / \log(1/\beta_j))}$  (Note:  $\Omega(\cdot)$  hides a constant depending on  $k$ ).

*Proof.* Using the definition of  $(d, \tau)$ -quasirandom function and Fact 2.1.7, if  $f$  is  $(d, \tau)$ -quasirandom then so is  $T_{1-\gamma} f$  for any  $0 \leq \gamma \leq 1$ . Also,  $T_{1-\gamma} f$  satisfies

$$\text{Var}[T_{1-\gamma} f^{>d}] = \sum_{\substack{T \subseteq [n] \\ |T| > d}} (1-\gamma)^{2|T|} \hat{f}(T)^2 \leq (1-\gamma)^{2d} \cdot \sum_{\substack{T \subseteq [n] \\ |T| > d}} \hat{f}(T)^2 \leq (1-\gamma)^{2d}.$$

The lemma follows from a direct application of Theorem 7.3.1.  $\blacksquare$

### 7.5.6 Making Gaussian variables independent

Our final lemma allows us to make the Gaussian variables independent. Here we crucially need the property that the polynomials we are dealing with are low degree polynomials. Before proving [Lemma 7.5.5](#), we need the following lemma which says that low degree functions are robust to small perturbations in the input on average.

**Lemma 7.5.8.** *Let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a multilinear polynomial of degree  $d$  such that  $\|f\|_2 \leq 1$  suppose  $\mathbf{x}, \mathbf{z} \sim \mathcal{N}(0, 1)^n$  be  $n$ -dimensional standard gaussian vectors such that  $\mathbf{E}[x_i z_i] \geq 1 - \delta$  for all  $i \in [n]$ . Then*

$$\mathbf{E}[(f(\mathbf{x}) - f(\mathbf{z}))^2] \leq 2\delta d.$$

*Proof.* For  $T \subseteq [n]$ , we have

$$\mathbf{E}[\chi_T(\mathbf{x})\chi_T(\mathbf{z})] = \prod_{i \in T} \mathbf{E}[x_i z_i] \geq \prod_{i \in T} (1 - \delta) \geq (1 - \delta)^{|T|}$$

We now bound the following expression,

$$\begin{aligned} \mathbf{E}[(f(\mathbf{x}) - f(\mathbf{z}))^2] &= \mathbf{E}[f(\mathbf{x})^2 + f(\mathbf{z})^2 - 2f(\mathbf{x})f(\mathbf{z})] \\ &= \sum_{T \subseteq [n], |T| \leq d} \hat{f}(T)^2 (2 - 2\mathbf{E}[\chi_T(\mathbf{x})\chi_T(\mathbf{z})]) \\ &\leq 2 \cdot \sum_{T \subseteq [n], |T| \leq d} \hat{f}(T)^2 (1 - (1 - \delta)^{|T|}) \\ &\leq 2 \cdot \sum_{T \subseteq [n], |T| \leq d} \hat{f}(T)^2 \delta |T| \\ &\leq 2\delta d \cdot \sum_{T \subseteq [n], |T| \leq d} \hat{f}(T)^2 \leq 2\delta d, \end{aligned}$$

where the last inequality uses  $\|f\|_2 \leq 1$ . ■

We are now ready to prove [Lemma 7.5.5](#).

**Lemma 7.5.9** (Restatement of [Lemma 7.5.5](#)). *Let  $k \geq 2$  and  $2 \leq t \leq k$  and let  $f : \mathbb{R}^n \rightarrow \mathbb{R}$  be a multilinear polynomial of degree  $D \geq 1$  such that  $\|f\|_2 \leq 1$ . If  $\mathcal{G}$  be a joint distribution on  $k$  standard gaussian random variable with covariance matrix  $(1 + \delta)\mathbf{I} - \delta\mathbf{J}$  and  $\mathcal{H}$  be a distribution on  $k$  independent standard gaussian then it holds*

that

$$\left| \mathbf{E}_{\mathcal{G}^{\otimes n}} \left[ \prod_{i \in [t]} f(\mathbf{g}_i) \right] - \mathbf{E}_{\mathcal{H}^{\otimes n}} \left[ \prod_{i \in [t]} f(\mathbf{h}_i) \right] \right| \leq \delta \cdot (2k)^{2Dk}.$$

*Proof.* Let  $\Sigma = (1 + \delta)\mathbf{I} - \delta\mathbf{J}$  be the covariance matrix. Let  $\mathbf{M} = (1 - \delta')((1 + \beta)\mathbf{I} - \beta\mathbf{J})$  be a matrix such that  $\mathbf{M}^2 = \Sigma$ . There are multiple  $\mathbf{M}$  which satisfy  $\mathbf{M}^2 = \Sigma$ . We chose the  $\mathbf{M}$  stated above to make the analysis simpler. From the way we chose  $\mathbf{M}$  and using the condition  $\mathbf{M}^2 = \Sigma$ , it is easy to observe that  $\beta$  and  $\delta'$  should satisfy the following two conditions:

$$1 - \delta' = \frac{1}{\sqrt{1 + (k - 1)\beta^2}} \quad \text{and} \quad \frac{(k - 2)\beta^2 - 2\beta}{1 + (k - 1)\beta^2} = -\delta.$$

Since  $\mathcal{H}$  is a distribution of  $k$  independent standard gaussians, we can generate a sample  $x \sim \mathcal{G}$  by sampling  $y \sim \mathcal{H}$  and setting  $x = \mathbf{M}y$ . In what follows, we stick to the following notation:  $(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k) \sim \mathcal{H}^{\otimes n}$  and  $(\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_k)_j = \mathbf{M}(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_k)_j$  for each  $j \in [n]$ .

Because of the way we chose to generate  $\mathbf{g}_i$ 's, we have for all  $i \in [k]$  and  $j \in [n]$ ,  $\mathbf{E}[(\mathbf{g}_i)_j(\mathbf{h}_i)_j] = 1 - \delta' \geq 1 - k\beta^2$ . To get an upper bound on  $\beta$ , notice that  $\beta$  is a root of the quadratic equation  $(k + \delta k - \delta - 2)\beta^2 - 2\beta + \delta = 0$ . Let  $k' = (k + \delta k - \delta - 2)$ , if  $\beta_1, \beta_2$  are the roots of the equation then they satisfy:  $k'\beta_1 + k'\beta_2 = 2$  and  $(k'\beta_1)(k'\beta_2) = \delta k'$  and  $\beta_1, \beta_2 > 0$ . Thus, we have  $\min\{k'\beta_1, k'\beta_2\} \leq \delta k'$  and hence, we can take  $\beta$  such that  $\beta \leq \delta$ .

We wish to upper bound the following expression:

$$\Gamma := \left| \mathbf{E}_{\mathcal{H}^{\otimes n}} \left[ \prod_{i \in [t]} f(\mathbf{g}_i) - \prod_{i \in [t]} f(\mathbf{h}_i) \right] \right|.$$

Define the following quantity

$$\Gamma_i := \left| \mathbf{E}_{\mathcal{H}^{\otimes n}} \left[ \prod_{j=1}^{i-1} f(\mathbf{h}_j) \prod_{j=i}^t f(\mathbf{g}_j) - \prod_{j=1}^i f(\mathbf{h}_j) \prod_{j=i+1}^t f(\mathbf{g}_j) \right] \right|.$$

By triangle inequality, we have  $\Gamma \leq \sum_{i \in [t]} \Gamma_i$ . We now proceed with upper bounding  $\Gamma_i$  for a given  $i \in [t]$ .

$$\begin{aligned}
\Gamma_i &= \left| \mathbf{E}_{\mathcal{H}^{\otimes n}} \left[ \prod_{j=1}^{i-1} f(\mathbf{h}_j) \prod_{j=i}^t f(\mathbf{g}_j) - \prod_{j=1}^i f(\mathbf{h}_j) \prod_{j=i+1}^t f(\mathbf{g}_j) \right] \right| \\
&= \left| \mathbf{E}_{\mathcal{H}^{\otimes n}} \left[ (f(\mathbf{g}_i) - f(\mathbf{h}_i)) \cdot \prod_{j=1}^{i-1} f(\mathbf{h}_j) \prod_{j=i+1}^t f(\mathbf{g}_j) \right] \right| \\
&\leq \sqrt{\mathbf{E}_{\mathcal{H}^{\otimes n}} [(f(\mathbf{g}_i) - f(\mathbf{h}_i))^2]} \cdot \prod_{j=1}^{i-1} \mathbf{E}_{\mathcal{H}^{\otimes n}} [f(\mathbf{h}_j)^{2(t-1)}]^{\frac{1}{2(t-1)}} \prod_{j=i+1}^t \mathbf{E}_{\mathcal{H}^{\otimes n}} [f(\mathbf{g}_j)^{2(t-1)}]^{\frac{1}{2(t-1)}},
\end{aligned}$$

where the last step uses Holder's Inequality. Now, the marginal distribution on each  $h_j$  and  $g_j$  is identical which is  $\mathcal{N}(0, 1)^n$ , we have

$$\begin{aligned}
\Gamma_i &\leq \sqrt{\mathbf{E}_{\mathcal{H}^{\otimes n}} [(f(\mathbf{g}_i) - f(\mathbf{h}_i))^2]} \cdot \prod_{j=1}^{i-1} \|f\|_{2(t-1)} \prod_{j=i+1}^t \|f\|_{2(t-1)} \\
&\leq \sqrt{\mathbf{E}_{\mathcal{H}^{\otimes n}} [(f(\mathbf{g}_i) - f(\mathbf{h}_i))^2]} \cdot (\|f\|_{2(t-1)})^{t-1}
\end{aligned}$$

Since a standard one dimensional Gaussian is  $(2, q, 1/\sqrt{q-1})$ -hypercontractive ([Theorem 2.3.2](#)), from [Proposition 2.3.3](#),  $\|f\|_{2(t-1)} \leq (\sqrt{2t-3})^D \|f\|_2 \leq (\sqrt{2t-3})^D < (2t)^{D/2}$ . Thus,

$$\Gamma_i \leq (2t)^{D(t-1)/2} \cdot \sqrt{\mathbf{E}_{\mathcal{H}^{\otimes n}} [(f(\mathbf{g}_i) - f(\mathbf{h}_i))^2]}$$

Now, each  $\mathbf{g}_i, \mathbf{h}_i$  are such that  $\mathbf{E}[(\mathbf{g}_i)_j \cdot (\mathbf{h}_i)_j] = 1 - \delta' \geq 1 - k\delta^2$  for every  $j \in [n]$ . We can apply [Lemma 7.5.8](#) to get  $\mathbf{E}_{\mathcal{H}^{\otimes n}} [(f(\mathbf{g}_i) - f(\mathbf{h}_i))^2] \leq 2k\delta^2 D$ . Hence, we can safely upper bound  $\Gamma_i$  as

$$\Gamma_i \leq (2t)^{D(t-1)/2} \cdot 2k\delta D.$$

Therefore,  $\Gamma \leq \sum_i \Gamma_i \leq t \cdot (2t)^{D(t-1)/2} \cdot 2k\delta D$  which is at most  $2k^2\delta D \cdot (2k)^{Dk/2} \leq \delta \cdot (2k)^{2Dk}$  as required.  $\blacksquare$

## Chapter 8

### Simultaneous Optimization

#### 8.1 Introduction

In this chapter, we initiate the study of *simultaneous approximation algorithms* for constraint satisfaction problems. A typical such problem is the simultaneous MAX-CUT problem: Given a collection of  $k$  graphs  $G_i = (V, E_i)$  on the same vertex set  $V$ , the problem is to find a single cut (i.e., a partition of  $V$ ) so that in *every*  $G_i$ , a large fraction of the edges go across the cut.

More generally, let  $q$  be a constant positive integer, and let  $\mathcal{F}$  be a set of bounded-arity predicates on  $[q]$ -valued variables. Let  $V$  be a set of  $n$   $[q]$ -valued variables. An  $\mathcal{F}$ -CSP is a weighted collection  $\mathcal{W}$  of constraints on  $V$ , where each constraint is an application of a predicate from  $\mathcal{F}$  to some variables from  $V$ . For an assignment  $f : V \rightarrow [q]$  and a  $\mathcal{F}$ -CSP instance  $\mathcal{W}$ , we let  $\text{val}(f, \mathcal{W})$  denote the total weight of the constraints from  $\mathcal{W}$  satisfied by  $f$ . The MAX- $\mathcal{F}$ -CSP problem is to find  $f$  which maximizes  $\text{val}(f, \mathcal{W})$ . If  $\mathcal{F}$  is the set of all predicates on  $[q]$  of arity  $w$ , then MAX- $\mathcal{F}$ -CSP is also called MAX- $w$ -CSP $_q$ .

We now describe the setting for the problem we consider: *k-fold simultaneous* MAX- $\mathcal{F}$ -CSP. Let  $\mathcal{W}_1, \dots, \mathcal{W}_k$  be  $\mathcal{F}$ -CSPs on  $V$ , each with total weight 1. Our high level goal is to find an assignment  $f : V \rightarrow [q]$  for which  $\text{val}(f, \mathcal{W}_\ell)$  is large for all  $\ell \in [k]$ .

These problems fall naturally into the domain of multi-objective optimization: there is a common search space, and multiple objective functions on that space. Since even optimizing one of these objective functions could be NP-hard, it is natural to resort to approximation algorithms. Below, we formulate some of the approximation criteria that we will consider, in decreasing order of difficulty:

1. **Pareto approximation:** Suppose  $(c_1, \dots, c_k) \in [0, 1]^k$  is such that there is an

assignment  $f^*$  with  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for each  $\ell \in [k]$ .

An  $\alpha$ -Pareto approximation algorithm in this context is an algorithm, which when given  $(c_1, \dots, c_k)$  as input, finds an assignment  $f$  such that  $\text{val}(f, \mathcal{W}_\ell) \geq \alpha \cdot c_\ell$ , for each  $\ell \in [k]$ .

2. **Minimum approximation:** This is basically the Pareto approximation problem when  $c_1 = c_2 = \dots = c_k$ . Define  $\text{OPT}$  to be the maximum, over all assignments  $f^*$ , of  $\min_{\ell \in [k]} \text{val}(f^*, \mathcal{W}_\ell)$ .

An  $\alpha$ -minimum approximation algorithm in this context is an algorithm which finds an assignment  $f$  such that  $\min_{\ell \in [k]} \text{val}(f, \mathcal{W}_\ell) \geq \alpha \cdot \text{OPT}$ .

3. **Detecting Positivity:** This is a very special case of the above, where the goal is simply to determine whether there is an assignment  $f$  which makes  $\text{val}(f, \mathcal{W}_\ell) > 0$  for all  $\ell \in [k]$ .

At the surface, this problem appears to be a significant weakening of the the simultaneous approximation goal.

When  $k = 1$ , minimum approximation and Pareto approximation correspond to the classical MAX-CSP approximation problems (which have received much attention). Our focus in this chapter is on general  $k$ . As we will see in the discussions below, the nature of the problem changes quite a bit for  $k > 1$ . In particular, direct applications of classical techniques like random assignments and convex programming relaxations fail to give even a constant factor approximation.

The theory of *exact* multiobjective optimization has been very well studied, (see eg. [PY00, Dia11] and the references therein). For several optimization problems such as shortest paths, minimum spanning trees, matchings, etc, there are polynomial time algorithms that solve the multiobjective versions exactly. For MAX-SAT, simultaneous approximation was studied by Glaßer *et al.* [GRW11].

We have two main motivations for studying simultaneous approximations for CSPs. Most importantly, these are very natural algorithmic questions, and capture naturally

arising constraints in a way which more naïve formulations (such as taking linear combinations of the given CSPs) cannot. Secondly, the study of simultaneous approximation algorithms for CSPs sheds new light on various aspects of standard approximation algorithms for CSPs. For example, our algorithms are able to favorably exploit some features of the trivial random-assignment-based  $1/2$ -approximation algorithm for MAX-CUT, that are absent in the more sophisticated SDP-based 0.878-approximation algorithm of Goemans-Williamson [GW95].

### 8.1.1 Observations about simultaneous approximation

We now discuss why a direct application of the classical CSP algorithms fails in this setting, and limitations on the approximation ratios that can be achieved.

We begin with a trivial remark. Finding an  $\alpha$ -minimum (or Pareto) approximation to the  $k$ -fold MAX- $\mathcal{F}$ -CSP is at least as hard as finding an  $\alpha$ -approximation the classical MAX- $\mathcal{F}$ -CSP problem (i.e.,  $k = 1$ ). Thus the known limits on polynomial-time approximability extend naturally to our setting.

**Max-1-SAT.** The simplest simultaneous CSP is MAX-1-SAT. The problem of getting a 1-Pareto or 1-minimum approximation to  $k$ -fold simultaneous MAX-1-SAT is essentially the **NP**-hard SUBSET-SUM problem. There is a simple  $2^{\text{poly}(k/\varepsilon)} \cdot \text{poly}(n)$ -time  $(1 - \varepsilon)$ -Pareto approximation algorithm based on dynamic programming.

It is easy to see that detecting positivity of a  $k$ -fold simultaneous MAX-1-SAT is exactly the same problem as detecting satisfiability of a SAT formula with  $k$  clauses (a problem studied in the fixed parameter tractability community. Thus, this problem can be solved in time  $2^{O(k)} \cdot \text{poly}(n)$  (see [Mar13]), and under the Exponential Time Hypothesis, one does not expect a polynomial time algorithm when  $k = \omega(\log n)$ .

**Random Assignments.** Let us consider algorithms based on random assignments. A typical example is MAX-CUT. A uniformly random cut in a weighted graph cuts  $1/2$  the total weight in expectation. This gives a  $1/2$ -approximation to the classical MAX-CUT problem.

If the cut value is concentrated around  $1/2$ , with high probability, we would obtain

a cut that's simultaneously good for all instances. For an *unweighted* graph<sup>1</sup>  $G$  with  $\omega(1)$  edges, a simple variance calculation shows that a uniformly random cut in the graph cuts a  $(\frac{1}{2} - o(1))$  fraction of the edges *with high probability*. Thus by a union bound, for  $k = O(1)$  simultaneous unweighted instances  $G_1, \dots, G_k$  of MAX-CUT, a uniformly random cut gives a  $(\frac{1}{2} - o(1))$ -minimum (and Pareto) approximation with high probability. However, for *weighted* graphs, the concentration no longer holds, and the algorithm fails to give any constant factor approximation.

For general CSPs, even for unweighted instances, the total weight satisfied by a random assignment does not necessarily concentrate. In particular, there is no “trivial” random-assignment-based constant factor approximation algorithm for simultaneous general CSPs.

**SDP Algorithms.** How do algorithms based on semi-definite programming (SDP) generalize to the simultaneous setting?

For the usual MAX-CUT problem ( $k = 1$ ), the celebrated Goemans-Williamson SDP algorithm [GW95] gives a 0.878-approximation. The SDP relaxation generalizes naturally to the simultaneous setting; it allows us to find a vector solution which is a simultaneously good cut for  $G_1, \dots, G_k$ . Perhaps we apply hyperplane rounding to the SDP solution to obtain a simultaneously good cut for all  $G_i$ ? We know that each  $G_i$  gets a good cut in expectation, but we need each  $G_i$  to get a good cut *with high probability* to guarantee a simultaneously good cut.

However, there are cases where the hyperplane rounding fails completely. For weighted instances, the SDP does not have *any* constant integrality gap. For unweighted instances, for every fixed  $k$ , we find an instance of  $k$ -fold simultaneous MAX-CUT (with arbitrarily many vertices and edges) where the SDP relaxation has value  $1 - \Omega(\frac{1}{k^2})$ , while the optimal simultaneous cut has value only  $1/2$ . Furthermore, applying the hyperplane rounding algorithm to this vector solution gives (with probability 1) a simultaneous cut value of 0. These integrality gaps are described in Section 8.8.

---

<sup>1</sup>We use the term “unweighted” to refer to instances where all the constraints have the same weight. When we talk about simultaneous approximation for unweighted instances  $\mathcal{W}_1, \dots, \mathcal{W}_k$  of MAX- $\mathcal{F}$ -CSP, we mean that in each instance  $\mathcal{W}_i$ , all constraints with nonzero weight have the equal weights (but that equal weight can be different for different  $i$ ).

Thus the natural extension of SDP based techniques for simultaneous approximation fail quite spectacularly. A-priori, this failure is quite surprising, since SDPs (and LPs) generalize to the multiobjective setting seamlessly.

**Matching Random Assignments?** Given the ease and simplicity of algorithms based on random assignments for  $k = 1$ , giving algorithms in the simultaneous setting that match their approximation guarantees is a natural benchmark. Perhaps it is always possible to do as well in the simultaneous setting as a random assignment for one instance?

Somewhat surprisingly, this is incorrect. For simultaneous MAX-Ew-SAT (CNF-SAT where every clause has exactly  $w$  distinct literals), a simple reduction from MAX-E3-SAT (with  $k = 1$ ) shows that it is **NP**-hard to give a  $(7/8 + \varepsilon)$ -minimum approximation for  $k$ -fold simultaneous MAX-Ew-SAT for large enough constants  $k$ .

**Proposition 8.1.1.** *For all integers  $w \geq 4$  and  $\varepsilon > 0$ , given  $k \geq 2^{w-3}$  instances of MAX-Ew-SAT that are simultaneously satisfiable, it is **NP**-hard to find a  $(7/8 + \varepsilon)$ -minimum (or Pareto) approximation.*

On the other hand, a random assignment to a single MAX-Ew-SAT instance satisfies a  $1 - 2^{-w}$  fraction of constraints in expectation.

This shows that simultaneous CSPs can have worse approximation factors than that expected from a random assignment. In particular, it shows that simultaneous CSPs can have worse approximation factors than their classical ( $k = 1$ ) counterparts.

### 8.1.2 Results

Our results address the approximability of  $k$ -fold simultaneous MAX- $\mathcal{F}$ -CSP for large  $k$ . Our main algorithmic result shows that for every  $\mathcal{F}$ , and  $k$  not too large,  $k$ -fold simultaneous MAX- $\mathcal{F}$ -CSP has a constant factor Pareto approximation algorithm.

**Theorem 8.1.2.** *Let  $q, w$  be constants. Then for every  $\varepsilon > 0$ , there is a  $2^{O(k^4/\varepsilon^2 \log(k/\varepsilon))}$  poly( $n$ )-time  $\left(\frac{1}{q^{w-1}} - \varepsilon\right)$ -Pareto approximation algorithm for  $k$ -fold simultaneous MAX- $w$ -CSP $_q$ .*

The dependence on  $k$  implies that the algorithm runs in polynomial time up to  $k = \tilde{O}((\log n)^{1/4})$  simultaneous instances<sup>2</sup>. The proof of the above Theorem appears in Section 8.4, and involves a number of ideas. In order to make the ideas clearer, we first describe the main ideas for approximating simultaneous MAX-2-AND (which easily implies the  $q = w = 2$  special case of the above theorem); this appears in Section 8.3.

For particular CSPs, our methods allow us to do significantly better, as demonstrated by our following result for MAX- $w$ -SAT.

**Theorem 8.1.3.** *Let  $w$  be a constant. For every  $\varepsilon > 0$ , there is a  $2^{O(k^3/\varepsilon^2 \log(k/\varepsilon))}$  poly( $n$ )-time  $(3/4 - \varepsilon)$ -Pareto approximation algorithm for  $k$ -fold MAX- $w$ -SAT.*

Given a single MAX- $Ew$ -SAT instance, a random assignment satisfies a  $1 - 2^{-w}$  fraction of the constraints in expectation. The approximation ratio achieved by the above theorem seems unimpressive in comparison (even though it is for general MAX- $w$ -SAT). However, Proposition 8.1.1 demonstrates it is **NP**-hard to do much better.

## Remarks

1. As demonstrated by Proposition 8.1.1, it is sometimes impossible to match the approximation ratio achieved by a random assignment for  $k = 1$ . By comparison, the approximation ratio given by Theorem 8.1.2 is slightly better than that achieved by a random assignment ( $1/q^w$ ). This is comparable to the best possible approximation ratio for  $k = 1$ , which is  $w/q^{w-1}$  up to constants [MM12, Cha13]. Our methods also prove that picking the best assignment out of  $2^{O(k^4/\varepsilon^2 \log(k/\varepsilon))}$  independent and uniformly random assignments achieves a  $(1/q^w - \varepsilon)$ -Pareto approximation with high probability.
2. Our method is quite general. For any CSP with a convex relaxation and an associated rounding algorithm that assigns each variable independently from a distribution with certain smoothness properties (see Section 8.3.2), it can be combined with our techniques to achieve essentially the same approximation ratio for  $k$  simultaneous instances.

---

<sup>2</sup>The  $\tilde{O}(\cdot)$  hides  $\text{poly}(\log \log n)$  factors.

3. We reiterate that Pareto approximation algorithms achieve a multiplicative approximation for each instance. One could also consider the problem of achieving simultaneous approximations with an  $\alpha$ -multiplicative and  $\varepsilon$ -additive error. This problem can be solved by a significantly simpler algorithm and analysis (but note that this variation does not even imply an algorithm for detecting positivity).

### 8.1.3 Complementary results

#### Refined hardness results

As we saw earlier, assuming ETH, there is no algorithm for even detecting positivity of  $k$ -fold simultaneous MAX-1-SAT for  $k = \omega(\log n)$ . There are trivial examples of CSPs for which detecting positivity (and in fact 1-Pareto approximation) can be solved efficiently: eg. simultaneous CSPs based on monotone predicates (where no negations of variables are allowed) are maximally satisfied by the all-1s assignment. Here we prove that for any “nontrivial” collection of Boolean predicates  $\mathcal{F}$ , assuming ETH, there is no polynomial time algorithm for detecting positivity for  $k$ -fold simultaneous MAX- $\mathcal{F}$ -CSP instances for  $k = \omega(\log n)$ . In particular, it is hard to obtain any poly-time constant factor approximation for  $k = \omega(\log n)$ . This implies a complete *dichotomy theorem* for constant factor approximations of  $k$ -fold simultaneous Boolean CSPs.

A predicate  $P : \{0, 1\}^w \rightarrow \{\text{TRUE}, \text{FALSE}\}$  is said to be *0-valid/1-valid* if the all-0-assignment/all-1-assignment satisfies  $P$ . We call a collection  $\mathcal{F}$  of predicates *0-valid/1-valid* if all predicates in  $\mathcal{F}$  are 0-valid/1-valid. Clearly, if  $\mathcal{F}$  is 0-valid or 1-valid, the simultaneous MAX- $\mathcal{F}$ -CSP instances can be solved exactly (by considering the all-0-assignment/all-1-assignment). Our next theorem shows that detecting positivity of  $\omega(\log n)$ -fold simultaneous MAX- $\mathcal{F}$ -CSP, for all other  $\mathcal{F}$ , is hard.

**Theorem 8.1.4.** *Assume the Exponential Time Hypothesis [IP01, IPZ01]. Let  $\mathcal{F}$  be a fixed finite set of Boolean predicates. If  $\mathcal{F}$  is not 0-valid or 1-valid, then for  $k = \omega(\log n)$ , detecting positivity of  $k$ -fold simultaneous MAX- $\mathcal{F}$ -CSP on  $n$  variables requires time super-polynomial in  $n$ .*

Crucially, this hardness result holds even if we require that every predicate in an

instance has all its inputs being *distinct variables*.

Our proof uses techniques underlying the dichotomy theorems of Schaefer [Sch78] for exact CSPs, and of Khanna *et al.* [KSTW01] for MAX-CSPs (although our easiness criterion is different from the easiness criteria in both these papers).

### Simultaneous approximations via SDPs

It is a tantalizing possibility that one could use SDPs to improve the LP-based approximation algorithms that we develop. Especially for constant  $k$ , it is not unreasonable to expect that one could obtain a constant factor Pareto or minimum approximation, for  $k$ -fold simultaneous CSPs, better than what can be achieved by linear programming methods.

In this direction, we show how to use simultaneous SDP relaxations to obtain a polynomial time  $(1/2 + \Omega(1/k^2))$ -minimum approximation for  $k$ -fold simultaneous MAX-CUT on *unweighted graphs*.

**Theorem 8.1.5.** *For large enough  $n$ , there is an algorithm that, given  $k$ -fold simultaneous unweighted MAX-CUT instances on  $n$  vertices, runs in time  $2^{2^{O(k)}} \cdot \text{poly}(n)$ , and computes a  $(\frac{1}{2} + \Omega(\frac{1}{k^2}))$ -minimum approximation.*

**Remark 8.1.6.** *We improve the above result in Chapter 9, where we achieve close to 0.878 approximation for simultaneous MAX-CUT*

#### 8.1.4 Our techniques

For the initial part of this discussion, we focus on the  $q = w = 2$  case, and only achieve a  $1/4 - \varepsilon$  Pareto approximation.

#### Preliminary Observations

First let us analyze the behavior of the uniformly random assignment algorithm. It is easy to compute, for each instance  $\ell \in [k]$ , the expected weight of satisfied constraints in instance  $\ell$ , which will be at least  $\frac{1}{4}$  of the total weight all constraints in instance  $\ell$ . If we knew for some reason that in each instance the weight of satisfied constraints

was concentrated around this expected value *with high probability*, then we could take a union bound over all the instances and conclude that a random assignment satisfies many constraints in each instance with high probability. It turns out that for any instance where the desired concentration does not occur, there is some variable in that instance which has high degree (i.e., the weight of all constraints involving that variable is a constant fraction of the total weight of all constraints). Knowing that there is such a high degree variable seems very useful for our goal of finding a good assignment, since we can potentially influence the satisfaction of the instance quite a bit by just by changing this one variable.

This motivates a high-level plan: either proceed by using the absence of influential variables to argue that a random assignment will succeed, or proceed by trying to set the influential variables.

### An attempt

The above high-level plan motivates the following high-level algorithm. First we identify a set  $S \subseteq V$  of “influential” variables. This set of influential variables should be of small ( $O(\log n)$ ) size, so that we can try out all assignments to these variables. Next, we take a random assignment to the remaining variables,  $g : V \setminus S \rightarrow \{0, 1\}$ . Finally, for each possible assignment  $h : S \rightarrow \{0, 1\}$ , we consider the assignment  $h \cup g : V \rightarrow \{0, 1\}$  as a candidate solution for our simultaneous CSP. We output the assignment, if any, that has  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq \alpha \cdot c_\ell$  for each  $\ell \in [k]$ . This concludes the description of the high-level algorithm.

For the analysis, we would start with the ideal assignment  $f^* : V \rightarrow \{0, 1\}$  achieving  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for each  $\ell \in [k]$ . Consider the step of the algorithm where  $h$  is taken to equal  $h^* \stackrel{\text{def}}{=} f^*|_S$ . We would like to say that for each  $\ell \in [k]$  we have:

$$\text{val}(h^* \cup g, \mathcal{W}_\ell) \geq \left(\frac{1}{4} - \varepsilon\right) \cdot \text{val}(f^*, \mathcal{W}_\ell),$$

with high probability, when  $g : V \setminus S \rightarrow \{0, 1\}$  is chosen uniformly at random. (We could then conclude the analysis by a union bound.)

A simple calculation shows that  $\mathbf{E}[\text{val}(h^* \cup g, \mathcal{W}_\ell)] \geq \frac{1}{4} \cdot \text{val}(f^*, \mathcal{W}_\ell)$ , so each instance

is well satisfied in expectation. Our hope is thus that  $\text{val}(h^* \cup g, \mathcal{W}_\ell)$  is concentrated around its mean with high probability.

There are two basic issues with this approach<sup>3</sup>:

1. The first issue is how to define the set  $S$  of influential variables. For some special CSPs (such as MAX-CUT and MAX-SAT), there is a natural choice which works (to choose a set of variables with high degree, which is automatically small). But for general CSPs, it could be the case that variables with exponentially small degree are important contributors to the ideal assignment  $f^*$ .
2. Even if one chooses the set  $S$  of influential variables appropriately, the analysis cannot hope to argue that  $\text{val}(h^* \cup g, \mathcal{W}_\ell)$  concentrates around its expectation with high probability. Indeed, it can be the case that for a random assignment  $g$ ,  $\text{val}(h^* \cup g, \mathcal{W}_\ell)$  is not concentrated at all.

### A working algorithm:

Our actual algorithm and analysis solve these problems by proceeding in a slightly different way. The first key idea is to find the set of influential variables by iteratively including variables into this set, and simultaneously assigning these variables. This leads to a tree-like evolution of the set of influential variables. The second key idea is in the analysis: instead of arguing about the performance of the algorithm when considering the partial assignment  $h^* = f^*|_S$ , we will perform a delicate *perturbation* of  $h^*$  to obtain an  $h' : S \rightarrow \{0, 1\}$ , and show that  $\text{val}(h' \cup g, \mathcal{W}_\ell)$  is as large as desired. Intuitively, this perturbation only slightly worsens the satisfied weight of  $h^*$ , while reducing the reliance of the good assignment  $f^*$  on any specialized properties of  $f^*|_S$ .

To implement this, the algorithm will maintain a tree of possible evolutions of a set  $S \subseteq V$  and a partial assignment  $\rho : S \rightarrow \{0, 1\}$ . In addition, every variable  $x \in S$  will be labelled by an instance  $\ell \in [k]$ . The first stage of the algorithm will grow this tree in

---

<sup>3</sup> These problems do not arise if we only aim for the weaker “additive-multiplicative” Pareto approximation guarantee (where one allows for both some additive loss and multiplicative loss in the approximation), and in fact the above mentioned high-level plan does work. The pure multiplicative approximation guarantee seems to be significantly more delicate.

several steps. In the beginning, at the root of the tree, we have  $S = \emptyset$ . At every stage, we will either terminate that branch of the tree, or else increase the size of the set  $S$  by 1 (or 2), and consider all 2 (or 4) extensions of  $\rho$  to the newly grown  $S$ .

To grow the tree, the algorithm considers a random assignment  $g : V \setminus S \rightarrow \{0, 1\}$ , and computes, for each instance  $i \in [k]$ , the expected satisfied weight  $\mathbf{E}_g[\text{val}(\rho \cup g, \mathcal{W}_\ell)]$  and the variance of the satisfied weight  $\text{Var}_g[\text{val}(\rho \cup g, \mathcal{W}_\ell)]$ . We can thus classify instances as concentrated or non-concentrated. If more than  $t$  variables in  $S$  are labelled by instance  $\ell$  (where  $t = O_{k,\varepsilon}(1)$  is some parameter to be chosen), we call instance  $\ell$  saturated. If every unsaturated instance is concentrated, then we are done with this  $S$  and  $\rho$ , and this branch of the tree gets terminated.

Otherwise, we know that there some unsaturated instance  $\ell$  which is not concentrated. We know that this instance  $\ell$  must have some variable  $x \in V \setminus S$  which has high *active degree* (this is the degree after taking into account the partial assignment  $\rho$ ). The algorithm now takes two cases:

- **Case 1:** If this high-active-degree variable  $x$  is involved in a high-weight constraint on  $\{x, y\}$  for some  $y \in V \setminus S$ , then we include both  $x, y$  into the set  $S$ , and consider all 4 possible extensions of  $\rho$  to this new  $S$ .  $x, y$  are both labelled with instance  $\ell$ .
- **Case 2:** Otherwise, every constraint involving  $x$  is low-weight (and in particular there must be many of them), and in this case we include  $x$  into the set  $S$ , and consider both possible extensions of  $\rho$  to this new  $S$ .  $x$  is labelled with instance  $\ell$ .

This concludes the first stage of the algorithm, which created a tree whose leaves contain various  $(S, \rho)$  pairs.

For the second stage of the algorithm we visit each leaf  $(S, \rho)$ . We choose a uniformly random  $g : V \setminus S \rightarrow \{0, 1\}$ , and consider for every  $h : S \rightarrow \{0, 1\}$ , the assignment  $h \cup g : V \rightarrow \{0, 1\}$ . Note that we go over all assignments to the set  $S$ , independent of the partial assignment to  $S$  associated with the leaf.

### The analysis:

At the end of the evolution, at every leaf of the tree every instance is either highly-concentrated or saturated. If instance  $\ell$  is highly-concentrated, we will have the property that the random assignment to  $V \setminus S$  has the right approximation factor for instance  $\ell$ . If the instance  $\ell$  is saturated, then we know that there are many variables in  $S$  labelled by instance  $\ell$ ; and at the time these variables were brought into  $S$ , they had high active degree.

The main part of the analysis is then a delicate *perturbation* procedure, which starts with the partial assignment  $h^* \stackrel{\text{def}}{=} f^*|_S$ , and perturbs it to some  $h' : S \rightarrow \{0, 1\}$  with a certain robustness property. Specifically, it ensures that for every saturated instance  $\ell \in [k]$ , we have  $\text{val}(h' \cup g, \mathcal{W}_\ell)$  is at least as large as the total weight in instance  $\ell$  of all constraints not wholly contained within  $S$ . At the same time, the perturbation ensures that for unsaturated instances  $\ell \in [k]$ ,  $\text{val}(h' \cup g, \mathcal{W}_\ell)$  is almost as large as  $\text{val}(h^* \cup g, \mathcal{W}_\ell)$ . This yields the desired Pareto approximation. The perturbation procedure modifies the assignment  $h^*$  at a few carefully chosen variables (at most two variables per saturated instance). After picking the variables for an instance, if the variables were brought into  $S$  by Case 1, we can satisfy the heavy constraint involving them. Otherwise, we use a Lipschitz concentration bound to argue that a large fraction of the constraints involving the variable and  $V \setminus S$  can be satisfied; this is the second place where we use the randomness in the choice of  $g$ .

As we mentioned earlier, this perturbation is necessary! It is not true the assignment  $h^* \cup g$  will give a good Pareto approximation with good probability <sup>4</sup>.

### Improved approximation, and generalization:

To get the claimed  $(\frac{1}{2} - \varepsilon)$ -Pareto approximation for the  $q = w = 2$  case, we replace the uniformly random choice of  $g : V \setminus S \rightarrow \{0, 1\}$  by a suitable LP relaxation + randomized rounding strategy. Concretely, at every leaf  $(S, \rho)$ , we do the following. First we write an LP relaxation of the residual MAX-2-CSP problem. Then, using a

---

<sup>4</sup>See Section 8.8.2 for an example

rounding algorithm of Trevisan (which has some desirable smoothness properties), we choose  $g : V \setminus S \rightarrow \{0, 1\}$  by independently rounding each variable. Finally, for all  $h : S \rightarrow \{0, 1\}$ , we consider the assignment  $h \cup g$ . The analysis is nearly identical (but crucially uses the smoothness of the rounding), and the improved approximation comes from the improved approximation factor of the classical LP relaxation for MAX-2-CSP.

The generalization of this algorithm to general  $q, w$  is technical but straightforward. One notable change is that instead of taking 2 cases each time we grow the tree, we end up taking  $w$  cases. In case  $j$ , we have a set of  $j$  variables such that the total weight of constraints involving all the  $j$  variables is large, however for every remaining variable  $z$ , the weight of constraints involving all the  $j$  variables together with  $z$  is small. The analysis of the perturbation is similar.

The algorithm for MAX- $w$ -SAT uses the fact that the LP rounding gives a  $3/4$  approximation for MAX- $w$ -SAT. Moreover, since a MAX- $w$ -SAT constraint can be satisfied by perturbing any one variable, the algorithm does not require a tree of evolutions. It only maintains a set of “influential” variables, and hence, is simpler.

### 8.1.5 Related Work

The theory of *exact* multiobjective optimization has been very well studied, (see eg. [PY00, Dia11] and the references therein).

The only directly comparable work for simultaneous approximation algorithms for CSPs we are aware of is the work of Glaßer *et al.* [GRW11]<sup>5</sup>. They give a  $1/2$ -Pareto approximation for MAX-SAT with a running time of  $n^{O(k^2)}$ . For bounded width clauses, our algorithm does better in both approximation guarantee and running time.

For MAX-CUT, there are a few results of a similar flavor. For two graphs, the results of Angel *et al.* [ABG06] imply a 0.439-Pareto approximation algorithm (though their actual results are incomparable to ours). Bollobás and Scott [BS04] asked what is the largest simultaneous cut in two unweighted graphs with  $m$  edges each. Kuhn and Osthus [KO07], using the second moment method, proved that for  $k$  simultaneous

---

<sup>5</sup>They also give Pareto approximation results for simultaneous TSP (also see references therein).

unweighted instances, there is a simultaneous cut that cuts at least  $m/2 - O(\sqrt{km})$  edges in each instance, and give a deterministic algorithm to find it (this leads to a  $(\frac{1}{2} - o(1))$ -Pareto approximation for unweighted instances with sufficiently many edges). Our main theorem implies the same Pareto approximation factor for simultaneous MAX-CUT on general weighted instances, while for  $k$ -fold simultaneous MAX-CUT on unweighted instances, our Theorem 8.1.5 gives a  $(\frac{1}{2} + \Omega(\frac{1}{k^2}))$ -minimum approximation algorithm.

### 8.1.6 Organization of this Chapter

We first present the notation required for our algorithms in Section 8.2. We then describe our Pareto approximation algorithm for MAX-2-AND (which is equivalent to MAX-2-CSP<sub>2</sub>), and its generalization to MAX- $w$ -CSP <sub>$q$</sub>  in Sections 8.3 and 8.4 respectively. We then present our improved Pareto approximation for MAX- $w$ -SAT in Section 8.5.

We present the additional results in the remaining sections - The dichotomy theorem for the hardness of arbitrary CSPs is presented in Section 8.6, followed by our improved minimum approximation algorithm for unweighted MAX-CUT in Section 8.7, and the SDP integrality gaps in Section 8.8.

### 8.1.7 Related Work

The theory of *exact* multiobjective optimization has been very well studied, (see eg. [PY00, Dia11] and the references therein).

The only directly comparable work for simultaneous approximation algorithms for CSPs we are aware of is the work of Glaßer *et al.* [GRW11].<sup>6</sup> They give a  $1/2$ -Pareto approximation for MAX-SAT with a running time of  $n^{O(k^2)}$ . For bounded width clauses, our algorithm does better in both approximation guarantee and running time.

For MAX-CUT, there are a few results of a similar flavor. For two graphs, the results of Angel *et al.* [ABG06] imply a 0.439-Pareto approximation algorithm (though their actual results are incomparable to ours). Bollobás and Scott [BS04] asked what is

---

<sup>6</sup>They also give Pareto approximation results for simultaneous TSP (also see references therein).

the largest simultaneous cut in two unweighted graphs with  $m$  edges each. Kuhn and Osthus [KO07], using the second moment method, proved that for  $k$  simultaneous unweighted instances, there is a simultaneous cut that cuts at least  $m/2 - O(\sqrt{km})$  edges in each instance, and give a deterministic algorithm to find it (this leads to a  $(\frac{1}{2} - o(1))$ -Pareto approximation for unweighted instances with sufficiently many edges). Our main theorem implies the same Pareto approximation factor for simultaneous MAX-CUT on general weighted instances, while for  $k$ -fold simultaneous MAX-CUT on unweighted instances, our Theorem 8.1.5 gives a  $(\frac{1}{2} + \Omega(\frac{1}{k^2}))$ -minimum approximation algorithm.

## 8.2 Notation for the Main Algorithms

We now define some common notation that will be required for the following sections on algorithms for MAX-2-AND and for general MAX- $\mathcal{F}$ -CSP. For the latter, will stop referring to the set of predicates  $\mathcal{F}$ , and simply present an algorithm for the problem MAX- $w$ -CSP $_q$ : this is the MAX- $\mathcal{F}$ -CSP problem, where  $\mathcal{F}$  equals the set of all predicates on  $w$  variables from the domain  $[q]$ . For MAX-2-AND, the alphabet  $q$  and arity  $w$  are both 2.

Let  $V$  be a set of  $n$  variables. Each variable will take values from the domain  $[q]$ . Let  $\mathcal{C}$  denote a set of constraints of interest on  $V$  (for example, for studying MAX-2-AND,  $\mathcal{C}$  would be the set of AND constraints on pairs of literals of variables coming from  $V$ ). We use the notation  $v \in C$  to denote that the  $v$  is one of the variables that the constraint  $C$  depends on. Analogously, we denote  $T \subseteq C$  if  $C$  depends on all the variables in  $T$ . A weighted MAXCSP instance on  $V$  is given by a weight function  $\mathcal{W} : \mathcal{C} \rightarrow \mathbb{R}_+$ , where for  $C \in \mathcal{C}$ ,  $\mathcal{W}(C)$  is the weight of the constraint  $C$ . We will assume that  $\sum_{C \in \mathcal{C}} \mathcal{W}(C) = 1$ .

A partial assignment  $\rho$  is a pair  $(S_\rho, h_\rho)$ , where  $S_\rho \subseteq V$  and  $h_\rho : S_\rho \rightarrow [q]$ . (We also call a function  $h : S \rightarrow [q]$ , a partial assignment, when  $S$  is understood from the context). We say a constraint  $C \in \mathcal{C}$  is active given  $\rho$  if  $C$  depends on some variable in  $V \setminus S_\rho$ , and there exists full assignments  $g_0, g_1 : V \rightarrow [q]$  with  $g_i|_{S_\rho} = h_\rho$ , such that  $C$  evaluates to FALSE under the assignment  $g_0$  and  $C$  evaluates to TRUE under

the assignment  $g_1$ . (colloquially:  $C$ 's value is not fixed by  $\rho$ ). We denote by  $\text{Active}(\rho)$  the set of constraints from  $\mathcal{C}$  which are active given  $\rho$ . For a partial assignment  $\rho$  and  $C \in \mathcal{C} \setminus \text{Active}(\rho)$ , let  $C(\rho) = 1$  if  $C$ 's value is fixed to **TRUE** by  $\rho$ , and let  $C(\rho) = 0$  if  $C$ 's value is fixed to **FALSE** by  $\rho$ . For disjoint subsets  $S_1, S_2 \subseteq V$  and partial assignments  $f_1 : S_1 \rightarrow [q]$  and  $f_2 : S_2 \rightarrow [q]$ , let  $f = f_1 \cup f_2$  denote the assignment  $f : S_1 \cup S_2 \rightarrow [q]$  with  $f(x) = f_1(x)$  if  $x \in S_1$ , and  $f(x) = f_2(x)$  if  $x \in S_2$ . Abusing notation, for a partial assignment  $\rho$  and an assignment  $g : V \setminus S_\rho \rightarrow [q]$ , we often write  $\rho \cup g$  instead of  $h_\rho \cup g$ . For two constraints  $C_1, C_2 \in \mathcal{C}$ , we say  $C_1 \sim_\rho C_2$  if they share a variable that is contained in  $V \setminus S_\rho$ .

Define the *active degree given  $\rho$*  of a variable  $v \in V \setminus S_\rho$  by:

$$\text{activedegree}_\rho(v, \mathcal{W}) \stackrel{\text{def}}{=} \sum_{C \in \text{Active}(\rho), C \ni v} \mathcal{W}(C).$$

For a subset  $T \subseteq V \setminus S_\rho$  of variables, define its active degree given  $\rho$  by:

$$\text{activedegree}_\rho(T, \mathcal{W}) \stackrel{\text{def}}{=} \sum_{C \in \text{Active}(\rho), C \supseteq T} \mathcal{W}(C).$$

Define the active degree of the whole instance  $\mathcal{W}$  given  $\rho$ :

$$\text{activedegree}_\rho(\mathcal{W}) \stackrel{\text{def}}{=} \sum_{v \in V \setminus S_\rho} \text{activedegree}_\rho(v, \mathcal{W}).$$

For a partial assignment  $\rho$ , we define its value on an instance  $\mathcal{W}$  by:

$$\text{val}(\rho, \mathcal{W}) \stackrel{\text{def}}{=} \sum_{C \in \mathcal{C} \setminus \text{Active}(\rho)} \mathcal{W}(C)C(\rho).$$

Thus, for a total assignment  $f : V \rightarrow [q]$  extending  $\rho$ , we have the equality:

$$\text{val}(f, \mathcal{W}) - \text{val}(\rho, \mathcal{W}) = \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C)C(f).$$

### 8.3 Simultaneous MAX-2-AND

In this section, we give our approximation algorithm for simultaneous MAX-2-AND. Via a simple reduction given Section 8.4.1, this implies the  $q = w = 2$  case of our main theorem, Theorem 8.1.2.

### 8.3.1 Random Assignments

We begin by giving a sufficient condition for the value of a MAX-2-AND to be highly concentrated under independent random assignments to the variables.

Let  $\rho$  be a partial assignment. Let  $p : V \setminus S_\rho \rightarrow [0, 1]$  be such that  $p(v) \in [\frac{1}{4}, \frac{3}{4}]$  for each  $v \in V \setminus S_\rho$ . Let  $g : V \setminus S_\rho \rightarrow [q]$  be a random assignment obtained by sampling  $g(v)$  for each  $v$  independently with  $\mathbf{E}[g(v)] = p(v)$ . Define the random variable

$$Y \stackrel{\text{def}}{=} \text{val}(\rho \cup g, \mathcal{W}) - \text{val}(\rho, \mathcal{W}) = \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C)C(g).$$

The random variable  $Y$  measures the contribution of active constraints to  $\text{val}(\rho \cup g, \mathcal{W})$ . Note that the two quantities  $\mathbf{E}[Y]$  and  $\text{Var}[Y]$  can be computed efficiently given  $p$ . We denote these by  $\text{TrueMean}_\rho(p, \mathcal{W})$  and  $\text{TrueVar}_\rho(p, \mathcal{W})$ . The following lemma proves that either  $Y$  is concentrated, or there exists an active variable that contributes a significant fraction of the total active-degree of the instance.

**Lemma 8.3.1.** *Let  $p, Y$  be as above.*

1. *If  $\text{TrueVar}_\rho(p, \mathcal{W}) < \delta_0 \varepsilon_0^2 \cdot \text{TrueMean}_\rho(p, \mathcal{W})^2$  then  $\Pr[Y < (1 - \varepsilon_0) \mathbf{E}[Y]] < \delta_0$ .*
2. *If  $\text{TrueVar}_\rho(p, \mathcal{W}) \geq \delta_0 \varepsilon_0^2 \cdot \text{TrueMean}_\rho(p, \mathcal{W})^2$ , then there exists  $v \in V \setminus S_\rho$  such that*

$$\text{activedegree}_\rho(v, \mathcal{W}) \geq \frac{\varepsilon_0^2 \delta_0}{64} \cdot \text{activedegree}_\rho(\mathcal{W}).$$

The above lemma is a special case of Lemma 8.4.2 which is proved in Section 8.4.2, and hence we skip the proof. The first part is then a simple application of the Chebyshev inequality. For the second part, we use the assumption that  $\text{TrueVar}$  is large, to deduce that there exists a constraint  $C$  such that the total weight of constraints that share a variable from  $V \setminus S$  with  $C$ , i.e.,  $\sum_{C_2 \sim_S C} \mathcal{W}(C_2)$ , is large. It then follows that at least one variable  $v \in C$  must have large activedegree given  $S$ .

### 8.3.2 LP Relaxations

Let  $(c_\ell)_{\ell \in [k]}$  be the given target values for the Pareto approximation problem. Given a partial assignment  $\rho$ , we can write the feasibility linear program for simultaneous

MAX-2-AND as shown in Fig. 8.1, 8.2. In this LP, for a constraint  $C$ ,  $C^+$  ( $C^-$ ) denotes set of variables that appears as a positive (negative) literal in  $C$ .

For  $\vec{t}, \vec{z}$  satisfying linear constraints MAX2AND-LP<sub>1</sub>( $\rho$ ), let  $\text{smooth}(\vec{t})$  denote the map  $p : V \setminus S_\rho \rightarrow [0, 1]$  with  $p(v) = \frac{1}{4} + \frac{t_v}{2}$ . Note that  $p(v) \in [1/4, 3/4]$  for all  $v$ .

Given  $\vec{t}, \vec{z}$  satisfying MAX2AND-LP<sub>1</sub>, the rounding algorithm from [Tre98] samples each variable  $v$  independently with probability  $\text{smooth}(\vec{t})(v)$ . Note that this rounding algorithm is *smooth* in the sense that each variable is sampled independently with a probability that is bounded away from 0 and 1. This will be crucial for our algorithm. The following theorem from [Tre98] proves that this rounding algorithm finds a good integral assignment.

**Lemma 8.3.2** ([Tre98]). *Let  $\rho$  be a partial assignment.*

1. **Relaxation:** *For every  $g_0 : V \setminus S_\rho \rightarrow \{0, 1\}$ , there exist  $\vec{t}, \vec{z}$  satisfying MAX2AND-LP<sub>1</sub>( $\rho$ ) such that for every MAX-2-AND instance  $\mathcal{W}$ :*

$$\sum_{C \in \mathcal{C}} \mathcal{W}(C) z_C = \text{val}(g_0 \cup \rho, \mathcal{W}).$$

2. **Rounding:** *Suppose  $\vec{t}, \vec{z}$  satisfy MAX2AND-LP<sub>1</sub>( $\rho$ ). Then for every MAX-2-AND instance  $\mathcal{W}$ :*

$$\text{val}(\rho, \mathcal{W}) + \text{TrueMean}_\rho(\text{smooth}(\vec{t}), \mathcal{W}) \geq \frac{1}{2} \cdot \sum_{C \in \mathcal{C}} \mathcal{W}(C) z_C.$$

*Proof.* We begin with the first part. For  $v \in S_\rho$ , define  $t_v = \rho(v)$ . For  $v \in V \setminus S_\rho$ , define  $t_v = g_0(v)$ . For  $C \in \mathcal{C}$ , define  $z_C = 1$  if  $C(g_0 \cup \rho) = 1$ , and define  $z_C = 0$  otherwise. It is easy to see that these  $\vec{t}, \vec{z}$  satisfies MAX2AND-LP<sub>1</sub>( $\rho$ ), and that for every instance  $\mathcal{W}$ :

$$\sum_{C \in \mathcal{C}} \mathcal{W}(C) z_C = \text{val}(g_0 \cup \rho, \mathcal{W}).$$

Now we consider the second part. Let  $\mathcal{W}$  be any instance of MAX-2-AND. Let  $p = \text{smooth}(\vec{t})$ . Let  $g : V \setminus S_\rho \rightarrow \{0, 1\}$  be sampled as follows: independently for each

$v \in V \setminus S_\rho$ ,  $g(v)$  is sampled from  $\{0, 1\}$  such that  $\mathbf{E}[g(v)] = p(v)$ . We have:

$$\begin{aligned} \text{val}(\rho, \mathcal{W}) + \text{TrueMean}_\rho(\text{smooth}(\vec{t}), \mathcal{W}) &= \sum_{C \in \mathcal{C} \setminus \text{Active}(\rho)} \mathcal{W}(C)C(\rho) + \\ &\quad \mathbf{E} \left[ \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C)C(\rho \cup g) \right]. \end{aligned} \quad (8.3.1)$$

We will now understand the two terms of the right hand side.

For  $C \in \mathcal{C} \setminus \text{Active}(\rho)$ , it is easy to verify that if  $z_C > 0$ , we must have  $C(\rho) = 1$ .

Thus:

$$\sum_{C \in \mathcal{C} \setminus \text{Active}(\rho)} \mathcal{W}(C)C(\rho) \geq \sum_{C \in \mathcal{C} \setminus \text{Active}(\rho)} \mathcal{W}(C)z_C.$$

To understand the second term, we have the following claim.

**Claim 8.3.3.** *For  $C \in \text{Active}(\rho)$ ,  $\mathbf{E}[C(\rho \cup g)] \geq \frac{1}{2} \cdot z_C$ .*

*Proof.* Suppose there are exactly  $h$  variables in  $C$  which are not in  $S_\rho$ . We have  $h \leq 2$ .

$$\begin{aligned} \mathbf{E}[C(\rho \cup g)] &= \Pr[C \text{ is satisfied by } \rho \cup g] \\ &= \left( \prod_{v \in C^+, v \in V \setminus S_\rho} \frac{1}{4} + \frac{t_v}{2} \right) \cdot \left( \prod_{v \in C^-, v \in V \setminus S_\rho} \frac{1}{4} + \frac{1 - t_v}{2} \right) \\ &\geq \left( \frac{1}{4} + \frac{z_C}{2} \right)^h \geq \left( \frac{1}{4} + \frac{z_C}{2} \right)^2 \geq \frac{z_C}{2}. \end{aligned}$$

■

This claim implies that:

$$\mathbf{E} \left[ \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C)C(\rho \cup g) \right] \geq \frac{1}{2} \sum_{C \in \mathcal{C}} \mathcal{W}(C)z_C.$$

Substituting back into Equation (8.3.1), we get the Lemma. ■

### 8.3.3 The Algorithm

We now give our Pareto approximation algorithm for MAX-2-AND in Fig. 8.3.

$$\begin{aligned}
z_C &\leq t_v & \forall C \in \mathcal{C}, v \in C^+ \\
z_C &\leq 1 - t_v & \forall C \in \mathcal{C}, v \in C^- \\
1 \geq t_v &\geq 0 & \forall v \in V \setminus S_\rho \\
t_v &= h_\rho(v) & \forall v \in S_\rho
\end{aligned}$$

Figure 8.1: Linear inequalities MAX2AND-LP<sub>1</sub>( $\rho$ )

$$\begin{aligned}
&\sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) \cdot z_C \geq c_\ell \quad \forall \ell \in [k] \\
&\vec{t}, \vec{z} \text{ satisfy MAX2AND-LP}_1(\rho).
\end{aligned}$$

Figure 8.2: Linear inequalities MAX2AND-LP<sub>2</sub>( $\rho$ )

**Input:**  $k$  instances of MAX-2-AND  $\mathcal{W}_1, \dots, \mathcal{W}_k$  on the variable set  $V$ ,  $\varepsilon > 0$  and target objective values  $c_1, \dots, c_k$ .

**Output:** An assignment to  $V$

**Parameters:**  $\delta_0 = \frac{1}{10(k+1)}$ ,  $\varepsilon_0 = \varepsilon$ ,  $\gamma = \frac{\varepsilon_0^2 \delta_0}{16}$ ,  $t = \left\lceil \frac{20k^2}{\gamma} \log \frac{k}{\gamma} \right\rceil$

1. Initialize tree  $T$  to be an empty quaternary tree (i.e., just 1 root node). Nodes of the tree will be indexed by strings in  $(\{0, 1\}^2)^*$ .
2. With each node  $\nu$  of the tree, we associate:
  - (a) A partial assignment  $\rho_\nu$ .
  - (b) A special pair of variables  $\mathcal{T}_\nu^1, \mathcal{T}_\nu^2 \in V \setminus S_{\rho_\nu}$ .
  - (c) A special instance  $\mathcal{I}_\nu \in [k]$ .
  - (d) A collection of integers  $\text{count}_{\nu,1}, \dots, \text{count}_{\nu,k}$ .
  - (e) A trit representing whether the node  $\nu$  is living, dead, or exhausted.
3. Initialize the root node  $\nu_0$  to (1)  $\rho_{\nu_0} \leftarrow (\emptyset, \emptyset)$ , (2)  $\forall \ell \in [k], \text{count}_{\nu_0, \ell} \leftarrow 0$ , (3) living.
4. While there is a living leaf  $\nu$  of  $T$ , do the following:
  - (a) Check the feasibility of linear inequalities  $\text{MAX2AND-LP}_2(\rho_\nu)$ .
    - i. If there is a feasible solution  $\vec{t}, \vec{z}$ , then define  $p_\nu : V \setminus S_{\rho_\nu} \rightarrow [0, 1]$  as  $p_\nu = \text{smooth}(\vec{t})$ .
    - ii. If not, then declare  $\nu$  to be dead and return to Step 4.
  - (b) For each  $\ell \in [k]$ , compute  $\text{TrueVar}_{\rho_\nu}(p_\nu, \mathcal{W}_\ell)$  and  $\text{TrueMean}_{\rho_\nu}(p_\nu, \mathcal{W}_\ell)$ .
  - (c) If  $\text{TrueVar}_{\rho_\nu}(p_\nu, \mathcal{W}_\ell) \geq \delta_0 \varepsilon_0^2 \cdot \text{TrueMean}_{\rho_\nu}(p_\nu, \mathcal{W}_\ell)^2$ , then set  $\text{flag}_\ell \leftarrow \text{TRUE}$ , else set  $\text{flag}_\ell \leftarrow \text{FALSE}$ .

- (d) Choose the smallest  $\ell \in [k]$ , such that  $\text{count}_\ell < t$  AND  $\text{flag}_\ell = \text{TRUE}$  (if any):
    - i. Find  $x \in V \setminus S_{\rho_\nu}$  that maximizes  $\text{activedegree}_{\rho_\nu}(x, \mathcal{W}_\ell)$ . Note that it will satisfy  $\text{activedegree}_{\rho_\nu}(x, \mathcal{W}_\ell) \geq \gamma \cdot \text{activedegree}_{\rho_\nu}(\mathcal{W}_\ell)$ .
    - ii. Among all the active constraints  $C \in \mathcal{C}$  such that  $x \in C$  and  $C \cap S_{\rho_\nu} = \emptyset$ , find the one that maximizes  $\mathcal{W}_\ell(C)$ . Call this constraint  $C^*$ . Let  $y$  be the other variable contained in  $C^*$  (if there is no other variable, set  $y = x$ ).  
Set  $\mathcal{T}_\nu^1 \leftarrow x$  and  $\mathcal{T}_\nu^2 \leftarrow y$ . Set  $\mathcal{I}_\nu \leftarrow \ell$ .
    - iii. Create four children of  $\nu$ , with labels  $\nu b_1 b_2$  for each  $b_1, b_2 \in \{0, 1\}$  and set
      - $\rho_{\nu b_1 b_2} \leftarrow (S_{\rho_\nu} \cup \{\mathcal{T}_\nu^1, \mathcal{T}_\nu^2\}, h^{b_1 b_2})$ , where  $h^{b_1 b_2}$  extends  $h_{\rho_\nu}$  by  $h^{b_1 b_2}(\mathcal{T}_\nu^1) = b_1$  and  $h^{b_1 b_2}(\mathcal{T}_\nu^2) = b_2$ .
      - $\forall \ell' \in [k]$  with  $\ell' \neq \ell$ , set  $\text{count}_{\nu b_1 b_2, \ell'} \leftarrow \text{count}_{\nu, \ell'}$ . Set  $\text{count}_{\nu b_1 b_2, \ell} \leftarrow \text{count}_{\nu, \ell} + 1$ .
      - Set  $\nu b_1 b_2$  to be living.
  - (e) If no such  $\ell$  exists, declare  $\nu$  to be exhausted.
- 5. Now every leaf of  $T$  is either exhausted or dead. For each exhausted leaf  $\nu$  of  $T$ :
  - (a) Let  $g_\nu : V \setminus S_{\rho_\nu} \rightarrow \{0, 1\}$  be a random assignment where, for each  $v \in V \setminus S_{\rho_\nu}$ ,  $g_\nu(v)$  is sampled independently with  $\mathbf{E}[g_\nu(v)] = p_\nu(v)$ . Note that  $\mathbf{E}[g_\nu(v)] \in [\frac{1}{4}, \frac{3}{4}]$ .
  - (b) For every assignment  $h : S_{\rho_\nu} \rightarrow \{0, 1\}$ , compute  $\text{out}_{h, g_\nu} \leftarrow \min_{\ell \in [k]} \frac{\text{val}(h \cup g_\nu, \mathcal{W}_\ell)}{c_\ell}$ . If  $c_\ell = 0$  for some  $\ell \in [k]$ , we interpret  $\frac{\text{val}(h \cup g_\nu, \mathcal{W}_\ell)}{c_\ell}$  as  $+\infty$ .
- 6. Output the largest  $\text{out}_{h, g_\nu}$  seen, and the assignment  $h \cup g_\nu$  that produced it.

Figure 8.3: Algorithm SIM-MAX2AND for approximating weighted simultaneous MAX-2-AND

### 8.3.4 Analysis

Notice that the depth of the tree  $T$  is at most  $kt$ , and that for every  $\nu$ , we have that  $|S_{\rho_\nu}| \leq 2kt$ . This implies that the running time is at most  $2^{O(kt)} \cdot \text{poly}(n)$ .

Let  $f^* : V \rightarrow \{0, 1\}$  be an assignment such that  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for each  $\ell \in [k]$ . Let  $\nu^*$  be the unique leaf of the tree  $T$  for which  $\rho_{\nu^*}$  is consistent with  $f^*$ . (This  $\nu^*$  can be found as follows: start with  $\nu$  equal to the root. Set  $\nu$  to equal the unique child of  $\nu$  for which  $\rho_\nu$  is consistent with  $f^*$ , and repeat until  $\nu$  becomes a leaf. This leaf is  $\nu^*$ ). Observe that since  $f^*$  is an assignment such that  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for every  $\ell \in [k]$ , by picking  $g_0 = f^*|_{V \setminus S^*}$  in part 1 of Lemma 8.3.2, we know that  $\text{MAX2AND-LP}_2(\rho^*)$  is feasible, and hence  $\nu^*$  must be an exhausted leaf (and not dead).

Define  $\rho^* = \rho_{\nu^*}$ ,  $S^* = S_{\rho^*}$ ,  $h^* = h_{\rho^*}$ , and  $p^* = p_{\nu^*}$ . At the completion of Step 4, if  $\ell \in [k]$  satisfies  $\text{count}_{\nu^*, \ell} = t$ , we call instance  $\ell$  a *high variance* instance. Otherwise we call instance  $\ell$  a *low variance* instance.

#### Low Variance Instances.

First we show that for the leaf  $\nu^*$  in Step 5, combining the partial assignment  $h^*$  with a random assignment  $g_{\nu^*}$  in step 5(b) is good for any low variance instances with high probability.

**Lemma 8.3.4.** *Let  $\ell \in [k]$  be any low variance instance. For the leaf node  $\nu^*$ , let  $g_{\nu^*}$  be the random assignment sampled in Step 5.(a). of SIM-MAX2AND. Then with probability at least  $1 - \delta_0$ , the assignment  $f = h^* \cup g_{\nu^*}$  satisfies:*

$$\Pr_{g_{\nu^*}} [\text{val}(f, \mathcal{W}_\ell) \geq (1/2 - \varepsilon/2) \cdot c_\ell] \geq 1 - \delta_0.$$

*Proof.* For every low variance instance  $\ell$ , we have that  $\text{TrueVar}_{\rho_{\nu^*}}(p^*, \mathcal{W}_\ell) < \delta_0 \varepsilon_0^2 \cdot \text{TrueMean}_{\rho_{\nu^*}}(p^*, \mathcal{W}_\ell)^2$ . Define  $Y \stackrel{\text{def}}{=} \text{val}(\rho^* \cup g_{\nu^*}, \mathcal{W}_\ell) - \text{val}(\rho^*, \mathcal{W}_\ell)$ . By Lemma 8.3.1, we

have  $\Pr[Y < (1 - \varepsilon_0) \mathbf{E}[Y]] < \delta_0$ . Thus, with probability at least  $1 - \delta_0$ , we have,

$$\begin{aligned}
\text{val}(f, \mathcal{W}_\ell) &\geq \text{val}(\rho^*, \mathcal{W}_\ell) + (1 - \varepsilon_0) \mathbf{E}[Y] \\
&= \text{val}(\rho^*, \mathcal{W}_\ell) + (1 - \varepsilon_0) \cdot \text{TrueMean}_{\rho^*}(\text{smooth}(\vec{t}), \mathcal{W}_\ell) \\
&= (1 - \varepsilon_0) \cdot (\text{val}(\rho^*, \mathcal{W}_\ell) + \text{TrueMean}_{\rho^*}(\text{smooth}(\vec{t}), \mathcal{W}_\ell)) \\
&\geq \frac{1}{2} \cdot (1 - \varepsilon_0) \cdot \sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) \cdot z_C \geq \frac{1}{2} \cdot (1 - \varepsilon_0) \cdot c_\ell \geq \left(\frac{1}{2} - \frac{\varepsilon}{2}\right) \cdot c_\ell,
\end{aligned}$$

where we have used the second part of Lemma 8.3.2. ■

Next, we will consider a small perturbation of  $h^*$  which will ensure that the algorithm performs well on high variance instances too. We will ensure that this perturbation does not affect the success on the low variance instances.

### High Variance Instances.

Fix a high variance instance  $\ell$ . Let  $\nu$  be an ancestor of  $\nu^*$  with  $\mathcal{I}_\nu = \ell$ . Define:

$$\text{activedegree}_\nu \stackrel{\text{def}}{=} \text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu^1, \mathcal{W}_\ell).$$

Let  $\mathcal{C}_\nu$  be the set of all constraints  $C$  containing  $\mathcal{T}_\nu^1$  which are active given  $\rho_\nu$ . We call a constraint  $C$  in  $\mathcal{C}_\nu$  a **backward** constraint if  $C$  only involves variables from  $S_{\rho_\nu} \cup \{\mathcal{T}_\nu^1\}$ . Otherwise we call  $C$  in  $\mathcal{C}_\nu$  a **forward** constraint. Let  $\mathcal{C}_\nu^{\text{backward}}$  and  $\mathcal{C}_\nu^{\text{forward}}$  denote the sets of these constraints. Finally, we denote  $\mathcal{C}_\nu^{\text{out}}$  the set of binary constraints on  $\mathcal{T}_\nu^1$  and a variable from  $V \setminus S^*$ .

Define **backward degree** and **forward degree** as follows:

$$\begin{aligned}
\text{backward}_\nu &\stackrel{\text{def}}{=} \sum_{C \in \mathcal{C}_\nu^{\text{backward}}} \mathcal{W}_\ell(C), \\
\text{forward}_\nu &\stackrel{\text{def}}{=} \sum_{C \in \mathcal{C}_\nu^{\text{forward}}} \mathcal{W}_\ell(C).
\end{aligned}$$

Note that:

$$\text{activedegree}_\nu = \text{backward}_\nu + \text{forward}_\nu.$$

Now we consider variable  $\mathcal{T}_\nu^2$ . Let  $\text{heaviest}_\nu$  be the total  $\mathcal{W}_\ell$  weight of all the constraints containing both  $\mathcal{T}_\nu^1$  and  $\mathcal{T}_\nu^2$ . Based on all this, we classify  $\nu$  into one of three categories:

1. If  $\text{backward}_\nu \geq \frac{1}{2} \cdot \text{activedegree}_\nu$ , then we call  $\nu$  a **typeA** node.
2. Otherwise, if  $\text{heaviest}_\nu \geq \frac{1}{100tk} \cdot \text{activedegree}_\nu$ , then we call  $\nu$  a **typeB** node. In this case we have some  $\mathcal{W}_\ell$  constraint  $C$  containing  $\mathcal{T}_\nu^1$  and  $\mathcal{T}_\nu^2$  with  $\mathcal{W}_\ell(C) \geq \frac{1}{1600tk} \cdot \text{activedegree}_\nu$ .
3. Otherwise, we call  $\nu$  a **typeC** node. In this case, for every  $v \in V \setminus S_{\rho_\nu}$ , the total weight of the constraints involving  $v$  and  $\mathcal{T}_\nu^1$ , i.e.,  $\text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu^1 \cup v, \mathcal{W}_\ell)$  is bounded by  $\frac{1}{100tk} \cdot \text{activedegree}_\nu$ . In particular, every constraint  $C \in \mathcal{C}_\nu^{\text{forward}}$  must have  $\mathcal{W}_\ell(C) < \frac{1}{100tk} \cdot \text{activedegree}_\nu$ . Since  $|S^*| \leq 2tk$ , the total weight of constraints containing  $\mathcal{T}_\nu^1$  and some variable in  $S^* \setminus S_{\rho_\nu}$  is at most  $|S^* \setminus S_{\rho_\nu}| \cdot \frac{1}{100tk} \cdot \text{activedegree}_\nu$  which is at most  $\frac{2}{100} \cdot \text{activedegree}_\nu$ . Hence we have:

$$\begin{aligned} \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) &= \text{forward}_\nu - \left\{ \text{total weight of constraints containing } \mathcal{T}_\nu^1 \text{ and some variable in } S^* \setminus S_{\rho_\nu} \right\} \\ &\geq \left( \frac{1}{2} - \frac{2}{100} \right) \text{activedegree}_\nu > \frac{1}{4} \cdot \text{activedegree}_\nu. \end{aligned}$$

For nodes  $\nu$  which are **typeC**, the variable  $\mathcal{T}_\nu^1$  has a large fraction of its active degree coming from constraints between  $\mathcal{T}_\nu^1$  and  $V \setminus S^*$ .

For a partial assignment  $g : V \setminus S^* \rightarrow \{0, 1\}$ , we say that  $g$  is **Cgood** for  $\nu$  if there exists a setting of variable  $\mathcal{T}_\nu^1$  that satisfies at least  $\frac{1}{64} \cdot \text{activedegree}_\nu$  weight amongst constraints containing variable  $\mathcal{T}_\nu^1$  and some other variable in  $V \setminus S^*$ . The next lemma shows that for every **typeC** node  $\nu$ , with high probability, the random assignment  $g_{\nu^*} : V \setminus S^* \rightarrow \{0, 1\}$  is **Cgood** for  $\nu$ .

**Lemma 8.3.5.** *Consider a **typeC** node  $\nu$ . Suppose  $g : V \setminus S^* \rightarrow \{0, 1\}$  is a partial assignment obtained by independently sampling  $g(v)$  with  $\mathbf{E}[g(v)] \in [1/4, 3/4]$  for each  $v \in V \setminus S^*$ . Then:*

$$\Pr_g[g \text{ is Cgood for } \nu] \geq 1 - 2 \cdot e^{-tk/100}.$$

*Proof.* Let  $\ell = \mathcal{I}_\nu$ .

For each constraint  $C \in \mathcal{C}_\nu^{\text{out}}$  and each  $g : \{0, 1\}^{V \setminus S^*} \rightarrow \{0, 1\}$ , define  $Z_C^{(1)}(g), Z_C^{(0)}(g) \in \{0, 1\}$  as follows.  $Z_C^{(1)}(g)$  equals 1 iff  $C$  is satisfied by extending the assignment  $g$  with

$\mathcal{T}_\nu^1 \leftarrow 1$ . Similarly,  $Z_C^{(0)}(g)$  equals 1 iff  $C$  is satisfied by extending the assignment  $g$  with  $\mathcal{T}_\nu^1 \leftarrow 0$ .

For  $b = 0, 1$ , we define  $\text{score}^{(b)} : \{0, 1\}^{V \setminus S^*} \rightarrow \mathbb{R}$  as follows:

$$\text{score}^{(b)}(g) \stackrel{\text{def}}{=} \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \cdot Z_C^{(b)}(g).$$

In words,  $\text{score}^{(b)}(g)$  is the total weight of constraints between  $\mathcal{T}_\nu^1$  and  $V \setminus S^*$  satisfied by setting  $\mathcal{T}_\nu^1$  to  $b$  and setting  $V \setminus S^*$  according to  $g$ .

Note that since  $g(v)$  is sampled independently for  $v \in V \setminus S^*$  with  $\mathbf{E}[g(v)] \in [1/4, 3/4]$ , we have  $\mathbf{E}_g[Z_C^{(1)}(g) + Z_C^{(0)}(g)] \geq \frac{1}{4}$ . Thus:

$$\begin{aligned} \mathbf{E}_g[\text{score}^{(1)}(g) + \text{score}^{(0)}(g)] &= \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \mathbf{E}[Z_C^{(1)}(g)] + \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \mathbf{E}[Z_C^{(0)}(g)] \\ &\geq \frac{1}{4} \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C). \end{aligned}$$

So one of  $\mathbf{E}[\text{score}^{(1)}(g)]$  and  $\mathbf{E}[\text{score}^{(0)}(g)]$  is at least  $\frac{1}{8} \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \geq \frac{1}{32} \text{activedegree}_\nu$ . Suppose it is  $\mathbf{E}[\text{score}^{(1)}(g)]$  (the other case is identical). We are going to use McDiarmid's inequality to show the concentration of  $\text{score}^{(1)}(g)$  around its mean.<sup>7</sup>

Since  $\nu$  is **typeC**, we know that for every vertex  $v \in V \setminus S^*$ , changing  $g$  on just  $v$  can change the value of  $\text{score}^{(1)}(g)$  by at most  $c_v \stackrel{\text{def}}{=} \text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu^1 \cup v, \mathcal{W}_\ell) \leq$

---

<sup>7</sup>In this case we could have simply used a Hoeffding-like inequality, but later when we handle larger-width constraints we will truly use the added generality of McDiarmid's inequality.

$\frac{1}{100tk} \cdot \text{activedegree}_\nu$ . Thus by McDiarmid's inequality (Lemma 8.8.5),

$$\begin{aligned}
\Pr_g[g \text{ is not Cgood for } \nu] &\leq \Pr_g \left[ \text{score}^{(1)}(g) < \frac{1}{64} \cdot \text{activedegree}_\nu \right] \\
&\leq \Pr_g \left[ |\text{score}^{(1)}(g) - \mathbf{E}_g[\text{score}^{(1)}(g)]| > \frac{1}{64} \cdot \text{activedegree}_\nu \right] \\
&\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_\nu^2}{(64)^2 \sum_{v \in V \setminus S^*} c_v^2} \right) \\
&\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_\nu^2}{(64)^2 \cdot (\max_v c_v) \cdot \sum_{v \in V \setminus S^*} c_v} \right) \\
&\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_\nu^2}{(64)^2 \cdot (\max_v c_v) \cdot \text{activedegree}_\nu} \right) \\
&\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_\nu}{(64)^2 \cdot (\frac{\text{activedegree}_\nu}{100tk})} \right) \\
&\leq 2 \cdot \exp \left( \frac{-200tk}{(64)^2} \right) \leq 2 \cdot \exp \left( \frac{-tk}{100} \right).
\end{aligned}$$

■

For a high variance instance  $\ell$ , let  $\nu_1^\ell, \dots, \nu_t^\ell$  be the sequence of  $t$  nodes with  $\mathcal{I}_\nu = \ell$  which lie on the path from the root to  $\nu^*$ . Set  $\text{finalwt}_\ell = \text{activedegree}_{\rho^*}(\mathcal{W}_\ell)$  (in words: this is the active degree left over in instance  $\ell$  after the restriction  $\rho^*$ ).

**Lemma 8.3.6.** *For every high variance instance  $\ell \in [k]$  and for each  $i \leq \lfloor t/2 \rfloor$ ,*

$$\text{activedegree}_{\nu_i^\ell} \geq \gamma \cdot (1 - \gamma)^{-t/2} \cdot \text{finalwt}_\ell \geq 1600tk \cdot \text{finalwt}_\ell.$$

*Proof.* Fix a high variance instance  $\ell \in [k]$ . Note that  $b_i = \text{activedegree}_{\rho_{\nu_i^\ell}}(\mathcal{W}_\ell)$  decreases as  $i$  increases. The main observation is that

1.  $b_{i+1} \leq (1 - \gamma) \cdot b_i$ .
2.  $\text{activedegree}_{\nu_i^\ell} \geq \gamma b_i$ .

Thus for all  $\nu_i^\ell$  with  $i \in \{1, \dots, t/2\}$ , we have  $\text{activedegree}_{\nu_i^\ell} \geq \gamma \cdot (1 - \gamma)^{-t/2} \cdot \text{finalwt}_\ell$  and also the choice of parameters implies for those  $\nu_i^\ell$   $\text{activedegree}_{\nu_i^\ell}$  is at least  $1600tk \cdot \text{finalwt}_\ell$ . ■

### Putting Everything Together.

We now show that when  $\nu$  is taken to equal  $\nu^*$  in Step 5, then with high probability over the choice of  $g$  in Step 5(a) there is a setting of  $h$  in Step 5(b) such that  $\forall \ell \in [k], \text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (\frac{1}{2} - \varepsilon) \cdot c_\ell$ .

**Theorem 8.3.7.** *Suppose the algorithm SIM-MAX2AND is given as inputs  $\varepsilon > 0$ ,  $k$  simultaneous weighted MAX-2-AND instances  $\mathcal{W}_1, \dots, \mathcal{W}_k$  on  $n$  variables, and target objective value  $c_1, \dots, c_k$  with the guarantee that there exists an assignment  $f^*$  such that for each  $\ell \in [k]$ , we have  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$ . Then, the algorithm runs in  $2^{O(k^4/\varepsilon^2 \log(k/\varepsilon))} \cdot \text{poly}(n)$  time, and with probability at least 0.9, outputs an assignment  $f$  such that for each  $\ell \in [k]$ , we have,  $\text{val}(f, \mathcal{W}_\ell) \geq (\frac{1}{2} - \varepsilon) \cdot c_\ell$ .*

*Proof.* Consider the case when  $\nu$  is taken to equal  $\nu^*$  in Step 5. By Lemma 8.3.4, with probability at least  $1 - k\delta_0$  over the choice random choices of  $g_{\nu^*}$ , we have that for every low variance instance  $\ell \in [k]$ ,  $\text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (\frac{1}{2} - \frac{\varepsilon}{2}) \cdot c_\ell$ . By Lemma 8.3.5 and a union bound, with probability at least  $1 - \frac{t}{2} \cdot k \cdot 2e^{-tk/100} \geq 1 - \delta_0$  over the choice of  $g_{\nu^*}$ , for every high variance instance  $\ell$  and for every typeC node  $\nu_i^\ell$ ,  $i \in [t/2]$ , we have that  $g_{\nu^*}$  is Cgood for  $\nu_i^\ell$ . Thus with probability at least  $1 - (k+1)\delta_0$ , both these events occur. Henceforth we assume that both these events occur in Step 5(a) of the algorithm.

Our next goal is to show that there exists a partial assignment  $h : S^* \rightarrow \{0, 1\}$  such that:

1. For every instance  $\ell \in [k]$ ,  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \frac{\varepsilon}{2}) \cdot \text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell)$ .
2. Moreover, for every high variance instance  $\ell \in [k]$ ,  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \frac{\varepsilon}{2}) \cdot \text{finalwt}_\ell$ .

Before giving a proof of the existence of such an  $h$ , we show that this completes the proof of the theorem. We claim that when the partial assignment  $h$  guaranteed above is considered in the Step 5(b) in the algorithm, we obtain an assignment with the required approximation guarantees.

For every low variance instance  $\ell \in [k]$ , since we started with  $\text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (\frac{1}{2} - \frac{\varepsilon}{2}) \cdot c_\ell$ , property 1 above implies that every low variance instance  $\text{val}(h \cup g_{\nu^*}) \geq (\frac{1}{2} - \varepsilon) \cdot c_\ell$ . For every high variance instance  $\ell \in [k]$ , since  $h^* = f^*|_S$ ,

$$\text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \text{val}(f^*, \mathcal{W}_\ell) - \text{activedegree}_{\rho^*}(\mathcal{W}_\ell) \geq c_\ell - \text{finalwt}_\ell.$$

Combining this with properties 1 and 2 above, we get,

$$\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot \max\{c_\ell - \text{finalwt}_\ell, \text{finalwt}_\ell\} \geq 1/2 \cdot (1 - \varepsilon/2) \cdot c_\ell.$$

Thus, for all instances  $\ell \in [k]$ , we get  $\text{val}(h \cup g_{\nu^*}) \geq (1/2 - \varepsilon) \cdot c_\ell$ .

Now, it remains to show the existence of such an  $h$  by giving a procedure for constructing  $h$  by perturbing  $h^*$  (Note that this procedure is only part of the analysis). For nodes  $\nu, \nu'$  in the tree, let us write  $\nu \prec \nu'$  if  $\nu$  is an ancestor of  $\nu'$ , and we also say that  $\nu'$  is “deeper” than  $\nu$ .

### Constructing $h$ .

1. Initialize  $H \subseteq [k]$  to be the set of high variance instances.
2. Let  $N_0 = \{\nu_i^\ell \mid \ell \in H, i \in [t/2]\}$ . Note that  $N$  is a chain in the tree (since all the elements of  $N$  are ancestors of  $\nu^*$ ). Since every  $\nu \in N$  is an ancestor of  $\nu^*$ , we have  $h_{\rho\nu} = h^*|_{S_{\rho\nu}}$ .
3. Initialize  $D = \emptyset$ ,  $N = N_0$ ,  $h = h^*$ .
4. During the procedure, we will be changing the assignment  $h$ , and removing elements from  $N$ . We will always maintain the following two invariants:
  - $|N| > \frac{t}{4}$ .
  - For every  $\nu \in N$ ,  $h|_{S_{\rho\nu}} = h^*|_{S_{\rho\nu}}$ .
5. While  $|D| \neq |H|$  do:
  - (a) Let

$$B = \left\{ v \in V \mid \exists \ell \in [k] \text{ with } \sum_{C \in \mathcal{C}, C \ni v} \mathcal{W}_\ell(C) \cdot C(h \cup g_{\nu^*}) \geq \frac{\varepsilon}{4k} \text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \right\}.$$

Note that  $|B| \leq \frac{8k^2}{\varepsilon} < \frac{t}{8}$ .

(b) Let  $\nu \in N$  be the deepest element of  $N$  for which:  $\{\mathcal{T}_\nu^1, \mathcal{T}_\nu^2\} \cap B = \emptyset$ .

Such a  $\nu$  exists because:

- $|N| > \frac{t}{4} > |B|$ , and
- there are at most  $|B|$  nodes  $\nu$  for which  $\{\mathcal{T}_\nu^1, \mathcal{T}_\nu^2\} \cap B \neq \emptyset$  (since  $\{\mathcal{T}_\nu^1, \mathcal{T}_\nu^2\}$  are all disjoint for distinct  $\nu$ ).

(c) Let  $\ell \in H, i \in [t/2]$  be such that  $\nu = \nu_i^\ell$ . Let  $x = \mathcal{T}_\nu^1$  and  $y = \mathcal{T}_\nu^2$ . Let  $\rho = \rho_\nu$ . We will now see a way of modifying the values of  $h(x)$  and  $h(y)$  to guarantee that  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \text{finalwt}_\ell$ . The procedure depends on whether  $\nu$  is **typeA**, **typeB**, or **typeC**.

- If  $\nu$  is **typeA**, then we know that  $\text{backward}_\nu \geq \frac{1}{2} \cdot \text{activedegree}_\nu \geq 2 \cdot \text{finalwt}_\ell$ .

The second invariant tells us that  $\rho = h^*|_{S_\rho} = h|_{S_\rho}$ . Thus we have:

$$\begin{aligned} \text{backward}_\nu &= \sum_{C \in \mathcal{C}_\nu^{\text{backward}}} \mathcal{W}_\ell(C) \\ &= \sum_{C \subseteq S_\rho \cup \{x\}, C \ni x, C \in \text{Active}(\rho)} \mathcal{W}_\ell(C) \\ &= \sum_{C \subseteq S_\rho \cup \{x\}, C \ni x, C \in \text{Active}(h|_{S_\rho})} \mathcal{W}_\ell(C). \end{aligned}$$

This implies that we can choose a setting of  $h(x) \in \{0, 1\}$  such that the total sum of weights of those constraints containing  $x$  which are satisfied by  $h$  is:

$$\begin{aligned} \sum_{C \subseteq S_\rho \cup \{x\}, C \ni x, C \in \text{Active}(h|_{S_\rho})} \mathcal{W}_\ell(C) &\geq \frac{1}{2} \sum_{C \subseteq S_\rho \cup \{x\}, C \ni x, C \in \text{Active}(h|_{S_\rho})} \mathcal{W}_\ell(C) \\ &= \frac{1}{2} \cdot \text{backward}_\nu \\ &\geq \frac{1}{4} \cdot \text{activedegree}_\nu \\ &\geq \text{finalwt}_\ell, \quad (\text{by Lemma 8.3.6}) \end{aligned}$$

where the  $\frac{1}{2}$  in the first inequality is because the variable can appear as a *positive* literal or a *negative* literal in those **backward** constraints. In particular, after making this change, we have  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \text{finalwt}_\ell$ .

- ii. If  $\nu$  is **typeB**, then we know that some constraint  $C$  containing  $x$  and  $y$  has  $\mathcal{W}_\ell(C) \geq \frac{1}{1600tk} \cdot \text{activedegree}_\nu \geq \text{finalwt}_\ell$ . Thus we may choose settings for  $h(x), h(y) \in \{0, 1\}$  such that  $C(h) = 1$ . Thus, after making this assignment to  $h(x)$  and  $h(y)$ , we have  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq \text{finalwt}_\ell$ .
- iii. If  $\nu$  is **typeC**, since  $g_{\nu^*}$  is **Cgood** for  $\nu$ , we can choose a setting of  $h(x)$  so that the total weight of satisfied constraints in  $\mathcal{W}_\ell$  between  $x$  and  $V \setminus S^*$  is at least  $\frac{1}{64} \cdot \text{activedegree}_\nu \geq \text{finalwt}_\ell$ . After this change, we have  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \text{finalwt}_\ell$ .

In all the above 3 cases, we only changed the value of  $h$  at the variables  $x, y$ . Since  $\{x, y\} \cap B = \emptyset$ , we have that for every  $j \in [k]$ , the new value  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_j)$  is at least  $(1 - \frac{\varepsilon}{2k})$  times the old value  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_j)$ .

- (d) Set  $D = D \cup \{\ell\}$ .
- (e) Set  $N = \{\nu_i^\ell \mid \ell \in H \setminus D, i \leq \lfloor t/2 \rfloor, \nu_i^\ell \prec \nu\}$ .

Observe that  $|N|$  decreases in size by at most  $\frac{t}{2} + |B|$ . Thus, if  $D \neq H$ , we have

$$\begin{aligned}
 |N| &\geq |N_0| - |D| \cdot \frac{t}{2} - |D||B| \\
 &= |H| \cdot \frac{t}{2} - |D| \cdot \frac{t}{2} - |D||B| \\
 &\geq \frac{t}{2} - k|B| > \frac{t}{4}
 \end{aligned}$$

Also observe that we only changed the values of  $h$  at the variables  $\mathcal{T}_\nu^1$  and  $\mathcal{T}_\nu^2$ . Thus for all  $\nu' \preceq \nu$ , we still have the property that  $h|_{S_{\rho_{\nu'}}} = h^*|_{S_{\rho_{\nu'}}}$ .

For each high variance instance  $\ell \in [k]$ , in the iteration where  $\ell$  gets added to the set  $D$ , the procedure ensures that at the end of the iteration  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \text{finalwt}_\ell$ .

Moreover, at each step we reduced the value of each  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell)$  by at most  $\frac{\varepsilon}{2k}$  fraction of its previous value. Thus, at the end of the procedure, for every  $\ell \in [k]$ , the value has decreased at most by a multiplicative factor of  $(1 - \frac{\varepsilon}{2k})^k \geq (1 - \frac{\varepsilon}{2})$ . Thus, for every  $\ell \in [k]$ , we get  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \frac{\varepsilon}{2}) \cdot \text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell)$ , and for every high variance instance  $\ell \in [k]$ , we have  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \frac{\varepsilon}{2}) \cdot \text{finalwt}_\ell$ . This proves the two properties of  $h$  that we set out to prove.

**Running time :** Running time of the algorithm is  $2^{O(kt)} \cdot \text{poly}(n)$  which is  $2^{O(k^4/\varepsilon^2 \log(k/\varepsilon^2))} \cdot \text{poly}(n)$ . ■

## 8.4 Simultaneous MAX- $w$ -CSP $_q$

In this section, we give our simultaneous approximation algorithm for MAX- $w$ -CSP $_q$ , and thus prove Theorem 8.1.2.

### 8.4.1 Reduction to Simple Constraints

For the problem MAX- $w$ -CSP $_q$ ,  $\mathcal{C}$  is the set of all possible  $q$ -ary constraints on  $V$  with arity at most  $w$ , *i.e.*, each constraint is of the form  $C_f : [q]^T \rightarrow \{0, 1\}$  depending only on the values of variables in an ordered tuple  $T \subseteq V$  with  $|T| \leq w$ . As a first step (mainly to simplify notation), we give a simple approximation preserving reduction which replaces  $\mathcal{C}$  with a smaller set of constraints. We will then present our main algorithm

Define a  $w$ -term to be a constraint  $C$  on exactly  $w$  variables which has exactly 1 satisfying assignment in  $[q]^w$ , *e.g.*  $(x_1 = 1) \wedge (x_2 = 7) \wedge \dots \wedge (x_w = q - 3)$ . An instance of the MAX- $w$ -CONJSAT $_q$  problem is one where the set of constraints  $\mathcal{C}$  is the set of all  $w$ -terms. We now use the following lemma from [Tre98] that allows us to reduce a MAX- $w$ -CSP $_q$  instance to a MAX- $w$ -CONJSAT $_q$  instance.

**Lemma 8.4.1** ([Tre98]). *Given an instance  $\mathcal{W}_1$  of MAX- $w$ -CSP $_q$ , we can find a instance  $\mathcal{W}_2$  of MAX- $w$ -CONJ-SAT $_q$  on the same set of variables, and a constant  $\beta > 0$  such that for every assignment  $f$ ,  $\text{val}(f, \mathcal{W}_2) = \beta \cdot \text{val}(f, \mathcal{W}_1)$ .*

*Proof.* Given an instance  $\mathcal{W}_1$  of MAX- $w$ -CSP $_q$ , consider a constraint  $C \in \mathcal{C}$  with weight  $\mathcal{W}_1(C)$ . We can assume without loss of generality that the arity of  $C$  is exactly  $w$ , and it depends on variables  $x_1, \dots, x_w$ . For each assignment in  $[q]^k$  that satisfies  $C$ , we create a  $w$ -CONJ-SAT $_q$  clause that is satisfied only for that assignment, and give it weight  $\mathcal{W}_1(C)$ . *e.g.* If  $C$  was satisfied by  $x_1 = \dots = x_w = 2$ , we create the clause  $(x_1 = 2) \wedge \dots \wedge (x_w = 2)$  with weight  $\mathcal{W}_1(C)$ . It is easy to see that for every assignment to  $x_1, \dots, x_n$ , the weight of constraints satisfied in the new instance is the same as the

weight of the constraints satisfied in the  $\text{MAX-}w\text{-CONJ-SAT}_q$  instance created. Define  $\beta$  to be the sum of weights of all the constraints in the new instance, then  $\mathcal{W}_2$  is obtained by multiplying the weight of all the constraints in the new instance by  $1/\beta$  (to make sure they sum up to 1). ■

Note that the scaling factor  $\beta$  in the lemma above is immaterial since we will give an algorithm with Pareto approximation guarantee.

We say  $(v, i) \in C$  if  $v \in C$  and  $v = i$  is in the satisfying assignment of  $C_f$ . By abuse of notation, we say for a set of variables  $T$ ,  $T \subseteq C$  if for all  $v \in T$ , there exists  $i \in [q]$ , such that  $(v, i) \in C$ .

#### 8.4.2 Random Assignments

In this section, we state and prove a lemma that gives a sufficient condition for the value of a  $\text{MAX-}w\text{-CONJSAT}_q$  to be highly concentrated under independent random assignments to the variables. Let  $\text{Dist}(q)$  denote the set of all probability distributions on the set  $[q]$ . For a distribution  $p \in \text{Dist}(q)$  and  $i \in q$ , we use  $p_i$  to denote the probability  $i$  in the distribution  $p$ . Let  $\rho$  be a partial assignment. Let  $p : V \setminus S_\rho \rightarrow \text{Dist}(q)$  be such that  $p(v)_i \geq \frac{1}{qw}$  for all  $v \in V \setminus S_\rho$  and all  $i \in [q]$ . Let  $g : V \setminus S_\rho \rightarrow [q]$  be a random assignment obtained by sampling  $g(v)$  for each  $v$  independently according to the distribution  $p(v)$ .

Define the random variable

$$Y \stackrel{\text{def}}{=} \text{val}(\rho \cup g, \mathcal{W}) - \text{val}(\rho, \mathcal{W}) = \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C) C(\rho \cup g).$$

The random variable  $Y$  measures the contribution of active constraints to  $\text{val}(\rho \cup g, \mathcal{W})$ . Note that the two quantities  $\mathbf{E}[Y]$  and  $\text{Var}[Y]$  can be computed efficiently given  $p$ . We denote these by  $\text{TrueMean}_\rho(p, \mathcal{W})$  and  $\text{TrueVar}_\rho(p, \mathcal{W})$ . The following lemma is a generalization of Lemma 8.3.1.

**Lemma 8.4.2.** *Let  $p, g, Y$  be as above.*

1. *If  $\text{TrueVar}_\rho(p, \mathcal{W}) < \delta_0 \varepsilon_0^2 \cdot \text{TrueMean}_\rho(p, \mathcal{W})^2$  then  $\Pr[Y < (1 - \varepsilon_0) \mathbf{E}[Y]] < \delta_0$ .*

2. If  $\text{TrueVar}_\rho(p, \mathcal{W}) \geq \delta_0 \varepsilon_0^2 \cdot \text{TrueMean}_\rho(p, \mathcal{W})^2$ , then there exists  $v \in V \setminus S_\rho$  such that

$$\text{activedegree}_\rho(v, \mathcal{W}) \geq \frac{\varepsilon_0^2 \delta_0}{w^2 (qw)^w} \cdot \text{activedegree}_\rho(\mathcal{W}).$$

*Proof.* Item 1 of the lemma follows immediately from Chebyshev's inequality. We now prove Item 2. First note that for every active constraint  $C$  given  $\rho$ ,  $\mathbf{E}[C(\rho \cup g)] \geq \frac{1}{(qw)^w}$  (this follows from our hypothesis that  $p(v)_i \geq \frac{1}{qw}$  for each  $v \in V \setminus S_\rho$  and each  $i \in [q]$ ).

We first bound  $\text{TrueMean}_\rho(p, \mathcal{W})$  and  $\text{TrueVar}_\rho(p, \mathcal{W})$  in terms of the weights of active constraints:

$$\begin{aligned} \text{TrueMean}_\rho(p, \mathcal{W}) &= \mathbf{E}[Y] = \mathbf{E} \left[ \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C) \cdot C(\rho \cup g) \right] \\ &= \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C) \cdot \mathbf{E}[C(\rho \cup g)] \geq \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C) \cdot \frac{1}{(qw)^w} \\ &= \frac{1}{(qw)^w} \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C) \end{aligned}$$

$$\begin{aligned} \text{TrueVar}_\rho(p, \mathcal{W}) &= \text{Var}[Y] = \text{Var} \left[ \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C) \cdot C(\rho \cup g) \right] \\ &= \sum_{C_1, C_2 \in \text{Active}(\rho)} \mathcal{W}(C_1) \mathcal{W}(C_2) \cdot \begin{pmatrix} \mathbf{E}[C_1(\rho \cup g) C_2(\rho \cup g)] \\ -\mathbf{E}[C_1(\rho \cup g)] \mathbf{E}[C_2(\rho \cup g)] \end{pmatrix} \\ &\leq \sum_{C_1 \sim_\rho C_2} \mathcal{W}(C_1) \mathcal{W}(C_2) \cdot \mathbf{E}[C_1(\rho \cup g)] \\ &= \sum_{C_1 \in \text{Active}(\rho)} \mathcal{W}(C_1) \mathbf{E}[C_1(\rho \cup g)] \cdot \sum_{C_2 \sim_\rho C_1} \mathcal{W}(C_2) \\ &\leq \sum_{C_1 \in \text{Active}(\rho)} \mathcal{W}(C_1) \mathbf{E}[C_1(\rho \cup g)] \cdot \max_{C \in \text{Active}(\rho)} \sum_{C_2 \sim_\rho C} \mathcal{W}(C_2) \\ &= \text{TrueMean}_\rho(p, \mathcal{W}) \cdot \max_{C \in \text{Active}(\rho)} \sum_{C_2 \sim_\rho C} \mathcal{W}(C_2). \end{aligned}$$

Hence, if the condition in case 2 is true then it follows that,

$$\max_{C \in \text{Active}(\rho)} \sum_{C_2 \sim_\rho C} \mathcal{W}(C_2) \geq \frac{\text{TrueVar}_\rho(p, \mathcal{W})}{\text{TrueMean}_\rho(p, \mathcal{W})} \geq \frac{\delta_0 \varepsilon_0^2}{(qw)^w} \cdot \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C).$$

We now relate these quantities to active degrees.

$$\begin{aligned}
\text{activedegree}_\rho(\mathcal{W}) &= \sum_{v \in V \setminus S_\rho} \text{activedegree}_\rho(v, \mathcal{W}) = \sum_{v \in V \setminus S_\rho} \sum_{C \in \text{Active}(\rho), C \ni v} \mathcal{W}_\ell(C) \\
&= \sum_{C \in \text{Active}(\rho)} \sum_{v \in C, v \in V \setminus S_\rho} \mathcal{W}_\ell(C) \leq \sum_{C \in \text{Active}(\rho)} w \cdot \mathcal{W}_\ell(C) \\
&= w \sum_{C \in \text{Active}(\rho)} \mathcal{W}_\ell(C)
\end{aligned}$$

This means that there is an active constraint  $C$ , such that

$$\sum_{C_2 \sim_\rho C} \mathcal{W}(C_2) \geq \frac{\delta_0 \varepsilon_0^2}{(qw)^w} \cdot \frac{1}{w} \text{activedegree}_\rho(\mathcal{W})$$

Since  $C$  is an active constraint and  $|C \cap V \setminus S_\rho| \leq w$ , there is some variable  $v \in C \cap V \setminus S_\rho$ , such that

$$\begin{aligned}
\text{activedegree}_\rho(v, \mathcal{W}) &= \sum_{C_2 \in \text{Active}(\rho), C_2 \ni v} \mathcal{W}(C_2) \geq \frac{1}{w} \sum_{C_2 \sim_\rho C} \mathcal{W}(C_2) \\
&\geq \frac{\varepsilon_0^2 \delta_0}{w^2 (qw)^w} \cdot \text{activedegree}_\rho(\mathcal{W}).
\end{aligned}$$

as required. ■

### 8.4.3 LP Relaxations

Our algorithm will use the Linear Programming relaxation for  $\text{MAX-}w\text{-CONJSAT}_q$  from the work of Trevisan [Tre98] (actually, a simple generalization to  $q$ -ary alphabets). The first LP,  $\text{ConjSAT-LP}_1(\rho)$ , described in Fig. 8.4, describes the set of all feasible solutions for the relaxation, consistent with the partial assignment  $\rho$ . Given a set of target values  $(c_\ell)_{\ell \in [k]}$ , the second LP,  $\text{ConjSAT-LP}_2(\rho)$  describes the set of feasible solutions to  $\text{ConjSAT-LP}_1(\rho)$  that achieve the required objective values.

For  $\vec{t}, \vec{z}$  satisfying linear constraints  $\text{ConjSAT-LP}_1(\rho)$ , let  $\text{smooth}(\vec{t})$  denote the map  $p : V \setminus S_\rho \rightarrow \text{Dist}(q)$  with  $p(v)_i = \frac{w-1}{qw} + \frac{t_{v,i}}{w}$ . The following theorem from [Tre98] provides an algorithm to round this feasible solution to obtain a good integral assignment.

**Lemma 8.4.3.** *Let  $\rho$  be a partial assignment.*

1. **Relaxation:** For every  $g_0 : V \setminus S_\rho \rightarrow [q]$ , there exist  $\vec{t}, \vec{z}$  satisfying  $\text{ConjSAT-LP}_1(\rho)$  such that for every  $\text{MAX-}w\text{-CONJSAT}_q$  instance  $\mathcal{W}$ :

$$\sum_{C \in \mathcal{C}} \mathcal{W}(C) z_C = \text{val}(g_0 \cup \rho, W).$$

2. **Rounding:** Suppose  $\vec{t}, \vec{z}$  satisfy  $\text{ConjSAT-LP}_1(\rho)$ . Then for every  $\text{MAX-}w\text{-CONJSAT}_q$  instance  $\mathcal{W}$ :

$$\text{val}(\rho, \mathcal{W}) + \text{TrueMean}_\rho(\text{smooth}(\vec{t}), \mathcal{W}) \geq \frac{1}{q^{w-1}} \cdot \sum_{C \in \mathcal{C}} z_C \mathcal{W}(C).$$

*Proof.* We begin with the first part. For  $v \in S_\rho$ ,  $i \in [q]$ , define  $t_{v,i} = 1$  if  $\rho(v) = i$ , and define  $t_{v,i} = 0$  otherwise. For  $v \in V \setminus S_\rho$ ,  $i \in [q]$ , define  $t_{v,i} = 1$  if  $g_0(v) = i$ , and define  $t_{v,i} = 0$  otherwise. For  $C \in \mathcal{C}$ , define  $z_C = 1$  if  $C(g_0 \cup \rho) = 1$ , and define  $z_C = 0$  otherwise. It is easy to see that these  $\vec{t}, \vec{z}$  satisfies  $\text{ConjSAT-LP}_1(\rho)$ , and that for every instance  $\mathcal{W}$ :

$$\sum_{C \in \mathcal{C}} \mathcal{W}(C) z_C = \text{val}(g_0 \cup \rho, W).$$

Now we consider the second part. Let  $\mathcal{W}$  be any instance of  $\text{MAX-}w\text{-CONJSAT}_q$ . Let  $p = \text{smooth}(t)$ . Let  $g : V \setminus S_\rho \rightarrow [q]$  be sampled as follows: independently for each  $v \in V \setminus S_\rho$ ,  $g(v)$  is sampled from the distribution  $p(v)$ . We need to show that:

$$\sum_{C \notin \text{Active}(\rho)} \mathcal{W}(C) C(\rho) + \mathbf{E} \left[ \sum_{C \in \text{Active}(\rho)} \mathcal{W}(C) C(\rho \cup g) \right] \geq \frac{1}{q^{w-1}} \cdot \sum_{C \in \mathcal{C}} z_C \mathcal{W}(C).$$

It is easy to check that for  $C \notin \text{Active}(\rho)$ ,  $z_C > 0$  only if  $C(\rho) = 1$ , and thus  $\sum_{C \notin \text{Active}(\rho)} C(\rho) \mathcal{W}(C) \geq \sum_{C \notin \text{Active}(\rho)} z_C \mathcal{W}(C)$ . For  $C \in \text{Active}(\rho)$ , we have the following claim:

**Claim 8.4.4.** For  $C \in \text{Active}(\rho)$ ,  $\mathbf{E}[C(\rho \cup g)] \geq \frac{z_C}{q^{w-1}}$ .

*Proof.* Suppose there are exactly  $h$  variables in  $C$  which are not in  $S_\rho$ . Let these variables be  $(v_i)_{i=1}^h$ . Let  $(v_i, a_i)_{i=1}^h$  be the assignment to these variables that makes  $C$

satisfied.

$$\begin{aligned}
\mathbf{E}[C(\rho \cup g)] &= \Pr[C \text{ is satisfied by } \rho \cup g] \geq \prod_{i=1}^h \left( \frac{w-1}{qw} + \frac{t_{v_i, a_i}}{w} \right) \\
&\geq \prod_{i=1}^h \left( \frac{w-1}{qw} + \frac{z_C}{w} \right) = \left( \frac{w-1}{qw} + \frac{z_C}{w} \right)^h \\
&= \left( \frac{w-1}{qw} + \frac{z_C}{w} \right)^w \geq \frac{z_C}{q^{w-1}}
\end{aligned}$$

Here the last inequality follows from the observation that the minimum of the function  $\frac{\left(\frac{w-1}{qw} + \frac{z}{w}\right)^w}{z}$  as  $z$  varies in  $[0, 1]$ , is attained at  $z = 1/q$ . ■

$$\begin{aligned}
z_C &\leq t_{v,i} && \forall C \in \mathcal{C}, \forall (v, i) \in C \\
1 \geq t_{v,i} &\geq 0 && \forall v \in V \setminus S_\rho, i \in [q] \\
\sum_{i=1}^q t_{v,i} &= 1 && \forall v \in V \\
t_{v,i} &= 1 && \forall v \in S_\rho \text{ and } i \in [q], \\
&&& \text{such that } h_\rho(v) = i
\end{aligned}$$

Figure 8.4: Linear inequalities  $\text{ConjSAT-LP}_1(\rho)$

$$\begin{aligned}
\sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) \cdot z_C &\geq c_\ell \quad \forall \ell \in [k] \\
\vec{t}, \vec{z} &\text{ satisfy } \text{ConjSAT-LP}_1(\rho).
\end{aligned}$$

Figure 8.5: Linear inequalities  $\text{ConjSAT-LP}_2(\rho)$

This completes the proof of the Lemma. ■

#### 8.4.4 The Algorithm

We now give our Pareto approximation algorithm for MAX- $w$ -CSP $_q$  in Fig. 8.7 (which uses the procedure from Fig. 8.6).

**Input:** A tree node  $\nu$  and an instance  $\mathcal{W}_\ell$ .

**Output:** A tuple of variables of size at most  $w$ .

1. Let  $v_1 \in V \setminus S_{\rho_\nu}$  be a variable which maximizes the value of  $\text{activedegree}_{\rho_\nu}(v_1, \mathcal{W}_\ell)$ . Set  $D \leftarrow \{v_1\}$ .
2. While  $|D| \leq w$ , do the following

- (a) If there is a variable  $v$  in  $V \setminus S_{\rho_\nu}$  such that

$$\text{activedegree}_{\rho_\nu}(D \cup v, \mathcal{W}_\ell) \geq \frac{\text{activedegree}_{\rho_\nu}(D, \mathcal{W}_\ell)}{(4qwt_k)^w},$$

set  $D \leftarrow D \cup v$ .

- (b) Otherwise, go to Step 3.

3. Return  $D$  as a tuple (in arbitrary order, with  $v_1$  as the first element).

Figure 8.6: TUPLESELECTION for MAX- $w$ -CONJSAT $_q$

**Input:**  $k$  instances of MAX- $w$ -CONJSAT $_q$   $\mathcal{W}_1, \dots, \mathcal{W}_k$  on the variable set  $V$ ,  $\varepsilon > 0$  and target objective values  $c_1, \dots, c_k$ .

**Output:** An assignment to  $V$

**Parameters:**  $\delta_0 = \frac{1}{10(k+1)}$ ,  $\varepsilon_0 = \varepsilon$ ,  $\gamma = \frac{\varepsilon_0^2 \delta_0}{w^2 (qw)^w}$ ,  $t = \left\lceil \frac{20w^2 k^2}{\gamma} \cdot \log \left( \frac{10k}{\gamma} \right) \right\rceil$

1. Initialize tree  $T$  to be an empty  $q^w$ -ary tree (i.e., just 1 root node and each node has at most  $q^w$  children).
2. We will associate with each node  $\nu$  of the tree:
  - (a) A partial assignment  $\rho_\nu$ .
  - (b) A special set of variables  $\mathcal{T}_\nu \subseteq V \setminus S_{\rho_\nu}$ .
  - (c) A special instance  $\mathcal{I}_\nu \in [k]$ .
  - (d) A collection of integers  $\text{count}_{\nu,1}, \dots, \text{count}_{\nu,k}$ .
  - (e) A trit representing whether the node  $\nu$  is living, exhausted or dead.
3. Initialize the root node  $\nu_0$  to (1)  $\rho_{\nu_0} = (\emptyset, \emptyset)$ , (2) have all  $\text{count}_{\nu_0,\ell} = 0$ , (3) living.
4. While there is a living leaf  $\nu$  of  $T$ , do the following:
  - (a) Check if the LP  $\text{ConjSAT-LP}_2(\rho_\nu)$  has a feasible solution.
    - i. If  $\vec{t}, \vec{z}$  is a feasible solution, then define  $p_\nu : V \setminus S_{\rho_\nu} \rightarrow \text{Dist}(q)$  by  $p = \text{smooth}(\vec{t})$ .
    - ii. If not, then declare  $\nu$  to be dead and return to Step 4.
  - (b) For each  $\ell \in [k]$ , compute  $\text{TrueVar}_{\rho_\nu}(p, \mathcal{W}_\ell)$ ,  $\text{TrueMean}_{\rho_\nu}(p, \mathcal{W}_\ell)$ .
  - (c) If  $\text{TrueVar}_{\rho_\nu}(p, \mathcal{W}_\ell) \geq \delta_0 \varepsilon_0^2 \text{TrueMean}_{\rho_\nu}(p, \mathcal{W}_\ell)^2$ , then set  $\text{flag}_\ell = \text{TRUE}$ , else set  $\text{flag}_\ell = \text{FALSE}$ .

- . (d) Choose the smallest  $\ell \in [k]$ , such that  $\text{count}_\ell < t$  AND  $\text{flag}_\ell = \text{TRUE}$  (if any):
    - i. Set  $\mathcal{T}_\nu \leftarrow \text{TUPLESELECTION}(\nu, \mathcal{W}_\ell)$ . Set  $\mathcal{I}_\nu = \ell$ .
    - ii. Create  $q^{w'}$  children of  $\nu$ , with labels  $\nu b$  for each  $b \in [q]^{w'}$  and define
      - $\rho_{\nu b} = (S_{\rho_\nu} \cup \mathcal{T}_\nu, h^b)$ , where  $h^b$  extends  $h_{\rho_\nu}$  by  $h^b(\mathcal{T}_\nu^i) = b(i)$ .
      - For each  $\ell' \in [k]$  with  $\ell' \neq \ell$ , initialize  $\text{count}_{\nu b, \ell'} = \text{count}_{\nu, \ell'}$ . Initialize  $\text{count}_{\nu b, \ell} = \text{count}_{\nu, \ell} + 1$ .
      - Set  $\nu b$  to be living.
  - (e) If no such  $\ell$  exists, declare  $\nu$  to be exhausted.
5. Now every leaf of  $T$  is either exhausted or dead. For each exhausted leaf  $\nu$  of  $T$ :
- (a) Sample  $g_\nu : V \setminus S_{\rho_\nu} \rightarrow [q]$  by independently sampling  $g_\nu(v)$  from the distribution  $p_\nu(v)$ .
  - (b) For every assignment  $h : S_{\rho_\nu} \rightarrow [q]$ , compute  $\text{out}_{h, g_\nu} \leftarrow \min_{\ell \in [k]} \frac{\text{val}(h \cup g_\nu, \mathcal{W}_\ell)}{c_\ell}$ . If  $c_\ell = 0$  for some  $\ell \in [k]$ , we interpret  $\frac{\text{val}(h \cup g_\nu, \mathcal{W}_\ell)}{c_\ell}$  as  $+\infty$ .
6. Output the largest  $\text{out}_{h, g_\nu}$  seen, and the assignment  $h \cup g_\nu$  that produced it.

Figure 8.7: Algorithm SIM-MAXCONJSAT for approximating weighted simultaneous MAX- $w$ -CONJSAT $_q$

### 8.4.5 Analysis

Notice that the depth of the tree  $T$  is at most  $kt$ , and that for every  $\nu$ , we have that  $|S_{\rho_\nu}| \leq wkt$ . This implies that the running time is at most  $q^{O(wkt)} \cdot \text{poly}(n)$ .

Let  $f^* : V \rightarrow [q]$  be an assignment such that  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for each  $\ell \in [k]$ . Let  $\nu^*$  be the unique leaf of the tree  $T$  for which  $\rho_{\nu^*}$  is consistent with  $f^*$ . (This  $\nu^*$  can be found as follows: start with  $\nu$  equal to the root. Set  $\nu$  to equal the unique child of  $\nu$  for which  $\rho_\nu$  is consistent with  $f^*$ , and repeat until  $\nu$  becomes a leaf. This leaf is  $\nu^*$ ). Observe that since  $f^*$  is an assignment such that  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for every  $\ell \in [k]$ , by picking  $g_0 = f^*|_{V \setminus S^*}$  in part 1 of Lemma 8.4.3, we know that  $\text{ConjSAT-LP}_2(\rho^*)$  is feasible, and hence  $\nu^*$  must be an exhausted leaf (and not dead).

Define  $\rho^* = \rho_{\nu^*}$ ,  $S^* = S_{\rho^*}$ ,  $h^* = h_{\rho^*}$  and  $p^* = p_{\nu^*}$ .

At the completion of Step 4, if  $\ell \in [k]$  satisfies  $\text{count}_{\nu^*, \ell} = t$ , we call instance  $\ell$  a *high variance* instance. Otherwise we call instance  $\ell$  a *low variance* instance.

#### Low Variance Instances.

First we show that for the leaf  $\nu^*$  in Step 8.5, combining the partial assignment  $h^*$  with a random assignment  $g_{\nu^*}$  in Step 0a is good for any low variance instances with high probability.

**Lemma 8.4.5.** *Let  $\ell \in [k]$  be any low variance instance. For the leaf node  $\nu^*$ , let  $g_{\nu^*}$  be the random assignment sampled in Step 0a of SIM-MAXCONJSAT. Then with probability at least  $1 - \delta_0$ , the assignment  $f = h^* \cup g_{\nu^*}$  satisfies:*

$$\Pr_{g_{\nu^*}} [\text{val}(f, \mathcal{W}_\ell) \geq (1/q^{w-1} - \varepsilon/2) \cdot c_\ell] \geq 1 - \delta_0.$$

*Proof.* For every low variance instance  $\ell$ , we have that  $\text{TrueVar}_{\rho_{\nu^*}}(p^*, \mathcal{W}_\ell) < \delta_0 \varepsilon_0^2 \cdot \text{TrueMean}_{\rho_{\nu^*}}(p^*, \mathcal{W}_\ell)^2$ . Define  $Y \stackrel{\text{def}}{=} \text{val}(\rho^* \cup g_{\nu^*}, \mathcal{W}_\ell) - \text{val}(\rho^*, \mathcal{W}_\ell)$ . By Lemma 8.4.2, we

have  $\Pr[Y < (1 - \varepsilon_0) \mathbf{E}[Y]] < \delta_0$ . Thus, with probability at least  $1 - \delta_0$ , we have,

$$\begin{aligned}
\text{val}(f, \mathcal{W}_\ell) &\geq \text{val}(\rho^\star, \mathcal{W}_\ell) + (1 - \varepsilon_0) \mathbf{E}[Y] \\
&= \text{val}(\rho^\star, \mathcal{W}_\ell) + (1 - \varepsilon_0) \cdot \text{TrueMean}_{\rho_{\nu^\star}}(\text{smooth}(\vec{t}), \mathcal{W}_\ell) \\
&= (1 - \varepsilon_0) \cdot (\text{val}(\rho^\star, \mathcal{W}_\ell) + \text{TrueMean}_{\rho_{\nu^\star}}(\text{smooth}(\vec{t}), \mathcal{W}_\ell)) \\
&\geq \frac{1}{q^{w-1}} \cdot (1 - \varepsilon_0) \cdot \sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) \cdot z_C \geq \frac{1}{q^{w-1}} \cdot (1 - \varepsilon_0) \cdot c_\ell \geq \left( \frac{1}{q^{w-1}} - \frac{\varepsilon}{2} \right) \cdot c_\ell,
\end{aligned}$$

where we have used the second part of Lemma 8.4.3.  $\blacksquare$

Next, we will consider a small perturbation of  $h^\star$  which will ensure that the algorithm performs well on high variance instances too. We will ensure that this perturbation does not affect the success on the low variance instances.

### High Variance Instances.

Fix a high variance instance  $\ell$ . Let  $\nu$  be an ancestor of  $\nu^\star$  with  $\mathcal{I}_\nu = \ell$ . Let  $\mathcal{T}_\nu^1$  denote the first element of the tuple  $\mathcal{T}_\nu$ . Define:

$$\begin{aligned}
\text{activedegree}_\nu &\stackrel{\text{def}}{=} \text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu^1, \mathcal{W}_\ell). \\
\text{activedegree}_{\mathcal{T}_\nu} &\stackrel{\text{def}}{=} \text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu, \mathcal{W}_\ell).
\end{aligned}$$

**Observation 8.4.1.** *For any node  $\nu$ , in the tree,*

$$\text{activedegree}_{\mathcal{T}_\nu} \geq \frac{\text{activedegree}_\nu}{(4qwt_k)^{w \cdot (|\mathcal{T}_\nu| - 1)}}.$$

*Proof.* For  $\nu$  such that  $|\mathcal{T}_\nu| = 1$ , we have, by definition,  $\text{activedegree}_{\mathcal{T}_\nu} = \text{activedegree}_\nu$  and the inequality follows. The lower bound is obvious from the TUPLE SELECTION procedure if  $|\mathcal{T}_\nu| > 1$ .  $\blacksquare$

Let  $\mathcal{C}_\nu$  be the set of all constraints  $C$  containing all variables in  $\mathcal{T}_\nu$  which are active given  $\rho_\nu$ .

We call a constraint  $C$  in  $\mathcal{C}_\nu$  a **backward** constraint if  $C$  only involves variables from  $S_{\rho_\nu} \cup \mathcal{T}_\nu$ . Otherwise we call  $C$  in  $\mathcal{C}_\nu$  a **forward** constraint. Let  $\mathcal{C}_\nu^{\text{backward}}$  and  $\mathcal{C}_\nu^{\text{forward}}$

denote the sets of these constraints. Finally, let  $\mathcal{C}_\nu^{\text{out}}$  denote the set of all constraints from  $\mathcal{C}_\nu$  that involve at least *one* variable from  $V \setminus S^*$  and none from  $S^* \setminus S_{\rho_\nu}$ .

Define backward degree and forward degree as follows:

$$\begin{aligned} \text{backward}_\nu &\stackrel{\text{def}}{=} \sum_{C \in \mathcal{C}_\nu^{\text{backward}}} \mathcal{W}_\ell(C), \\ \text{forward}_\nu &\stackrel{\text{def}}{=} \sum_{C \in \mathcal{C}_\nu^{\text{forward}}} \mathcal{W}_\ell(C). \end{aligned}$$

Note that:

$$\text{activedegree}_{\mathcal{T}_\nu} = \text{backward}_\nu + \text{forward}_\nu.$$

Based on the above definitions, we classify  $\nu$  into one of three categories:

1. If  $\text{backward}_\nu \geq \frac{1}{2} \cdot \text{activedegree}_{\mathcal{T}_\nu}$ , then we call  $\nu$  typeAB.
2. Otherwise, we call  $\nu$  typeC.

We have the following lemma about typeC nodes.

**Lemma 8.4.6.** *For every typeC node  $\nu$ , we have*

1. *For every  $v \in V \setminus (S_{\rho_\nu} \cup \mathcal{T}_\nu)$ ,  $\text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu \cup \{v\}, \mathcal{W}_\ell) \leq \frac{\text{activedegree}_{\mathcal{T}_\nu}}{(4qwt k)^w}$ .*
2.  $\sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \geq \frac{1}{4} \cdot \text{activedegree}_{\mathcal{T}_\nu}$ .

*Proof.* If  $\nu$  is a typeC node, we must have that for every  $v \in V \setminus (S_{\rho_\nu} \cup \mathcal{T}_\nu)$ ,

$$\text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu \cup \{v\}, \mathcal{W}_\ell) < \frac{\text{activedegree}_{\mathcal{T}_\nu}}{(4qwt k)^w}.$$

This follows from the description of the TUPLESELECTION procedure, and the observation that  $\text{activedegree}_\nu(T, \mathcal{W}_\ell) = 0$  for any  $T \subset V$  with  $|T| > w$ .

In particular, since  $|S^*| \leq wtk$ , the total weight of constraints containing  $\mathcal{T}_\nu$  and some variable in  $S^* \setminus (S_{\rho_\nu} \cup \mathcal{T}_\nu)$  is at most

$$\begin{aligned} \sum_{v \in S^* \setminus (S_{\rho_\nu} \cup \mathcal{T}_\nu)} \text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu \cup \{v\}, \mathcal{W}_\ell) &\leq \sum_{v \in S^* \setminus (S_{\rho_\nu} \cup \mathcal{T}_\nu)} \frac{\text{activedegree}_{\mathcal{T}_\nu}}{(4qwt k)^w} \\ &\leq |S^* \setminus (S_{\rho_\nu} \cup \mathcal{T}_\nu)| \cdot \frac{\text{activedegree}_{\mathcal{T}_\nu}}{(4qwt k)^w} \\ &\leq wtk \cdot \frac{\text{activedegree}_{\mathcal{T}_\nu}}{(4qwt k)^w} \leq \frac{1}{4} \cdot \text{activedegree}_{\mathcal{T}_\nu}. \end{aligned}$$

Thus, we get,

$$\begin{aligned} \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) &= \text{forward}_\nu - \left\{ \begin{array}{l} \text{total weight of constraints containing} \\ \mathcal{T}_\nu \text{ and some variable in } S^* \setminus (S_{\rho_\nu} \cup \mathcal{T}_\nu) \end{array} \right\} \\ &\geq \frac{1}{2} \cdot \text{activedegree}_{\mathcal{T}_\nu} - \frac{1}{4} \cdot \text{activedegree}_{\mathcal{T}_\nu} = \frac{1}{4} \cdot \text{activedegree}_{\mathcal{T}_\nu}. \end{aligned}$$

This completes the proof of second statement.  $\blacksquare$

For a partial assignment  $g : V \setminus S^* \rightarrow [q]$ , we say that  $g$  is **Cgood** for  $\nu$  if there exists a setting of variables in  $\mathcal{T}_\nu$  that satisfies at least  $\frac{1}{8 \cdot (qw)^w} \cdot \text{activedegree}_{\mathcal{T}_\nu}$  weight amongst constraints in  $\mathcal{C}_\nu^{\text{out}}$ .

The next lemma allows us to prove that that for every node  $\nu$  of **typeC**, with high probability, the random assignment  $g_{\nu^*} : V \setminus S^* \rightarrow [q]$ , is **Cgood** for  $\nu$ .

**Lemma 8.4.7.** *Let  $\nu$  be **typeC**. Suppose  $g : V \setminus S^* \rightarrow [q]$  is a random assignment obtained by independently sampling  $g(v)$  for each  $v \in V \setminus S^*$  from a distribution such that distribution  $\Pr[g(v) = i] \geq \frac{1}{qw}$  for each  $i \in [q]$ . Then:*

$$\Pr_g[g \text{ is Cgood for } \nu] \geq 1 - 2 \cdot e^{-tk/8qw}.$$

*Proof.* Let  $\ell = \mathcal{I}_\nu$ .

Consider a constraint  $C \in \mathcal{C}_\nu^{\text{out}}$ . For partial assignments  $b : \mathcal{T}_\nu \rightarrow [q]$  and  $g : V \setminus S^* \rightarrow [q]$ , define  $C(\rho_\nu \cup b \cup g) \in \{0, 1\}$  to be 1 iff  $C$  is satisfied by  $\rho_\nu \cup b \cup g$ . Since  $C$  only contains variables from  $S_{\rho_\nu} \cup \mathcal{T}_\nu \cup (V \setminus S^*)$ , we have that  $C(\rho_\nu \cup b \cup g)$  is well defined.

Define  $\text{score}^b : [q]^{V \setminus S^*} \rightarrow \mathbb{R}$  by

$$\text{score}^b(g) = \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \cdot C(\rho_\nu \cup b \cup g).$$

In words,  $\text{score}^{(b)}(g)$  is the total weight of constraints in  $\mathcal{C}_\nu^{\text{out}}$  satisfied by setting  $S_{\rho_\nu}$  according to  $\rho_\nu$ , setting  $\mathcal{T}_\nu$  to  $b$ , and setting  $V \setminus S^*$  according to  $g$ .

Note that for all  $C \in \mathcal{C}_\nu^{\text{out}}$ ,  $\mathbf{E}_g[\sum_{b: \mathcal{T}_\nu \rightarrow [q]} C(\rho_\nu \cup b \cup g)] \geq \frac{1}{(qw)^{w - |\mathcal{T}_\nu|}}$ . This follows since  $C$  is an active constraint given  $\rho_\nu$ , and involves all variables from  $\mathcal{T}_\nu$ ; hence there exists an assignment  $b$  to  $\mathcal{T}_\nu$  and an assignment for at most  $w - |\mathcal{T}_\nu|$  variables from constraint  $C$  in  $V \setminus S^*$  such that  $C$  is satisfied. Since,  $g$  is a **smooth** distribution, this particular

assignment to  $w - |\mathcal{T}_\nu|$  in  $V \setminus S^*$  is sampled with probability at least  $\frac{1}{(qw)^{w-|\mathcal{T}_\nu|}}$ . Hence, for this particular choice of  $b$ ,  $C$  is satisfied with probability at least  $\frac{1}{(qw)^{w-|\mathcal{T}_\nu|}}$ . Thus:

$$\begin{aligned} \sum_{b: \mathcal{T}_\nu \rightarrow [q]} \mathbf{E}_g [\text{score}^b(g)] &= \sum_{b: \mathcal{T}_\nu \rightarrow [q]} \mathbf{E}_g \left[ \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \cdot C(\rho_\nu \cup b \cup g) \right] \\ &= \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \cdot \mathbf{E}_g \left[ \sum_{b: \mathcal{T}_\nu \rightarrow [q]} C(\rho_\nu \cup b \cup g) \right] \\ &\geq \frac{1}{(qw)^{w-|\mathcal{T}_\nu|}} \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C). \end{aligned}$$

Thus there exists  $b: \mathcal{T}_\nu \rightarrow [q]$  such that

$$\mathbf{E}_g [\text{score}^b(g)] \geq \frac{1}{q^{|\mathcal{T}_\nu|}} \cdot \frac{1}{(qw)^{w-|\mathcal{T}_\nu|}} \sum_{C \in \mathcal{C}_\nu^{\text{out}}} \mathcal{W}_\ell(C) \geq \frac{1}{4} \cdot \frac{1}{(qw)^w} \cdot \text{activedegree}_{\mathcal{T}_\nu},$$

where the last inequality follows by Lemma 8.4.6.

Fix this particular  $b$  for which the above inequality holds. We are going to use McDiarmid's inequality to show the concentration of  $\text{score}^b(g)$  around its mean. Since  $\nu$  is **typeC**, from Lemma 8.4.6, we know that for every vertex  $v \in V \setminus S^*$ , changing  $g$  on just  $v$  can change the value of  $\text{score}^b(g)$  by at most  $c_v \stackrel{\text{def}}{=} \text{activedegree}_{\rho_\nu}(\mathcal{T}_\nu \cup \{v\}, \mathcal{W}_\ell) \leq \frac{\text{activedegree}_{\mathcal{T}_\nu}}{(4qwt_k)^w}$ . Thus by McDiarmid's inequality (Lemma 8.8.5),

$$\begin{aligned} \Pr_g [g \text{ is not Cgood for } \nu] &\leq \Pr_g \left[ \text{score}^b(g) < \frac{1}{8 \cdot (qw)^w} \cdot \text{activedegree}_{\mathcal{T}_\nu} \right] \\ &\leq \Pr_g \left[ |\text{score}^b(g) - \mathbf{E}_g [\text{score}^b(g)]| > \frac{1}{8 \cdot (qw)^w} \cdot \text{activedegree}_{\mathcal{T}_\nu} \right] \\ &\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_{\mathcal{T}_\nu}^2}{64(qw)^{2w} \cdot \sum_{v \in V \setminus S^*} c_v^2} \right) \\ &\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_{\mathcal{T}_\nu}^2}{64(qw)^{2w} \cdot (\max_v c_v) \cdot \sum_{v \in V \setminus S^*} c_v} \right) \\ &\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_{\mathcal{T}_\nu}^2}{64(qw)^{2w} \cdot (\max_v c_v) \cdot \text{activedegree}_{\mathcal{T}_\nu}} \right) \\ &\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_{\mathcal{T}_\nu}}{64(qw)^{2w} \cdot (\max_v c_v)} \right) \\ &\leq 2 \cdot \exp \left( \frac{-2 \cdot \text{activedegree}_{\mathcal{T}_\nu}}{64(qw)^{2w} \cdot \left( \frac{\text{activedegree}_{\mathcal{T}_\nu}}{(4qwt_k)^w} \right)} \right) \\ &= 2 \cdot \exp \left( \frac{-2 \cdot (4qwt_k)^w}{64 \cdot (qw)^{2w}} \right) \leq 2 \cdot \exp \left( \frac{-tk}{8qw} \right). \end{aligned}$$

■

For a high variance instance  $\ell$ , let  $\nu_1^\ell, \dots, \nu_t^\ell$  be the  $t$  nodes with  $\mathcal{I}_\nu = \ell$  which lie on the path from the root to  $\nu^\star$ , numbered in order of their appearance on the path from the root to  $\nu^\star$ . Set  $\text{finalwt}_\ell = \text{activedegree}_{\rho^\star}(\mathcal{W}_\ell)$ . This is the active degree left over in instance  $\ell$  after the restriction  $\rho^\star$ .

**Lemma 8.4.8.** *For every high variance instance  $\ell \in [k]$  and for each  $i \leq [t/2]$ ,*

$$\text{activedegree}_{\nu_i^\ell} \geq \gamma \cdot (1 - \gamma)^{-t/2} \cdot \text{finalwt}_\ell \geq 100 \cdot (qw)^w \cdot (4qwt k)^{w^2} \cdot \text{finalwt}_\ell.$$

We skip the proof of this lemma. The first inequality is identical to the second part of Lemma 8.5.6, and the second inequality follows from the choice of  $t$ .

### Putting Everything Together.

We now show that when  $\nu$  is taken to equal  $\nu^\star$  in Step 8.5, then with high probability over the choice of  $g_{\nu^\star}$  in Step 5(a) there is a setting of  $h$  in Step 5(b) such that  $\min_{\ell \in [k]} \text{val}(h \cup g_{\nu^\star}, \mathcal{W}_\ell) \geq (\frac{1}{q^{w-1}} - \varepsilon) \cdot c_\ell$ .

**Theorem 8.4.9.** *Suppose the algorithm SIM-MAXCONJSAT is given as inputs  $\varepsilon > 0$ ,  $k$  simultaneous weighted MAX- $w$ -CONJSAT $_q$  instances  $\mathcal{W}_1, \dots, \mathcal{W}_k$  on  $n$  variables, and target objective value  $c_1, \dots, c_k$  with the guarantee that there exists an assignment  $f^\star$  such that for each  $\ell \in [k]$ , we have  $\text{val}(f^\star, \mathcal{W}_\ell) \geq c_\ell$ . Then, the algorithm runs in  $2^{O(k^4/\varepsilon^2 \log(k/\varepsilon))} \cdot \text{poly}(n)$  time, and with probability at least 0.9, outputs an assignment  $f$  such that for each  $\ell \in [k]$ , we have,  $\text{val}(f, \mathcal{W}_\ell) \geq (\frac{1}{q^{w-1}} - \varepsilon) \cdot c_\ell$ .*

*Proof.* Consider the case when  $\nu$  is taken to equal  $\nu^\star$  in Step 8.5. By Lemma 8.4.5, with probability at least  $1 - k\delta_0$  over the random choices of  $g_{\nu^\star}$ , we have that for every low variance instance  $\ell \in [k]$ ,  $\text{val}(h^\star \cup g_{\nu^\star}, \mathcal{W}_\ell) \geq (\frac{1}{q^{w-1}} - \frac{\varepsilon}{2}) \cdot c_\ell$ . By Lemma 8.4.7 and a union bound, with probability at least  $1 - \frac{t}{2} \cdot k \cdot 2e^{-tk/8qw} \geq 1 - \delta_0$  over the choice of  $g_{\nu^\star}$ , for every high variance instance  $\ell$  and for every typeC node  $\nu_i^\ell$ ,  $i \in [t/2]$ , we have that  $g_{\nu^\star}$  is **Cgood** for  $\nu_i^\ell$ . Thus with probability at least  $1 - (k+1)\delta_0$ , both these events occur. Henceforth we assume that both these events occur in Step 5(a) of the algorithm.

Our next goal is to show that there exists a partial assignment  $h : S^* \rightarrow [q]$  such that

1. For every instance  $\ell \in [k]$ ,  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell)$
2. For every high variance instance  $\ell \in [k]$ ,  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot 10 \cdot \text{finalwt}_\ell$ .

Before giving a proof of the existence of such an  $h$ , we show that this completes the proof of the theorem. We claim that when the partial assignment  $h$  guaranteed above is considered in the Step 5(b) in the algorithm, we obtain an assignment with the required approximation guarantees.

For every low variance instance  $\ell \in [k]$ , since we started with  $\text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (\frac{1}{q^{w-1}} - \frac{\varepsilon}{2}) \cdot c_\ell$ , property 1 above implies that every low variance instance  $\text{val}(h \cup g_{\nu^*}) \geq (\frac{1}{q^{w-1}} - \varepsilon) \cdot c_\ell$ . For every high variance instance  $\ell \in [k]$ , since  $h^* = f^*|_S$ ,

$$\text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \text{val}(f^*, \mathcal{W}_\ell) - \text{activedegree}_{\rho^*}(\mathcal{W}_\ell) \geq c_\ell - \text{finalwt}_\ell.$$

Combining this with properties 1 and 2 above, we get,

$$\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \left(1 - \frac{\varepsilon}{2}\right) \cdot \max\{c_\ell - \text{finalwt}_\ell, 10 \cdot \text{finalwt}_\ell\} \geq \frac{10}{11} \left(1 - \frac{\varepsilon}{2}\right) \cdot c_\ell.$$

Thus, for all instances  $\ell \in [k]$ , we get  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq \left(\frac{1}{q^{w-1}} - \frac{\varepsilon}{2}\right) \cdot c_\ell$ .

Now, it remains to show the existence of such an  $h$  by giving a procedure for constructing  $h$  by perturbing  $h^*$  (Note that this procedure is only part of the analysis). For nodes  $\nu, \nu'$  in the tree, let us write  $\nu \prec \nu'$  if  $\nu$  is an ancestor of  $\nu'$ , and we also say that  $\nu'$  is “deeper” than  $\nu$ .

### Constructing $h$ .

1. Initialize  $H \subseteq [k]$  to be the set of high variance instances.
2. Let  $N_0 = \{\nu_i^\ell \mid \ell \in H, i \in [t/2]\}$ . Note that  $N$  is a chain in the tree (since all the elements of  $N$  are ancestors of  $\nu^*$ ). Since every  $\nu \in N$  is an ancestor of  $\nu^*$ , we have  $h_{\rho_\nu} = h^*|_{S_{\rho_\nu}}$ .

3. Initialize  $D = \emptyset$ ,  $N = N_0$ ,  $h = h^*$ .
4. During the procedure, we will be changing the assignment  $h$ , and removing elements from  $N$ . We will always maintain the following two invariants:

- $|N| > \frac{t}{4}$ .
- For every  $\nu \in N$ ,  $h|_{S_{\rho\nu}} = h^*|_{S_{\rho\nu}}$ .

5. While  $|D| \neq |H|$  do:

(a) Let

$$B = \left\{ v \in V \mid \exists \ell \in [k] \text{ with } \sum_{C \in \mathcal{C}, C \ni v} \mathcal{W}_\ell(C) \cdot C(h \cup g_{\nu^*}) \geq \frac{\varepsilon}{2wk} \text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \right\}.$$

Note that  $|B| \leq \frac{2w^2k^2}{\varepsilon} < \frac{t}{4}$ .

- (b) Let  $\nu \in N$  be the deepest element of  $N$  for which:  $\mathcal{T}_\nu \cap B = \emptyset$ .

Such a  $\nu$  exists because:

- $|N| > \frac{t}{4} > |B|$ , and
- there are at most  $|B|$  nodes  $\nu$  for which  $\mathcal{T}_\nu \cap B \neq \emptyset$  (since  $\mathcal{T}_\nu$  are all disjoint for distinct  $\nu$ ).

- (c) Let  $\ell \in H$  and  $i \in [t/2]$  be such that  $\nu = \nu_i^\ell$ . Let  $\rho = \rho_\nu$ . We will now modify the assignment  $h$  for variables in  $\mathcal{T}_\nu$  to guarantee that  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq 10 \cdot \text{finalwt}_\ell$ . The procedure depends on whether  $\nu$  is **typeAB** or **typeC**.

- i. If  $\nu$  is **typeAB**, then we know that  $\text{backward}_\nu \geq \frac{1}{2} \cdot \text{activedegree}_{\mathcal{T}_\nu}$ .

The second invariant tells us that  $\rho = h^*|_{S_\rho} = h|_{S_\rho}$ . Thus we have:

$$\begin{aligned} \text{backward}_\nu &= \sum_{C \in \mathcal{C}_\nu^{\text{backward}}} \mathcal{W}_\ell(C) \\ &= \sum_{C \subseteq S_\rho \cup \mathcal{T}_\nu, C \supseteq \mathcal{T}_\nu, C \in \text{Active}(\rho)} \mathcal{W}_\ell(C) \\ &= \sum_{C \subseteq S_\rho \cup \mathcal{T}_\nu, C \supseteq \mathcal{T}_\nu, C \in \text{Active}(h|_{S_\rho})} \mathcal{W}_\ell(C). \end{aligned}$$

This implies that we can modify the assignment  $h$  on the variables  $\mathcal{T}_\nu$  such that after the modification, the weights of satisfied backward constraints is:

$$\begin{aligned}
\sum_{\substack{C \subseteq S_\rho \cup \mathcal{T}_\nu, C \supseteq \mathcal{T}_\nu, \\ C \in \text{Active}(h|_{S_\rho})}} \mathcal{W}_\ell(C) C(h) &\geq \frac{1}{q^w} \sum_{\substack{C \subseteq S_\rho \cup \mathcal{T}_\nu, C \supseteq \mathcal{T}_\nu, \\ C \in \text{Active}(h|_{S_\rho})}} \mathcal{W}_\ell(C) \\
&= \frac{1}{q^w} \cdot \text{backward}_\nu \\
&\geq \frac{1}{2q^w} \cdot \text{activedegree}_{\mathcal{T}_\nu} \\
&\geq 10 \cdot \text{finalwt}_\ell.
\end{aligned}$$

where the  $\frac{1}{q^w}$  factor in the first inequality appears because there could be as many as  $q^w$  possible assignments to variables in  $\mathcal{T}_\nu$ , and the last inequality holds because of Observation 8.4.1 and Lemma 8.4.8. In particular, after making this change, we have  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq 10 \cdot \text{finalwt}_\ell$ .

- ii. If  $\nu$  is **typeC**, then we know that  $g$  is **Cgood** for  $\nu$ . Thus, by the definition of **Cgood**, we can choose a setting of  $\mathcal{T}_\nu$  so that at least a total of  $\frac{1}{8 \cdot (qw)^w} \cdot \text{activedegree}_{\mathcal{T}_\nu} \geq 10 \cdot \text{finalwt}_\ell$   $\mathcal{W}_\ell$ -weight constraints between  $\mathcal{T}_\nu$  and  $V \setminus S^*$  is satisfied. After this change, we have  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq 10 \cdot \text{finalwt}_\ell$ .

In both the above cases, we only changed the value of  $h$  at the variables  $\mathcal{T}_\nu$ . Since  $\mathcal{T}_\nu \cap B = \emptyset$ , we have that for every  $j \in [k]$ , the new value  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_j)$  is at least  $(1 - \frac{\epsilon}{2k})$  times the old value  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_j)$ .

(d) Set  $D = D \cup \{\ell\}$ .

(e) Set  $N = \{\nu_i^\ell \mid \ell \in H \setminus D, i \leq \lfloor t/2 \rfloor, \nu_i^\ell \prec \nu\}$ .

Observe that  $|N|$  decreases in size by at most  $\frac{t}{2} + |B|$ . Thus, if  $D \neq H$ , we have

$$\begin{aligned}
|N| &\geq |N_0| - |D| \cdot \frac{t}{2} - |D||B| \\
&= |H| \cdot \frac{t}{2} - |D| \cdot \frac{t}{2} - |D||B| \\
&\geq \frac{t}{2} - k|B| > \frac{t}{4}
\end{aligned}$$

Also observe that we only changed the values of  $h$  at the variables  $\mathcal{T}_\nu$ . Thus for all  $\nu' \preceq \nu$  (i.e.  $\nu' \in N$ ), we still have the property that  $h|_{S_{\rho_{\nu'}}} = h^*|_{S_{\rho_{\nu'}}}$ .

For each high variance instance  $\ell \in [k]$ , in the iteration where  $\ell$  gets added to the set  $D$ , the procedure ensures that at the end of the iteration  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq 10 \cdot \text{finalwt}_\ell$ .

Moreover, at each step we reduced the value of each  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell)$  by at most  $\frac{\varepsilon}{2k}$  fraction of its previous value. Thus, at the end of the procedure, for every  $\ell \in [k]$ , the value has decreased at most by a multiplicative factor of  $(1 - \frac{\varepsilon}{2k})^k \geq (1 - \frac{\varepsilon}{2})$ . Thus, for every  $\ell \in [k]$ , we get  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \frac{\varepsilon}{2}) \cdot \text{val}(h^* \cup g_{\nu^*}, \mathcal{W}_\ell)$ , and for every high variance instance  $\ell \in [k]$ , we have  $\text{val}(h \cup g_{\nu^*}, \mathcal{W}_\ell) \geq (1 - \frac{\varepsilon}{2}) \cdot 10 \cdot \text{finalwt}_\ell$ . This proves the two properties of  $h$  that we set out to prove.

**Running time :** Running time of the algorithm is  $2^{O(kt)} \cdot \text{poly}(n)$  which is  $2^{O(k^4/\varepsilon^2 \log(k/\varepsilon^2))} \cdot \text{poly}(n)$ . ■

## 8.5 Simultaneous MAX- $w$ -SAT

In this section, we give our algorithm for simultaneous MAX- $w$ -SAT. The algorithm follows the basic paradigm from MAX-2-AND and MAX-CSP, but does not require a tree of evolutions (only a set of influential variables), and uses an LP to boost the Pareto approximation factor to  $(\frac{3}{4} - \varepsilon)$ .

### 8.5.1 Preliminaries

Let  $V$  be a set of  $n$  Boolean variables. Define  $\mathcal{C}$  to be the set of all possible  $w$ -SAT constraints on the  $n$  variable set  $V$ . A MAX- $w$ -SAT instance is then described by a weight function  $\mathcal{W} : \mathcal{C} \rightarrow \mathbb{R}_{\geq 0}$  (here  $\mathcal{W}(C)$  denotes the weight of the constraint  $C$ ). We will assume that  $\sum_{C \in \mathcal{C}} \mathcal{W}(C) = 1$ .

We say  $v \in C$  if the variable  $v$  appears in the constraint  $C$ . For a constraint  $C$ , let  $C^+$  (resp.  $C^-$ ) denote the set of variables  $v \in V$  that appear unnegated (resp. negated) in the constraint  $C$ .

Let  $f : V \rightarrow \{0, 1\}$  be an assignment. For a constraint  $C \in \mathcal{C}$ , define  $C(f)$  to be 1 if the constraint  $C$  is satisfied by the assignment  $f$ , and define  $C(f) = 0$  otherwise.

Then, we have the following expression for  $\text{val}(f, \mathcal{W})$ :

$$\text{val}(f, \mathcal{W}) \stackrel{\text{def}}{=} \sum_{C \in \mathcal{C}} \mathcal{W}(C) \cdot C(f).$$

### Active Constraints.

Our algorithm will maintain a small set  $S \subseteq V$  of variables, for which we will try all assignments by brute-force, and then use a randomized rounding procedure for a linear program to obtain an assignment for  $V \setminus S$ . We now introduce some notation for dealing with this.

Let  $S \subseteq V$ . We say a constraint  $C \in \mathcal{C}$  is active given  $S$  if at least one of the variables of  $C$  is in  $V \setminus S$ . We denote by  $\text{Active}(S)$  the set of constraints from  $\mathcal{C}$  which are active given  $S$ . For two constraints  $C_1, C_2 \in \mathcal{C}$ , we say  $C_1 \sim_S C_2$  if they share a variable that is contained in  $V \setminus S$ . Note that if  $C_1 \sim_S C_2$ , then  $C_1, C_2$  are both in  $\text{Active}(S)$ . For two partial assignments  $f_1 : S \rightarrow \{0, 1\}$  and  $f_2 : V \setminus S \rightarrow \{0, 1\}$ , let  $f = f_1 \cup f_2$  is an assignment  $f : V \rightarrow \{0, 1\}$  such that  $f(x) = f_1(x)$  if  $x \in S$  otherwise  $f(x) = f_2(x)$ .

Define the active degree of a variable  $v \in V \setminus S$  given  $S$  by:

$$\text{activedegree}_S(v, \mathcal{W}) \stackrel{\text{def}}{=} \sum_{C \in \text{Active}(S), C \ni v} \mathcal{W}(C).$$

We then define the active degree of the whole instance  $\mathcal{W}$  given  $S$ :

$$\text{activedegree}_S(\mathcal{W}) \stackrel{\text{def}}{=} \sum_{v \in V \setminus S} \text{activedegree}_S(v, \mathcal{W}).$$

For a partial assignment  $h : S \rightarrow \{0, 1\}$ , we define

$$\text{val}(h, \mathcal{W}) \stackrel{\text{def}}{=} \sum_{\substack{C \in \mathcal{C} \\ C \notin \text{Active}(S)}} \mathcal{W}(C) \cdot C(h).$$

Thus, for an assignment  $g : V \setminus S \rightarrow \{0, 1\}$ , to the remaining set of variables, we have the equality:

$$\text{val}(h \cup g, \mathcal{W}) - \text{val}(h, \mathcal{W}) = \sum_{C \in \text{Active}(S)} \mathcal{W}(C) \cdot C(h \cup g).$$

### LP Rounding.

Let  $h : S \rightarrow \{0, 1\}$  be a partial assignment. We will use the Linear Program MAXwSAT-LP<sub>1</sub>( $h$ ) to complete the assignment to  $V \setminus S$ . For MAX-2-SAT, Goemans and Williamson [GW93] showed, via a rounding procedure, that this LP can be used to give a  $3/4$  approximation. However, as in MAX-2-AND, we will be using the rounding procedure due to Trevisan [Tre98] that also gives a  $3/4$  approximation for MAX- $w$ -SAT, because of its smoothness properties.

Let  $\vec{t}, \vec{z}$  be a feasible solution to the LP MAXwSAT-LP<sub>1</sub>( $h$ ). Let  $\text{smooth}(\vec{t})$  denote the map  $p : V \setminus S \rightarrow [0, 1]$  given by:  $p(v) = \frac{1}{4} + \frac{t_v}{2}$ . Note that  $p(v) \in [1/4, 3/4]$  for all  $v$ .

**Theorem 8.5.1.** *Let  $h : S \rightarrow \{0, 1\}$  be a partial assignment.*

1. *For every  $g_0 : V \setminus S \rightarrow \{0, 1\}$ , there exist  $\vec{t}, \vec{z}$  satisfying MAXwSAT-LP<sub>1</sub>( $h$ ) such that for every MAX- $w$ -SAT instance  $\mathcal{W}$ :*

$$\sum_{C \in \mathcal{C}} \mathcal{W}(C) z_C = \text{val}(g_0 \cup h, \mathcal{W}).$$

2. *Suppose  $\vec{t}, \vec{z}$  satisfy MAXwSAT-LP<sub>1</sub>( $h$ ). Let  $p = \text{smooth}(\vec{t})$ . Then for every MAX- $w$ -SAT instance  $\mathcal{W}$ :*

$$\mathbf{E}_g[\text{val}(h \cup g, \mathcal{W})] \geq \frac{3}{4} \cdot \sum_{C \in \mathcal{C}} \mathcal{W}(C) z_C,$$

where  $g : V \setminus S \rightarrow \{0, 1\}$  is such that each  $g(v)$  is sampled independently with  $\mathbf{E}[g(v)] = p(v)$ .

*Proof.* The first part is identical to the first part of Lemma 8.3.2. For the second part.

Let  $\mathcal{W}$  be any instance of MAX- $w$ -SAT. Let  $g : V \setminus S \rightarrow \{0, 1\}$  be sampled as follows: independently for each  $v \in V \setminus S$ ,  $g(v)$  is sampled from  $\{0, 1\}$  such that  $\mathbf{E}[g(v)] = p(v)$ .

We need to show that

$$\begin{aligned} \mathbf{E}_g[\text{val}(h \cup g, \mathcal{W})] &= \sum_{C \in \mathcal{C} \setminus \text{Active}(S)} \mathcal{W}(C) C(h) + \mathbf{E} \left[ \sum_{C \in \text{Active}(S)} \mathcal{W}(C) C(\rho \cup g) \right] \\ &\geq \frac{3}{4} \cdot \sum_{C \in \mathcal{C} \setminus \text{Active}(S)} \mathcal{W}(C) z_C + \frac{3}{4} \cdot \sum_{C \in \text{Active}(S)} \mathcal{W}(C) z_C \end{aligned}$$

For  $C \in \mathcal{C} \setminus \text{Active}(S)$ , it is easy to verify that if  $z_C > 0$ , we must have  $C(h) = 1$ . For  $C \in \text{Active}(S)$  the following claim gives us the required inequality:

**Claim 8.5.2.** *For  $C \in \text{Active}(S)$ ,  $\mathbf{E}[C(h \cup g)] \geq \frac{3}{4} \cdot z_C$ .*

*Proof.* The claim is true if  $C$  is satisfied by  $h$ . Consider a clause  $C$  which contains  $l$  active variables but not satisfied by partial assignment  $h$ . Under the smooth rounding, we have

$$\begin{aligned} \mathbf{E}[C(h \cup g)] &= \Pr[C \text{ is satisfied by } h \cup g] \\ &= 1 - \left( \prod_{v \in C^+, v \in V \setminus S} \frac{3}{4} - \frac{t_v}{2} \right) \cdot \left( \prod_{v \in C^-, v \in V \setminus S} \frac{3}{4} - \frac{1 - t_v}{2} \right) \\ &\geq 1 - \left( \frac{3}{4} - \frac{\sum_{v \in C^+, v \in V \setminus S} t_v + \sum_{v \in C^-, v \in V \setminus S} (1 - t_v)}{2l} \right)^l \\ &\geq 1 - \left( \frac{3}{4} - \frac{z_C}{2l} \right)^l \geq \frac{3}{4} \cdot z_C, \end{aligned}$$

where first inequality follows from AM-GM inequality. For any integer  $l \geq 1$ , the last inequality follows by noting that for a function  $f(x) = 1 - \left(\frac{3}{4} - \frac{x}{2l}\right)^l - \frac{3}{4} \cdot x$ ,  $f(0) \geq 0$ ,  $f(1) \geq 0$  along with the fact the the function has no local minima in  $(0, 1)$ . ■

■

### 8.5.2 Random Assignments

We now give a sufficient condition for the value of a MAX- $w$ -SAT instance to be highly concentrated under a *sufficiently smooth* independent random assignment to the variables of  $V \setminus S$  (This smooth distribution will come from the rounding algorithm for the LP). When the condition does not hold, we will get a variable of high active degree.

Let  $S \subseteq V$ , and let  $h : S \rightarrow \{0, 1\}$  be an arbitrary partial assignment to  $S$ . Let  $p : V \setminus S \rightarrow [0, 1]$  be such that  $p(v) \in [1/4, 3/4]$  for each  $v \in V \setminus S$ . Consider the random assignment  $g : V \setminus S \rightarrow \{0, 1\}$ , where for each  $v \in V \setminus S$ ,  $g(v) \in \{0, 1\}$  is sampled independently with  $\mathbf{E}[g(v)] = p(v)$ . Define the random variable

$$Y \stackrel{\text{def}}{=} \text{val}(h \cup g, \mathcal{W}) - \text{val}(h, \mathcal{W}) = \sum_{C \in \text{Active}(S)} \mathcal{W}(C) \cdot C(h \cup g).$$

The random variable  $Y$  measures the contribution of active constraints to the instance  $\mathcal{W}$ .

We now define two quantities depending only on  $S$  (and importantly, not on  $h$ ), which will be useful in controlling the expectation and variance of  $Y$ . The first quantity is an upper bound on  $\text{Var}[Y]$ :

$$\text{Uvar} \stackrel{\text{def}}{=} \sum_{C_1 \sim_S C_2} \mathcal{W}(C_1) \mathcal{W}(C_2).$$

The second quantity is a lower bound on  $\mathbf{E}[Y]$ :

$$\text{Lmean} \stackrel{\text{def}}{=} \frac{1}{4} \cdot \sum_{C \in \text{Active}(S)} \mathcal{W}(C).$$

**Lemma 8.5.3.** *Let  $S \subseteq V$  be a subset of variables and  $h : S \rightarrow \{0, 1\}$  be an arbitrary partial assignment to  $S$ . Let  $p, Y, \text{Uvar}, \text{Lmean}$  be as above.*

1. *If  $\text{Uvar} \leq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}^2$ , then  $\Pr[Y < (1 - \varepsilon_0) \mathbf{E}[Y]] < \delta_0$ .*

2. *If  $\text{Uvar} \geq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}^2$ , then there exists  $v \in V \setminus S$  such that*

$$\text{activedegree}_S(v, \mathcal{W}) \geq \frac{1}{16w^2} \varepsilon_0^2 \delta_0 \cdot \text{activedegree}_S(\mathcal{W}).$$

The crux of the proof is that independent of the assignment  $h : S \rightarrow \{0, 1\}$ ,  $\mathbf{E}[Y] \geq \text{Lmean}$  and  $\text{Var}(Y) \leq \text{Uvar}$  (this crucially requires that the rounding is independent and *smooth*, i.e.,  $p(v) \in [1/4, 3/4]$  for all  $v$ ; this is why we end up using Trevisan's rounding procedure in Theorem 8.5.1). The first part is then a simple application of the Chebyshev inequality. For the second part, we use the assumption that  $\text{Uvar}$  is large, to deduce that there exists a constraint  $C$  such that the total weight of constraints that share a variable from  $V \setminus S$  with  $C$ , i.e.,  $\sum_{C_2 \sim_S C} \mathcal{W}(C_2)$ , is large. It then follows that at least one variable  $v \in C$  must have large activedegree given  $S$ .

*Proof.* We first prove that  $\text{Var}(Y) \leq \text{Uvar}$ . Recall that the indicator variable  $C(h \cup g)$  denotes whether a constraint  $C$  is satisfied by the assignment  $h \cup g$ , and note that:

$$Y = \sum_{C \in \text{Active}(S)} \mathcal{W}(C) \cdot C(h \cup g).$$

Thus, the variance of  $Y$  is given by

$$\begin{aligned} \text{Var}(Y) &= \sum_{C_1, C_2 \in \text{Active}(S)} \mathcal{W}(C_1)\mathcal{W}(C_2) \cdot (\mathbf{E}[C_1(h \cup g)C_2(h \cup g)] - \mathbf{E}[C_1(h \cup g)] \mathbf{E}[C_2(h \cup g)]) \\ &\leq \sum_{C_1 \sim_S C_2} \mathcal{W}(C_1)\mathcal{W}(C_2) = \text{Uvar}, \end{aligned}$$

where the inequality holds because  $\mathbf{E}[C_1(h \cup g)C_2(h \cup g)] - \mathbf{E}[C_1(h \cup g)] \mathbf{E}[C_2(h \cup g)] \leq 1$  for all  $C_1, C_2$ , and  $\mathbf{E}[C_1(h \cup g)C_2(h \cup g)] - \mathbf{E}[C_1(h \cup g)] \mathbf{E}[C_2(h \cup g)] = 0$  unless  $C_1 \sim_S C_2$  because the rounding is performed independently for all the variables.

Moreover, since  $p(v) \in [1/4, 3/4]$  for all  $v$ , we get that  $\mathbf{E}[C(h \cup g)] \geq 1/4$  for all  $C \in \text{Active}(S)$ . Thus, we have  $\mathbf{E}[Y] \geq \text{Lmean}$ . Given this, the first part of the lemma easily follows from Chebyshev's inequality:

$$\Pr[Y < (1 - \varepsilon_0) \mathbf{E}[Y]] \leq \frac{\text{Var}(Y)}{\varepsilon_0^2 (\mathbf{E}[Y])^2} \leq \frac{\text{Uvar}}{\varepsilon_0^2 \text{Lmean}^2} \leq \delta_0.$$

For the second part of the lemma, we have:

$$\begin{aligned} \delta_0 \varepsilon_0^2 \text{Lmean}^2 &< \text{Uvar} = \sum_{C_1 \sim_S C_2} \mathcal{W}(C_1)\mathcal{W}(C_2) \\ &\leq \sum_{C_1 \in \text{Active}(S)} \mathcal{W}(C_1) \sum_{C_2 \sim_S C_1} \mathcal{W}(C_2) \\ &\leq \left( \sum_{C_1 \in \text{Active}(S)} \mathcal{W}(C_1) \right) \cdot \max_{C \in \text{Active}(S)} \sum_{C_2 \sim_S C} \mathcal{W}(C_2) \\ &= 4 \cdot \text{Lmean} \cdot \max_{C \in \text{Active}(S)} \sum_{C_2 \sim_S C} \mathcal{W}(C_2). \end{aligned}$$

Thus, there exists a constraint  $C \in \text{Active}(S)$  such that:

$$\sum_{C_2 \sim_S C} \mathcal{W}(C_2) \geq \frac{1}{4} \cdot \delta_0 \varepsilon_0^2 \cdot \text{Lmean} \geq \frac{1}{16w} \delta_0 \varepsilon_0^2 \cdot \text{activedegree}_S(\mathcal{W}), \quad (8.5.1)$$

where we used the fact that  $\text{Lmean} = \frac{1}{4} \cdot (\sum_{C \in \text{Active}(S)} \mathcal{W}(C)) \geq \frac{1}{4w} \cdot \text{activedegree}_S(\mathcal{W})$ , since we are counting the weight of a constraint at most  $w$  times in the expression  $\text{activedegree}_S(\mathcal{W})$ . Finally, the LHS of equation (8.5.1) is at most

$\sum_{u \in C \cap (V \setminus S)} \text{activedegree}_S(u, \mathcal{W})$ . Thus, there is some  $u \in V \setminus S$  with:

$$\text{activedegree}_S(u, \mathcal{W}) \geq \frac{1}{16w^2} \delta_0 \varepsilon_0^2 \cdot \text{activedegree}_S(\mathcal{W}).$$

■

### 8.5.3 Algorithm for Simultaneous MAX- $w$ -SAT

In Fig. 8.8, we give our algorithm for simultaneous MAX- $w$ -SAT. The input to the algorithm consists of an integer  $k \geq 1$ ,  $\varepsilon > 0$ , and  $k$  instances of MAX- $w$ -SAT, specified by weight functions  $\mathcal{W}_1, \dots, \mathcal{W}_k$ , and target objective values  $c_1, \dots, c_k$ .

### 8.5.4 Analysis of Algorithm SIM-MAXWSAT

It is easy to see that the algorithm always terminates in polynomial time. Part 2 of Lemma 8.5.3 implies that that Step 3.(d)i always succeeds in finding a variable  $v$ . Next, we note that Step 3. always terminates. Indeed, whenever we find an instance  $\ell \in [k]$  in Step 3.d such that  $\text{count}_\ell < t$  and  $\text{flag}_\ell = \text{TRUE}$ , we increment  $\text{count}_\ell$ . This can happen only  $tk$  times before the condition  $\text{count}_\ell < t$  fails for all  $\ell \in [k]$ . Thus the loop must terminate within  $tk$  iterations.

Let  $S^*$  denote the final set  $S$  that we get at the end of Step 3. of SIM-MAXWSAT. To analyze the approximation guarantee of the algorithm, we classify instances according to how many vertices were brought into  $S^*$  because of them.

**Definition 8.5.4** (Low and high variance instances). *At the completion of Step 3.d in Algorithm SIM-MAXWSAT, if  $\ell \in [k]$  satisfies  $\text{count}_\ell = t$ , we call instance  $\ell$  a high variance instance. Otherwise we call instance  $\ell$  a low variance instance.*

At a high level, the analysis will go as follows: First we analyze what happens when we give the optimal assignment to  $S^*$  in Step 4. For low variance instances, the fraction of the constraints satisfied by the LP rounding will concentrate around its expectation, and will give the desired approximation. For every high variance instance, we will see that many of its “heavy-weight” vertices were brought into  $S^*$ , and we will use this to argue that we can satisfy a large fraction of the constraints from these high variance instances by suitably perturbing the optimal assignment to  $S^*$  to these “heavy-weight” vertices. It is crucial that this perturbation is carried out without significantly affecting the value of the low variance instances.

Let  $f^* : V \rightarrow \{0, 1\}$  be an assignment such that  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for each  $\ell$ . Let  $h^* = f^*|_{S^*}$ . Claim 1 from Theorem 8.5.1 implies that MAXwSAT-LP<sub>2</sub>( $h^*$ ) has a feasible

**Input:**  $k$  instances of MAX- $w$ -SAT  $\mathcal{W}_1, \dots, \mathcal{W}_k$  on the variable set  $V$ , target objective values  $c_1, \dots, c_k$ , and  $\varepsilon > 0$ .

**Output:** An assignment to  $V$ .

**Parameters:**  $\delta_0 = \frac{1}{10k}$ ,  $\varepsilon_0 = \frac{\varepsilon}{2}$ ,  $\gamma = \frac{\varepsilon_0^2 \delta_0}{16w^2}$ ,  $t = \frac{2k}{\gamma} \cdot \log\left(\frac{11}{\gamma}\right)$ .

1. Initialize  $S \leftarrow \emptyset$ .
2. For each instance  $\ell \in [k]$ , initialize  $\text{count}_\ell \leftarrow 0$  and  $\text{flag}_\ell \leftarrow \text{TRUE}$ .
3. Repeat the following until for every  $\ell \in [k]$ , either  $\text{flag}_\ell = \text{FALSE}$  or  $\text{count}_\ell = t$ :
  - (a) For each  $\ell \in [k]$ , compute  $\text{Uvar}_\ell = \sum_{C_1 \sim_S C_2} \mathcal{W}_\ell(C_1) \mathcal{W}_\ell(C_2)$ .
  - (b) For each  $\ell \in [k]$ , compute  $\text{Lmean}_\ell = \frac{1}{4} \sum_{C \in \text{Active}(S)} \mathcal{W}_\ell(C)$ .
  - (c) For each  $\ell \in [k]$ , if  $\text{Uvar}_\ell \geq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2$ , then set  $\text{flag}_\ell = \text{TRUE}$ , else set  $\text{flag}_\ell = \text{FALSE}$ .
  - (d) Choose any  $\ell \in [k]$ , such that  $\text{count}_\ell < t$  AND  $\text{flag}_\ell = \text{TRUE}$  (if any):
    - i. Find a variable  $v \in V$  such that  $\text{activedegree}_S(v, \mathcal{W}_\ell) \geq \gamma \cdot \text{activedegree}_S(\mathcal{W}_\ell)$ .
    - ii. Set  $S \leftarrow S \cup \{v\}$ . We say that  $v$  was brought into  $S$  because of instance  $\ell$ .
    - iii. Set  $\text{count}_\ell \leftarrow \text{count}_\ell + 1$ .
4. For each partial assignment  $h_0 : S \rightarrow \{0, 1\}$ :
  - (a) If there is a feasible solution  $\vec{t}, \vec{z}$  to the LP in Fig. 8.10, set  $p = \text{smooth}(\vec{t})$ . If not, return to Step 4. and proceed to the next  $h_0$ .
  - (b) Define  $g : V \setminus S \rightarrow \{0, 1\}$  by independently sampling  $g(v) \in \{0, 1\}$  with  $\mathbf{E}[g(v)] = p(v)$ , for each  $v \in V \setminus S$ .
  - (c) For each  $h : S \rightarrow \{0, 1\}$ , compute  $\text{out}_{h,g} = \min_{\ell \in [k]} \frac{\text{val}(h \cup g, \mathcal{W}_\ell)}{c_\ell}$ . If  $c_\ell = 0$  for some  $\ell \in [k]$ , we interpret  $\frac{\text{val}(h \cup g, \mathcal{W}_\ell)}{c_\ell}$  as  $+\infty$ .
5. Output the largest  $\text{out}_{h,g}$  seen, and the assignment  $h \cup g$ .

Figure 8.8: Algorithm SIM-MAXWSAT for approximating weighted simultaneous

$$\begin{aligned}
\sum_{v \in C^+} t_v + \sum_{v \in C^-} (1 - t_v) &\geq z_C & \forall C \in \mathcal{C} \\
1 \geq z_C &\geq 0 & \forall C \in \mathcal{C} \\
1 \geq t_v &\geq 0 & \forall v \in V \setminus S \\
t_v &= h_0(v) & \forall v \in S
\end{aligned}$$

Figure 8.9: Linear program MAXwSAT-LP<sub>1</sub>( $h_0$ ), for a given partial assignment  $h_0 : S \rightarrow \{0, 1\}$

$$\begin{aligned}
\sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) \cdot z_C &\geq c_\ell \quad \forall \ell \in [k] \\
\vec{t}, \vec{z} &\text{ satisfy MAXwSAT-LP}_1(h_0).
\end{aligned}$$

Figure 8.10: Linear program MAXwSAT-LP<sub>2</sub>( $h_0$ ) for a given partial assignment  $h_0 : S \rightarrow \{0, 1\}$

solution. For low variance instances, by combining Theorem 8.5.1 and Lemma 8.5.3, we show that  $\text{val}(h^* \cup g, \mathcal{W}_\ell)$  is at least  $(3/4 - \varepsilon/2) \cdot c_\ell$  with high probability.

**Lemma 8.5.5.** *Let  $\ell \in [k]$  be any low variance instance. Let  $\vec{t}, \vec{z}$  be a feasible solution to MAXwSAT-LP<sub>2</sub>( $h^*$ ). Let  $p = \text{smooth}(\vec{t})$ . Let  $g : V \setminus S^* \rightarrow \{0, 1\}$  be such that each  $g(v)$  is sampled independently with  $\mathbf{E}[g(v)] = p(v)$ . Then the assignment  $h^* \cup g$  satisfies:*

$$\Pr_g [\text{val}(h^* \cup g, \mathcal{W}_\ell) \geq (3/4 - \varepsilon/2) \cdot c_\ell] \geq 1 - \delta_0.$$

*Proof.* Since  $\ell$  is a low variance instance,  $\text{flag}_\ell = \text{FALSE}$  when the algorithm terminates. Thus  $\text{Uvar}_\ell < \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2$ . Let  $g : V \rightarrow \{0, 1\}$  be the random assignment picked in Step 4.b. Define the random variable

$$Y_\ell \stackrel{\text{def}}{=} \text{val}(h^* \cup g, \mathcal{W}_\ell) - \text{val}(h^*, \mathcal{W}_\ell).$$

By Lemma 8.5.3, we know that with probability at least  $1 - \delta_0$ , we have  $Y_\ell \geq$

$(1 - \varepsilon_0) \mathbf{E}[Y_\ell]$ . Thus, with probability at least  $1 - \delta_0$ , we have,

$$\begin{aligned} \text{val}(h^* \cup g, \mathcal{W}_\ell) &= \text{val}(h^*, \mathcal{W}_\ell) + Y_\ell \geq \text{val}(h^*, \mathcal{W}_\ell) + (1 - \varepsilon_0) \mathbf{E}[Y_\ell] \\ &\geq (1 - \varepsilon_0) \cdot \mathbf{E}[\text{val}(h^*, \mathcal{W}_\ell) + Y_\ell] = (1 - \varepsilon_0) \cdot \mathbf{E}[\text{val}(h^* \cup g, \mathcal{W}_\ell)] \\ &\geq 3/4 \cdot (1 - \varepsilon_0) \cdot \sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) z_C \geq (3/4 - \varepsilon/2) \cdot c_\ell, \end{aligned}$$

where the last two inequalities follow from Claim 2 in Theorem 8.5.1 and the constraints in MAXwSAT-LP<sub>2</sub> respectively.  $\blacksquare$

Now we analyze the high variance instances. We prove the following lemma that proves that at the end of the algorithm, the activedegree of high variance instances is small, and is dominated by the activedegree of any variable that was included in  $S$  “early on”.

**Lemma 8.5.6.** *For all high variance instances  $\ell \in [k]$ , we have*

1.  $\text{activedegree}_{S^*}(\mathcal{W}_\ell) \leq w(1 - \gamma)^t$ .
2. *For each of the first  $t/2$  variables that were brought inside  $S^*$  because of instance  $\ell$ , the total weight of constraints incident on each of that variable and totally contained inside  $S^*$  is at least  $10 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ .*

The crucial observation is that when a variable  $u$  is brought into  $S$  because of an instance  $\ell$ , the activedegree of  $u$  is at least a  $\gamma$  fraction of the total activedegree of instance  $\ell$ . Thus, the activedegree of instance  $\ell$  goes down by a multiplicative factor of  $(1 - \gamma)$ . This immediately implies the first part of the lemma. For the second part, we use the fact that  $t$  is large, and hence the activedegree of early vertices must be much larger than the final activedegree of instance  $\ell$ .

*Proof.* Consider any *high variance* instance  $\ell \in [k]$ . Initially, when  $S = \emptyset$ , we have  $\text{activedegree}_\emptyset(\mathcal{W}_\ell) \leq w$  since the weight of every constraint is counted at most  $w$  times, once for each of the 2 active variables of the constraint, and  $\sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) = 1$ . For every  $v$ , note that  $\text{activedegree}_{S_2}(v, \mathcal{W}_\ell) \leq \text{activedegree}_{S_1}(v, \mathcal{W}_\ell)$  whenever  $S_1 \subseteq S_2$ .

Let  $u$  be one of the variables that ends up in  $S^*$  because of instance  $\ell$ . Let  $S_u$  denote the set  $S \subseteq S^*$  just before  $u$  was brought into  $S^*$ . When  $u$  is added to  $S_u$ , we know

that  $\text{activedegree}_{S_u}(u, \mathcal{W}_\ell) \geq \gamma \cdot \text{activedegree}_{S_u}(\mathcal{W}_\ell)$ . Hence,  $\text{activedegree}_{S_u \cup \{u\}}(\mathcal{W}_\ell) \leq \text{activedegree}_{S_u}(\mathcal{W}_\ell) - \text{activedegree}_{S_u}(u, \mathcal{W}_\ell) \leq (1 - \gamma) \cdot \text{activedegree}_{S_u}(\mathcal{W}_\ell)$ . Since  $t$  variables were brought into  $S^*$  because of instance  $\ell$ , and initially  $\text{activedegree}_\emptyset(\mathcal{W}_\ell) \leq w$ , we get  $\text{activedegree}_{S^*}(\mathcal{W}_\ell) \leq w(1 - \gamma)^t$ .

Now, let  $u$  be one of the first  $t/2$  variables that ends up in  $S^*$  because of instance  $\ell$ . Since at least  $t/2$  variables are brought into  $S^*$  because of instance  $\ell$ , after  $u$ , as above, we get  $\text{activedegree}_{S^*}(\mathcal{W}_\ell) \leq (1 - \gamma)^{t/2} \cdot \text{activedegree}_{S_u}(\mathcal{W}_\ell)$ . Combining with  $\text{activedegree}_{S_u}(u, \mathcal{W}_\ell) \geq \gamma \cdot \text{activedegree}_{S_u}(\mathcal{W}_\ell)$ , we get  $\text{activedegree}_{S_u}(u, \mathcal{W}_\ell) \geq \gamma(1 - \gamma)^{-t/2} \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ , which is at least  $11 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ , by the choice of parameters. Since any constraint incident on a vertex in  $V \setminus S^*$  contributes its weight to  $\text{activedegree}_{S^*}(\mathcal{W}_\ell)$ , the total weight of constraints incident on  $u$  and totally contained inside  $S^*$  is at least  $10 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$  as required. ■

We now describe a procedure **PERTURB** (see Fig. 8.11) which takes  $h^* : S^* \rightarrow \{0, 1\}$  and  $g : V \setminus S^* \rightarrow \{0, 1\}$ , and produces a new  $h : S^* \rightarrow \{0, 1\}$  such that for all (low variance as well as high variance) instances  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell)$  is not much smaller than  $\text{val}(h^* \cup g, \mathcal{W}_\ell)$ , and furthermore, for all high variance instances  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell)$  is large. The procedure works by picking a special variable in  $S^*$  for every high variance instance and perturbing the assignment of  $h^*$  to these special variables. The crucial feature used in the perturbation procedure, which holds for **MAX- $w$ -SAT** (but not for **MAX-2-AND**), is that it is possible to satisfy a constraint by just changing one of the variables it depends on. The partial assignment  $h$  is what we will be using to argue that Step 4. of the algorithm produces a good Pareto approximation. More formally, we have the following Lemma.

**Lemma 8.5.7.** *For the assignment  $h$  obtained from Procedure **PERTURB** (see Fig. 8.11), for each  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^* \cup g, \mathcal{W}_\ell)$ . Furthermore, for each high variance instance  $\mathcal{W}_\ell$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq 4 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ .*

*Proof.* Consider the special variable  $v_\ell$  that we choose for *high variance* instance  $\ell \in [k]$ . Since  $v_\ell \notin B$ , the constraints incident on  $v_\ell$  only contribute at most a  $\varepsilon/2k$  fraction of the objective value in each instance. Thus, changing the assignment  $v_\ell$  can reduce the

**Input:**  $h^* : S^* \rightarrow \{0, 1\}$  and  $g : V \setminus S^* \rightarrow \{0, 1\}$

**Output:** A perturbed assignment  $h : S^* \rightarrow \{0, 1\}$ .

1. Initialize  $h \leftarrow h^*$ .
2. For  $\ell = 1, \dots, k$ , if instance  $\ell$  is a high variance instance case (i.e.,  $\text{count}_\ell = t$ ), we pick a special variable  $v_\ell \in S^*$  associated to this instance as follows:
  - (a) Let  $B = \{v \in V \mid \exists \ell \in [k] \text{ with } \sum_{C \in \mathcal{C}, C \ni v} \mathcal{W}_\ell(C) \cdot C(h \cup g) \geq \frac{\varepsilon}{2k} \cdot \text{val}(h \cup g, \mathcal{W}_\ell)\}$ . Since the weight of each constraint is counted at most  $w$  times, we know that  $|B| \leq \frac{2wk^2}{\varepsilon}$ .
  - (b) Let  $U$  be the set consisting of the first  $t/2$  variables brought into  $S^*$  because of instance  $\ell$ .
  - (c) Since  $t/2 > |B| + k$ , there exists some  $u \in U$  such that  $u \notin B \cup \{v_1, \dots, v_{\ell-1}\}$ . We define  $v_\ell$  to be  $u$ .
  - (d) By Lemma 8.5.6, the total  $\mathcal{W}_\ell$  weight of constraints that are incident on  $v_\ell$  and only containing variables from  $S^*$  is at least  $10 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ . We update  $h$  by setting  $h(v_\ell)$  to be that value from  $\{0, 1\}$  such that at least half of the  $\mathcal{W}_\ell$  weight of these constraints is satisfied.
3. Return the assignment  $h$ .

Figure 8.11: Procedure PERTURB for perturbing the optimal assignment

value of any instance by at most a  $\frac{\varepsilon}{2k}$  fraction of their current objective value. Also, we pick different special variables for each *high variance* instance. Hence, the total effect of these perturbations on any instance is that it reduces the objective value (given by  $h^* \cup g$ ) by at most  $1 - (1 - \frac{\varepsilon}{2k})^k \leq \frac{\varepsilon}{2}$  fraction. Hence for all instances  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^* \cup g, \mathcal{W}_\ell)$ .

For a *high variance instance*  $\ell \in [k]$ , since  $v_\ell \in U$ , the variable  $v_\ell$  must be one of the first  $t/2$  variables brought into  $S^*$  because of  $\ell$ . Hence, by Lemma 8.5.6 the total weight of constraints that are incident on  $v_\ell$  and entirely contained inside  $S^*$  is at least  $10 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ . Hence, there is an assignment to  $v_\ell$  that satisfies at least at least half the weight of these MAX- $w$ -SAT constraints<sup>8</sup> in  $\ell$ . At the end of the iteration, when we pick an assignment to  $v_\ell$ , we have  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq 5 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ . Since the later perturbations do not affect value of this instance by more than  $\varepsilon/2$  fraction, we get that for the final assignment  $h$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot 5 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell) \geq 4 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ .  $\blacksquare$

Given all this, we now show that with high probability the algorithm finds an assignment that satisfies, for each  $\ell \in [k]$ , at least  $(3/4 - \varepsilon) \cdot c_\ell$  weight from instance  $\mathcal{W}_\ell$ . The following theorem immediately implies Theorem 8.1.3.

**Theorem 8.5.8.** *Let  $w$  be a constant. Suppose we're given  $\varepsilon \in (0, 2/5]$ ,  $k$  simultaneous MAX- $w$ -SAT instances  $\mathcal{W}_1, \dots, \mathcal{W}_\ell$  on  $n$  variables, and target objective value  $c_1, \dots, c_k$  with the guarantee that there exists an assignment  $f^*$  such that for each  $\ell \in [k]$ , we have  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$ . Then, the algorithm SIM-MAXWSAT runs in time  $2^{O(k^3/\varepsilon^2 \log(k/\varepsilon^2))}$ .  $\text{poly}(n)$ , and with probability at least 0.9, outputs an assignment  $f$  such that for each  $\ell \in [k]$ , we have,  $\text{val}(f, \mathcal{W}_\ell) \geq (3/4 - \varepsilon) \cdot c_\ell$ .*

*Proof.* Consider the iteration of Step 4. of the algorithm when  $h_0$  is taken to equal  $h^*$ . Then, by Part 1 of Theorem 8.5.1, the LP in Step 4.a will be feasible (this uses the fact that  $\text{val}(f^*, \mathcal{W}_\ell) \geq c_\ell$  for each  $\ell$ ).

By Lemma 8.5.5 and a union bound, with probability at least  $1 - k\delta_0 > 0.9$ , over

---

<sup>8</sup>This is not true if they are MAX-2-AND constraints.

the choice of  $g$ , we have that for *every* low variance instance  $\ell \in [k]$ ,  $\text{val}(h^* \cup g, \mathcal{W}_\ell) \geq (3/4 - \varepsilon/2) \cdot c_\ell$ . Henceforth we assume that the assignment  $g$  sampled in Step 4.b of the algorithm is such that this event occurs. Let  $h$  be the output of the procedure PERTURB given in Fig. 8.11 for the input  $h^*$  and  $g$ . By Lemma 8.5.7,  $h$  satisfies

1. For every instance  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^* \cup g, \mathcal{W}_\ell)$ .
2. For every high variance instance  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq 4 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)$ .

We now show that the desired Pareto approximation behavior is achieved when  $h$  is considered as the partial assignment in Step 4.c of the algorithm. We analyze the guarantee for low and high variance instances separately.

For any *low variance* instance  $\ell \in [k]$ , from property 1 above, we have  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^* \cup g, \mathcal{W}_\ell)$ . Since we know that  $\text{val}(h^* \cup g, \mathcal{W}_\ell) \geq (3/4 - \varepsilon/2) \cdot c_\ell$ , we have  $\text{val}(h \cup g, \mathcal{W}_\ell) \geq (3/4 - \varepsilon) \cdot c_\ell$ .

For every high variance instance  $\ell \in [k]$ , since  $h^* = f^*|_{S^*}$ , for any  $g$  we must have,

$$\text{val}(h^* \cup g, \mathcal{W}_\ell) \geq \text{val}(f^*, \mathcal{W}_\ell) - \text{activedegree}_{S^*}(\mathcal{W}_\ell) \geq c_\ell - \text{activedegree}_{S^*}(\mathcal{W}_\ell).$$

Combining this with properties 1 and 2 above, we get,

$$\begin{aligned} \text{val}(h \cup g, \mathcal{W}_\ell) &\geq (1 - \varepsilon/2) \cdot \max\{c_\ell - \text{activedegree}_{S^*}(\mathcal{W}_\ell), 4 \cdot \text{activedegree}_{S^*}(\mathcal{W}_\ell)\} \\ &\geq (3/4 - \varepsilon) \cdot c_\ell. \end{aligned}$$

Thus, for all instances  $\ell \in [k]$ , we get  $\text{val}(h \cup g) \geq (3/4 - \varepsilon) \cdot c_\ell$ . Since we are taking the best assignment  $h \cup g$  at the end of the algorithm SIM-MAXWSAT, the theorem follows.

**Running time :** Running time of the algorithm is  $2^{O(kt)} \cdot \text{poly}(n)$  which is  $2^{O(k^3/\varepsilon^2 \log(k/\varepsilon^2))} \cdot \text{poly}(n)$ . ■

## 8.6 Hardness Results for Large $k$

In this section, we prove our hardness results for simultaneous CSPs. Recall the theorem that we are trying to show.

**Theorem 8.6.1** (restated). *Assume the Exponential Time Hypothesis. Let  $\mathcal{F}$  be a fixed finite set of Boolean predicates. If  $\mathcal{F}$  is not 0-valid or 1-valid, then for  $k = \omega(\log n)$ , then detecting positivity of  $k$ -fold simultaneous MAX- $\mathcal{F}$ -CSPs on  $n$  variables requires time superpolynomial in  $n$ .*

The main notion that we will use for our hardness reductions is the notion of a “simultaneous-implementation”.

**Definition 8.6.2** (Simultaneous-Implementation). *Let  $\{x_1, \dots, x_w\}$  be a collection of variables (called **primary** variables). Let  $P : \{0, 1\}^w \rightarrow \{\text{TRUE}, \text{FALSE}\}$  be a predicate. Let  $\{y_1, \dots, y_t\}$  be another collection of variables (called **auxiliary** variables).*

*Let  $\mathcal{C}_1, \dots, \mathcal{C}_k$  be sets of constraints on  $\{x_1, \dots, x_w, y_1, \dots, y_t\}$ , where for each  $i \in [k]$ ,  $\mathcal{C}_i$  consists of various applications of predicates to tuples of distinct variables from  $\{x_1, \dots, x_w, y_1, \dots, y_t\}$ . We say that  $\mathcal{C}_1, \dots, \mathcal{C}_k$  **simultaneously-implements**  $P$  if for every assignment to  $x_1, \dots, x_w$ , we have,*

- *If  $P(x_1, \dots, x_w) = \text{TRUE}$ , then there exists a setting of the variables  $y_1, \dots, y_t$  such that each collection  $\mathcal{C}_1, \dots, \mathcal{C}_k$  has at least one satisfied constraint.*
- *If  $P(x_1, \dots, x_w) = \text{FALSE}$ , then for every setting of the variables  $y_1, \dots, y_t$ , at least one of the collections  $\mathcal{C}_1, \dots, \mathcal{C}_k$  has no satisfied constraints.*

We say that a collection of predicates  $\mathcal{F}$  simultaneously-implements  $P$  if there is a simultaneous-implementation of  $P$  where for each collection  $\mathcal{C}_i$  ( $i \in [k]$ ), every constraint in  $\mathcal{C}_i$  is an application of some predicate from  $\mathcal{F}$ .

The utility of simultaneous-implementation lies in the following lemma.

**Lemma 8.6.3.** *Let  $P$  be a predicate. Suppose checking satisfiability of CSPs on  $n$  variables with  $m$  constraints, where each constraint is an application of the predicate  $P$ , requires time  $T(n, m)$ , with  $T(n, m) = \omega(m + n)$ . Suppose  $\mathcal{F}$  simultaneously-implements  $P$ . Then detecting positivity of  $O(m)$ -fold simultaneous MAX- $\mathcal{F}$ -CSP on  $O(m + n)$  variables requires time  $\Omega(T(n, m))$ .*

*Proof.* Suppose we have a  $P$ -CSP instance  $\Phi$  with  $m$  constraints on  $n$  variables. For each of the constraints  $C \in \Phi$ , we simultaneously-implement  $C$  using the original set of

variables as primary variables, and new auxiliary variables for each constraint. Thus, for every  $C \in \Phi$ , we obtain  $k$  MAX- $\mathcal{F}$ -CSP instances  $\mathcal{C}_1^C, \dots, \mathcal{C}_k^C$ , for some constant  $k$ . The collection of instances  $\{\mathcal{C}_i^C\}_{C \in \Phi, i \in [k]}$  constitute the  $O(m)$ -simultaneous MAX- $\mathcal{F}$ -CSP instance on  $O(m + n)$  variables.

If  $\Phi$  is satisfiable, we know that there exists an assignment to the original variables such that each  $C \in \Phi$  is satisfied. Hence, by the simultaneously-implements property, there exists an assignment to all the auxiliary variables such that each  $\mathcal{C}_i^C$  has at least one satisfied constraint. If  $\Phi$  is unsatisfiable, for any assignment to the primary variables, at least one constraint  $C$  must be unsatisfied. Hence, by the simultaneously-implements property, for any assignment to the auxiliary variables, there is an  $i \in [k]$  such that  $\mathcal{C}_i^C$  has no satisfied constraints. Thus, our simultaneous MAX- $\mathcal{F}$ -CSP instance has a non-zero objective value iff  $\Phi$  is satisfiable. Since this reduction requires only  $O(m + n)$  time, suppose we require  $T'$  time for detecting positivity of a  $O(m)$ -simultaneous MAX- $\mathcal{F}$ -CSP instance on  $O(m + n)$  variables, we must have  $T' + O(m + n) \geq T(m, n)$ , giving  $T' = \Omega(T(m, n))$  since  $T(m, n) = \omega(m + n)$ . ■

The simultaneous-implementations we construct will be based on a related notion of implementation arising in approximation preserving reductions. We recall this definition below.

**Definition 8.6.4** (Implementation). *Let  $x_1, \dots, x_w$  be a collection of variables (called **primary variables**). Let  $P : \{0, 1\}^w \rightarrow \{\text{TRUE}, \text{FALSE}\}$  be a predicate.*

*Let  $y_1, \dots, y_t$  be another collection of variables (called **auxiliary variables**). Let  $C_1, \dots, C_d$  be constraints on  $\{x_1, \dots, x_w, y_1, \dots, y_t\}$ , where for each  $i \in [d]$ , the variables feeding into  $C_i$  are all distinct.*

*We say that  $C_1, \dots, C_d$  **e-implements**  $P$  if for every assignment to  $x_1, \dots, x_w$  we have,*

- *If  $P(x_1, \dots, x_w) = \text{TRUE}$ , then there exists a setting of the variables  $y_1, \dots, y_t$  such that at least  $e$  of the constraints  $C_1, \dots, C_d$  evaluate to TRUE.*
- *If  $P(x_1, \dots, x_w) = \text{FALSE}$ , then for every setting of the variables  $y_1, \dots, y_t$ , at most  $e - 1$  of the constraints  $C_1, \dots, C_d$  evaluate to TRUE.*

We say that a collection of predicates  $\mathcal{F}$  implements  $P$  if there is some  $e$  and an  $e$ -implementation of  $C$  where all the constraints  $C_1, \dots, C_d$  come from  $\mathcal{F}$ .

We will be using following predicates in our proofs.

- **Id, Neg** : These are the unary predicates defined as  $\text{Id}(x) = x$  and  $\text{Neg}(x) = \bar{x}$ .
- **NAE**:  $w$ -ary NAE predicate on variables  $x_1, \dots, x_w$  is defined as  $\text{NAE}(x_1, \dots, x_w) = \text{FALSE}$  iff all the  $x_i$ 's are equal.
- **Equality** : Equality is a binary predicate given as  $\text{Equality}(x, y) = \text{TRUE}$  iff  $x$  equals  $y$ .

We will use the following Lemmas from [KSTW01].

**Lemma 8.6.5** ([KSTW01]). *Let  $f$  be a predicate which is not 0-valid, and which is closed under complementation. Then  $\{f\}$  implements  $\text{XOR}(x, y)$ .*

**Lemma 8.6.6** ([KSTW01]). *Let  $f$  be a predicate not closed under complementation, and let  $g$  be a predicate that is not 0-valid. Then  $\{f, g\}$  implements  $\text{Id}$ , and  $\{f, g\}$  implements  $\text{Neg}$ .*

We will now prove lemmas that will capture the property of *simultaneous implementation* which will be used in proving Theorem 8.6.1.

**Lemma 8.6.7.** *If  $\{f\}$  simultaneously-implements predicate  $\text{XOR}$  on two variables, then  $\{f\}$  also simultaneously-implements the predicate  $\text{NAE}$  on three variables.*

*Proof.* Consider an NAE constraint  $\text{NAE}(x, y, z)$ . Let  $\mathcal{A}_1, \dots, \mathcal{A}_d$  be the simultaneous implementation of constraint  $\text{XOR}(x, y)$ , using predicate  $f$  and a set of auxiliary variables  $y_1, \dots, y_t$  for some  $t$ . Similarly, let  $\mathcal{B}_1, \dots, \mathcal{B}_d$  and  $\mathcal{C}_1, \dots, \mathcal{C}_d$  be the simultaneous implementation of constraint  $\text{XOR}(y, z)$  and  $\text{XOR}(x, z)$  respectively using  $f$  and on a same set of auxiliary variables  $y_1, \dots, y_t$ , constructed by replacing the variables  $(x, y)$  in  $\{\mathcal{A}_1, \dots, \mathcal{A}_d\}$  with  $(y, z)$  and  $(x, z)$  respectively. We construct sets of constraints  $\mathcal{D}_1, \dots, \mathcal{D}_d$  as follows: for each  $i \in [d]$ ,  $\mathcal{D}_i$  consists of all constraints from  $\mathcal{A}_i, \mathcal{B}_i$ , and  $\mathcal{C}_i$ . We now show that  $\{\mathcal{D}_1, \dots, \mathcal{D}_d\}$  simultaneously-implement  $\text{NAE}(x, y, z)$ .

First, notice that  $\text{NAE}(x, y, z)$  is FALSE iff all constraints  $\text{XOR}(x, y)$ ,  $\text{XOR}(y, z)$  and  $\text{XOR}(x, z)$  are FALSE. Consider the case when  $\text{NAE}(x, y, z)$  is FALSE. Since we are using same set of auxiliary variables and the implementation is symmetric, for every setting of variables  $y_1, \dots, y_t$ , there exists a fixed  $i \in [d]$  such that each of  $\mathcal{A}_i, \mathcal{B}_i$  and  $\mathcal{C}_i$  has no satisfied constraints. And hence, instance  $\mathcal{D}_i$  has no satisfied constraints. If  $\text{NAE}(x, y, z)$  is TRUE then at least one of  $\text{XOR}(x, y)$ ,  $\text{XOR}(y, z)$  or  $\text{XOR}(x, z)$  must be TRUE. Without loss of generality, we assume that  $\text{XOR}(x, y)$  is TRUE. Thus, there exists a setting of variables  $y_1, \dots, y_t$  such that each of  $\mathcal{A}_1, \dots, \mathcal{A}_d$ , has at least one satisfied constraint, and hence each of  $\mathcal{D}_1, \dots, \mathcal{D}_d$  too has at least one such constraint.  $\blacksquare$

**Lemma 8.6.8.** *Let  $f$  be a predicate not closed under complementation, not 0-valid and not 1-valid.  $f$  can simultaneously-implement Equality.*

*Proof.* Consider an equality constraint  $\text{Equality}(x, y)$ , our aim is to simultaneously-implement this constraint using predicate  $f$ .

Since  $f$  satisfies the properties of Lemma 8.6.6, we can implement  $\text{Id}(x)$  and  $\text{Id}(y)$  using  $f$ . Let  $X_1^T, \dots, X_{d_1}^T$  be an  $e_1$ -implementation of  $\text{Id}(x)$  using  $f$  and some set of auxiliary variables  $A_1$  for some  $e_1 < d_1$ . Similarly, let  $Y_1^T, \dots, Y_{d_1}^T$  be an  $e_1$ -implementation of  $\text{Id}(y)$  using  $f$  and a set of auxiliary variables  $A_2$ .

We can also implement  $\text{Neg}(x)$  and  $\text{Neg}(y)$  using  $f$ . Let  $X_1^F, \dots, X_{d_2}^F$  be an  $e_2$ -implementation of  $\text{Neg}(x)$  using  $f$  and a set of auxiliary variables  $B_1$  for some  $e_2 < d_2$ . Similarly, let  $Y_1^F, \dots, Y_{d_1}^F$  be an  $e_2$ -implementation of  $\text{Neg}(y)$  using  $f$  and a set of auxiliary variables  $B_2$ .

We now describe the construction of the simultaneous-implementation. The implementation uses all auxiliary variables in  $A_1, A_2, B_1$ , and  $B_2$ . Each instance in the simultaneous-implementation is labeled by a tuple  $(M, N, a, b)$  where  $M \subseteq [d_1]$  with  $|M| = d_1 - e_1 + 1$ ,  $N \subseteq [d_2]$  with  $|N| = d_2 - e_2 + 1$ , and  $(a, b) \in \{(T, F), (F, T)\}$ . An instance corresponding to a tuple  $(M, N, a, b)$  has following set of constraints in  $f$ :

$$\{X_m^a, Y_n^b | m \in M, n \in N\}$$

We will now prove the simultaneous-implementation property of the above created instance. Consider the case when  $x = y = \text{TRUE}$  (other case being similar). We know

that in this case, there exists a setting of auxiliary variables  $A_1$  used in the implementation of  $\text{ld}(x)$  which satisfies at least  $e_1$  constraints out of  $X_1^T, \dots, X_{d_1}^T$ . Similarly, there exists a setting of auxiliary variables  $A_2$  used in the implementation of  $\text{ld}(y)$  which satisfies at least  $e_1$  constraints out of  $Y_1^T, \dots, Y_{d_1}^T$ . Fix this setting of auxiliary variables in  $A_1, A_2$ , and any arbitrary setting for auxiliary variables in  $B_1$  and  $B_2$ . Thus, the instance labeled by tuple the  $(M, N, a, b)$  either contains  $d_1 - e_1 + 1$  constraints from  $X_1^T, \dots, X_{d_1}^T$  if  $a = T$ , or else, it contains  $d_1 - e_1 + 1$  constraints from  $Y_1^T, \dots, Y_{d_1}^T$ . In any case, the property of  $e_1$ -implementation implies that at least one constraint is satisfied for this instance.

Now we need to show that if  $x \neq y$ , then for any setting of auxiliary variables, there exists an instance which has no satisfied constraints. Consider the case when  $x = \text{TRUE}$  and  $y = \text{FALSE}$  (other case being similar). Consider any fixed assignment to the auxiliary variables in  $A_1, A_2, B_1$ , and  $B_2$ . We know that for this fixed assignment to the auxiliary variables in  $B_1$ , there exists a subset  $N \subseteq [d_2]$  of size at least  $d_2 - e_2 + 1$ , such that all constraints in  $\{X_j^F | j \in N\}$  are unsatisfied. Similarly, for this fixed assignment to variables in  $A_2$ , there exists a subset  $M \subseteq [d_1]$  of size at least  $d_1 - e_1 + 1$  such that all constraints in  $\{Y_i^T | i \in M\}$  are unsatisfied. Thus, the instance corresponding to tuple  $(M, N, F, T)$  has no satisfied constraints. ■

We now prove Theorem 8.6.1.

*Proof.* We take cases on whether  $\mathcal{F}$  contains some  $f$  which is closed under complementation.

**Case 1:** Suppose there exists some  $f \in \mathcal{F}$  which is closed under complementation. In this case, it is enough to show that  $f$  simultaneously-implements XOR. To see this, assume that we can simultaneously-implement XOR using  $f$ . Hence, by Lemma 8.6.7, we can simultaneously-implement the predicate NAE on *three* variables using  $f$ . We start with an NAE-3-SAT instance  $\phi$ , on  $n$  variables with  $m$  constraints. For each constraint  $C \in \phi$ , we create a set of  $O(1)$  many instances which simultaneously-implement  $C$ . The final simultaneous instance is the collection of all instances that we get with each simultaneous-implementation of constraints in  $\phi$ .

In the completeness case, when  $\phi$  is satisfiable, then by the property of simultaneous-implementation, we have that there exists a setting of auxiliary variables, from each implementation of NAE constraints, such that each instance has at least one constraint satisfied. And hence, the value of the final simultaneous instance is non *zero*.

In the soundness case, for any assignment to the variables  $x_1, \dots, x_n$  there exists a constraint (say  $C$ ) which is not satisfied. Hence one of the instance from the simultaneous implementation of this constraint has value *zero* no matter how we set the auxiliary variables. And hence, the whole simultaneous instance has value zero in this case.

To prove the theorem in this case, it remains to show that we can simultaneously-implement  $\text{XOR}(x, y)$  using  $f$ . Since  $f$  is closed under complementation, we can  $e$ -implement  $\text{XOR}$  using  $f$  (for some  $e$ ) by Lemma 8.6.5. Let  $C_1, \dots, C_d$  be the set of  $f$ -constraints that we get from this  $e$ -implementation,  $e < d$ . The collection of instances contains one instance for every subset  $J \subseteq [d]$  of size  $d - e + 1$ . The instance labeled by  $J \subseteq [d]$  contains all constraints from the set  $\{C_j | j \in J\}$ . Hence, there  $\binom{d}{e-1}$  instances in the collection. Note that we used the same set of auxiliary variables in this simultaneous-implementation. We now show that this collection of instances simultaneously-implements  $\text{XOR}(x, y)$ . To see this, consider the case when  $\text{XOR}(x, y)$  is TRUE. Thus, there is an assignment to the auxiliary variables that satisfies at least  $e$  constraints out of  $C_1, \dots, C_d$ . Hence, for this particular assignment, the instance labeled by  $J$ , where  $J \subseteq [d]$  is any subset of size  $d - e + 1$ , has at least one satisfied constraint. When  $\text{XOR}(x, y)$  is FALSE, then for any assignment to the auxiliary variables, there is some  $J \subseteq [d]$  of size  $d - e + 1$  such that no constraints in the set  $\{C_j | j \in J\}$  are satisfied. Hence, for this assignment, the instance labeled with  $J$  has no satisfied constraints. This shows that  $f$  simultaneously-implements predicate  $\text{XOR}$  on two variables.

Combining the two arguments above, we get that  $\{f\}$  simultaneously-implements 3-NAE. Since 3-NAE has a linear time gadget reduction from 3-SAT [Sch78], and the ETH implies that 3-SAT on  $s$  variables and  $O(s)$  clauses requires time  $2^{\Omega(s)}$  [IP01, IPZ01], we get that checking satisfiability of a 3-NAE instance with  $\omega(\log n)$  constraints on  $\omega(\log n)$  variables requires time super-polynomial in  $n$ . Thus, using Lemma 8.6.3 implies that detecting positivity of an  $\omega(\log n)$ -simultaneous MAX- $f$ -CSP requires time

superpolynomial in  $n$ .

**Case 2:** Suppose that for all  $f \in \mathcal{F}$ ,  $f$  is not closed under complementation. Let  $f \in \mathcal{F}$  be any predicate of arity  $r$ . Since,  $f$  is not closed under complementation, there exist  $\alpha, \beta \in \{0, 1\}^r$  that satisfy  $\alpha_i \oplus \beta_i = 1$  for all  $i \in [r]$ , and  $f(\alpha) = 0$ ,  $f(\beta) = 1$ . We can reduce a 3-SAT instance with  $n$  variables and  $m = \text{poly}(n)$  clauses to  $m$  simultaneous instances over  $n$  variables involving the predicate  $f$ . For every clause  $C$  of the form  $x \vee y \vee z$ , we create an instance with 3 equal weight constraints  $\{f(\alpha \oplus (x, \dots, x)), f(\alpha \oplus (y, \dots, y)), f(\alpha \oplus (z, \dots, z))\}$ , where  $\oplus$  denotes bitwise-xor, or equivalently, we negate the variable in the  $i$ -th position iff  $\alpha_i = 1$ .

It is straightforward to see that the original 3-SAT formula is satisfiable if and only if there is an assignment to the variables that simultaneously satisfies a non zero fraction of the constraints in each of the instances.

In the above reduction, we must be able to apply the predicate to several copies of the same variable. In order to remove this restriction, we replace each instance with a collection  $\mathcal{C}$  of instances as follows: Consider an instance  $\{f(\alpha \oplus (x, \dots, x)), f(\alpha \oplus (y, \dots, y)), f(\alpha \oplus (z, \dots, z))\}$ . We add to our collection  $\mathcal{C}$ , an instance  $\{f(\alpha \oplus (a_1, \dots, a_r)), f(\alpha \oplus (b_1, \dots, b_r)), f(\alpha \oplus (c_1, \dots, c_r))\}$ , where  $a_i, b_i$  and  $c_i$  for all  $i \in [r]$ , are the fresh set of variables. Using Lemma 8.6.8, we can simultaneously-implement each constraint of the form  $x = a_i$ ,  $y = b_i$  and  $z = c_i$  using  $f$ . We add all the instances obtained from the simultaneous-implementations to the collection  $\mathcal{C}$ . Notice that, we have replaced each original instance with only  $O(1)$  many instances. Hence, we have  $O(m)$  many instances in our final construction. Thus, as in the first case, assuming ETH we deduce that detecting positivity of an  $\omega(\log n)$ -simultaneous MAX- $f$ -CSP requires time superpolynomial in  $n$ . ■

### 8.6.1 Hardness for Simultaneous MAX- $w$ -SAT

**Proposition 8.6.9** (Proposition 8.1.1 restated). *For all integers  $w \geq 4$  and  $\varepsilon > 0$ , given  $k \geq 2^{w-3}$  simultaneous instances of MAX-E $w$ -SAT that are simultaneously satisfiable, it is **NP**-hard to find a  $(7/8 + \varepsilon)$ -minimum approximation.*

*Proof.* We know that given a satisfiable MAX-E3-SAT instance, it is **NP**-hard to find an assignment that satisfies a  $(7/8 + \varepsilon)$  fraction of the constraints [Hås01]. We reduce a single MAX-E3-SAT instance to the given problem as follows : Let  $\Phi$  be an instance of MAX-E3-SAT with clauses  $\{C_i\}_{i=1}^m$  on variable set  $\{x_1, \dots, x_n\}$ . Given  $w \geq 4$ , let  $\{z_1, \dots, z_{w-3}\}$  be a fresh set of variables. For every,  $a \in \{0, 1\}^{w-3}$ , we construct a MAX-E $w$ -SAT instance with clauses  $\{C_i \vee \bigvee_{j=1}^w (z_j \oplus a_j)\}_{i=1}^m$ , where  $z_j \oplus 0 = z_j$  and  $z_j \oplus 1 = \bar{z}_j$ . It is straightforward to see that for any assignment, its value on  $\Phi$  is the same as the minimum of its value on the MAX-E $w$ -SAT instances, immediately implying the result. ■

## 8.7 Algorithm for Unweighted MAX-CUT

For simultaneous unweighted MAX-CUT instances, we can use the Goemans-Williamson SDP to obtain a slightly better approximation. The algorithm, UNWEIGHTEDMC, is described in Fig. 8.12.

Let  $V$  be the set of vertices. Our input consists of an integer  $k \geq 1$ , and  $k$  unweighted instances of MAX-CUT, specified by indicator functions  $\mathcal{W}_1, \dots, \mathcal{W}_k$  of edge set. Let  $m_\ell$  denotes the number of edges in graph  $\ell \in [k]$ . We consider these graphs as weighted graphs with all non-zero edge weights as  $\frac{1}{m_\ell}$  so that the total weight of edges of in a graph is 1. For a given subset  $S$  of vertices, we say an edge is *active* if at least one of its endpoints is in  $V \setminus S$ .

### 8.7.1 Analysis of SIM-UNWEIGHTEDMC

For analysing the algorithm SIM-UNWEIGHTEDMC, we need the following lemma that is proven by combining SDP rounding for 2-SAT from [CMM06] with a Markov argument. A proof is included in Section 8.8 for completeness.

**Lemma 8.7.1.** *For  $k$  simultaneous instances of any MAX-2-CSP such that there exists an assignment which satisfies a  $1 - \varepsilon$  weight of the constraints in each of the instances, there is an efficient algorithm that, for  $n$  large enough, given an optimal partial assignment  $h$  to a subset of variables, returns a full assignment which is consistent with*

**Input:**  $k$  unweighted instances of MAX-CUT  $\mathcal{W}_1, \dots, \mathcal{W}_k$  on the vertex set  $V$ .

**Output:** A cut of  $V$ .

1. Set  $\varepsilon \stackrel{\text{def}}{=} \frac{1}{1600 \cdot c_0^2 k^2}$ ,  $t = \frac{100k}{\varepsilon^2}$ ,  $S = \emptyset$ ,  $D = \emptyset$  ( $c_0$  is the constant from Lemma 8.7.1).
2. If every graph has more than  $t$  edges, then go to Step 4..
3. Repeat until there is no  $\ell \in [k] \setminus D$  such that the instance  $\mathcal{W}_\ell$  has less than  $t^{3^{|D|}}$  active edges given  $S$ .
  - (a) Let  $\ell \in [k]$  be an instance with the least number of edges active edges given  $S$ .
  - (b) Add all the endpoints of the edge set of instance  $\mathcal{W}_\ell$  into set  $S$ .
  - (c)  $D \leftarrow D \cup \ell$
4. For each partial assignment  $h : S \rightarrow \{0, 1\}$  (If  $S = \emptyset$  then do the following steps without considering partial assignment  $h$ )
  - (a) Run the SDP algorithm for instances in  $[k]$  given by Lemma 8.7.1 with  $h$  as a partial assignment. Let  $h_1$  be the assignment returned by the algorithm. (Note  $h_1|_S = h$ )
  - (b) Define  $g : V \setminus S \rightarrow \{0, 1\}$  by independently sampling  $g(v) \in \{0, 1\}$  with  $\mathbf{E}[g(v)] = 1/2$ , for each  $v \in V \setminus S$ . In this case the cut is given by an assignment  $h \cup g$ .
  - (c) Let  $\text{out}_h$  be the better of the two solutions ( $h_1$  and  $h \cup g$ ).
5. Output the largest  $\text{out}_h$  seen.

Figure 8.12: Algorithm SIM-UNWEIGHTEDMC for approximating unweighted simultaneous MAX-CUT

$h$  and simultaneously satisfies at least  $1 - c_0 k \sqrt{\varepsilon}$  (for an absolute constant  $c_0$ ) fraction of the constraints in each instance with probability 0.9.

Let  $S^*, D^*$  denote the set  $S$  and  $D$  that we get at the end of step 3 of the algorithm SIM-UNWEIGHTEDMC. Let  $f^* : V \rightarrow \{0, 1\}$  denote the optimal assignment and let  $h^* = f^*|_{S^*}$ .

**Theorem 8.7.2.** *For large enough  $n$ , given  $k$  simultaneous unweighted MAX-CUT instances on  $n$  vertices, the algorithm SIM-UNWEIGHTEDMC returns computes a  $(\frac{1}{2} + \Omega(\frac{1}{k^2}))$ -minimum approximate solution with probability at least 0.9. The running time is  $2^{2^{O(k)}}$ .  $\text{poly}(n)$ .*

*Proof.* We will analyze the approximation guarantee of the algorithm when the optimal partial assignment  $h^*$  to the variables  $S^*$  is picked for  $h$  in Step 4. of the algorithm. Note that Step 4.a and 4.b maintain the assignment to the set  $S^*$  given in Step 4. Hence, for all instances  $\ell \in D^*$ , we essentially get the optimal cut value  $\text{val}(f^*, \mathcal{W}_\ell)$ . We will analyze the effect of rounding done in Step 4.a and 4.b on instances in  $[k] \setminus D^*$  for a partial assignment  $h^*$  to  $S^*$ . Since we are taking the best of the two roundings, it is enough to show the claimed guarantee for at least one of these two steps.

Let OPT be the value of optimal solution for a given set of instances  $[k]$ . We consider two cases depending on the value of this optimal solution.

1.  $\text{OPT} \geq (1 - \varepsilon)$ : In this case, we show that the cut returned in Step 4.a is good with high probability.

Since the OPT is at least  $(1 - \varepsilon)$ , and  $h^*$  is an optimal partial assignment, we can apply Lemma 8.7.1 such that with probability at least 0.9 we get a cut of value at least  $(1 - 10c_0 k \cdot \sqrt{\varepsilon})$  for all graphs  $\ell \in [k] \setminus D^*$ , for some constant  $c_0$ . In this case, the approximation guarantee is at least :

$$(1 - 10c_0 k \cdot \sqrt{\varepsilon}) \geq \frac{3}{4}.$$

2.  $\text{OPT} < (1 - \varepsilon)$ : In this case, we show that the cut returned in Step 4.b gives the claimed approximation guarantee with high probability.

Fix a graph  $\ell \in [k] \setminus D^*$ , if any. Let  $m_\ell$  be the number of edges in this graph. We know that  $m_\ell \geq t^{3|D^*|}$  and also  $|S^*| \leq 4t^{3|D^*|-1}$ . Let  $Y_\ell$  be a random variable defined as

$$Y_\ell \stackrel{\text{def}}{=} \text{val}(h^* \cup g, \mathcal{W}_\ell),$$

that specifies the fraction of *total* edges that are cut by assignment  $h^* \cup g$  where  $g$  is a random partition  $g$  of a vertex set  $V \setminus S^*$ . The number of edges of graph  $\ell$  that are not active given  $S^*$  is at most  $1/2 \cdot |S^*|^2$ . If  $|D^*| = 0$ , we know that all the edges in graph  $\ell$  are active. Otherwise, using the bounds on  $m_\ell$  and  $|S^*|$ , we get that at least a  $(1 - 1/t)$  fraction of the total edges are active given  $S^*$ . This implies that for uniformly random partition  $g$ ,

$$\mathbf{E}_g[Y_\ell] \geq \frac{1}{2} \sum_{\substack{C \in \mathcal{C} \\ C \in \text{Active}(S^*)}} \mathcal{W}_\ell(C) \geq 1/2(1 - 1/t).$$

We now analyze the variance of a random variable  $Y_\ell$  under uniformly random assignment  $g : V \setminus S^* \rightarrow \{0, 1\}$ .

$$\text{Var}_g[Y_\ell] = \sum_{C_1, C_2 \in \text{Active}(S^*)} \mathcal{W}(C_1)\mathcal{W}(C_2) \cdot \left( \frac{\mathbf{E}[C_1(h^* \cup g)C_2(h^* \cup g)] - \mathbf{E}[C_1(h^* \cup g)]\mathbf{E}[C_2(h^* \cup g)]}{\mathbf{E}[C_1(h^* \cup g)]\mathbf{E}[C_2(h^* \cup g)]} \right).$$

The term in the above summation is *zero* unless we have either  $C_1 = C_2$  (in which case we know  $\mathbf{E}[C_1(h^* \cup g)C_2(h^* \cup g)] - \mathbf{E}[C_1(h^* \cup g)]\mathbf{E}[C_2(h^* \cup g)] = 1/4$ ) or when the edges  $C_1$  and  $C_2$  have a common endpoint in  $V \setminus S^*$  and the other endpoint in  $S^*$  (in this case  $\mathbf{E}[C_1(h^* \cup g)C_2(h^* \cup g)] - \mathbf{E}[C_1(h^* \cup g)]\mathbf{E}[C_2(h^* \cup g)] \leq 1/4$ ). For  $v \in V \setminus S^*$ , let  $\kappa_v$  be the set of edges whose one endpoint is  $v$  and other endpoint in  $S^*$ . Thus,

$$\begin{aligned} \text{Var}_g[Y_\ell] &\leq \frac{1}{4} \sum_{C \in \text{Active}(S^*)} \mathcal{W}(C)^2 + \frac{1}{4} \sum_{v \in V \setminus S^*} \sum_{C_1, C_2 \in \kappa_v} \mathcal{W}(C_1)\mathcal{W}(C_2) \\ &= \frac{1}{4m_\ell} + \frac{1}{4} \frac{1}{m_\ell^2} \sum_{v \in V \setminus S^*} |\kappa_v|^2 \\ &\leq \frac{1}{4m_\ell} + \max_{v \in V \setminus S^*} |\kappa_v| \cdot \frac{1}{4} \frac{1}{m_\ell^2} \sum_{v \in V \setminus S^*} |\kappa_v| \\ &\leq \frac{1}{4m_\ell} + |S^*| \cdot \frac{1}{4} \frac{1}{m_\ell^2} \cdot m_\ell \\ &\leq \frac{1}{4m_\ell} + \frac{1}{4} \frac{|S^*|}{m_\ell} \leq \frac{1}{4t^{3|D^*|}} + \frac{1}{4} \frac{|S^*|}{t^{3|D^*|}} \leq \frac{1}{2t}. \end{aligned}$$

Hence, by Chebyshev's Inequality, we have

$$\Pr \left[ Y_\ell < \frac{1}{2} \cdot (1 - \varepsilon_0 - 1/t) \right] \leq \frac{4 \mathbf{Var}_g[Y_\ell]}{\varepsilon_0^2} \leq \frac{4 \cdot 1/2t}{\varepsilon_0^2} \leq \frac{2}{\varepsilon_0^2 t}.$$

By a union bound, with probability at least  $1 - \frac{2k}{\varepsilon_0^2 t}$ , we get a simultaneous cut of value at least  $\frac{1}{2} \cdot (1 - \varepsilon_0 - 1/t)$  for all  $\ell \in [k] \setminus D^*$ . If we take  $\varepsilon_0 = \frac{\sqrt{20k}}{\sqrt{t}}$ , then with probability at least 0.9 we get a cut of value at least  $\frac{1}{2} \cdot (1 - \varepsilon_0 - 1/t)$  for all  $\ell \in [k] \setminus D^*$ .

In this case, the approximation guarantee is at least

$$\frac{\frac{1}{2} \cdot (1 - \varepsilon_0 - 1/t)}{\left(1 - \frac{1}{(40c_0k)^2}\right)} = \left(\frac{1}{2} + \Omega\left(\frac{1}{k^2}\right)\right).$$

■

## 8.8 SDP for Simultaneous Instances

In this section, we study Semidefinite Programming (SDP) relaxations for simultaneous MAX-2-CSP instances.

### 8.8.1 Integrality gaps for Simultaneous MAX-CUT SDP

In this section, we show the integrality gaps associated with the natural SDP of *minimum approximation* problem for  $k$ -fold simultaneous MAX-CUT.

Suppose we have  $k$  simultaneous MAX-CUT instances on the set of vertices  $V = \{x_1, \dots, x_n\}$ , specified by the associated weight functions  $\mathcal{W}_1, \dots, \mathcal{W}_k$ . As before, let  $\mathcal{C}$  denotes the set of all possible edges on  $V$ . We assume that for each  $\ell \in [k]$ ,  $\sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) = 1$ . Following Goemans and Williamson [GW95], the semi-definite programming relaxation for such an instance is described in Fig. 8.13. We now prove the following claims about integrality gap for the above SDP.

**Claim 8.8.1.** *For weighted instances, the SDP for minimum approximation of simultaneous MAX-CUT does not have any constant integrality gap.*

*Proof.* Consider 3 simultaneous instances such that all but a tiny fraction of the weight of instance  $i$  is on edge  $i$  of a 3-cycle. Clearly, no cut can simultaneously cut all the three edges in the three cycle, and hence the optimum is tiny. However, for the simultaneous

$$\begin{array}{ll}
& \text{maximize} \quad t \\
\text{s.t.} \quad & \sum_{\substack{C \in \mathcal{C} \\ C=(x_i, x_j)}} \frac{1}{2} \cdot \mathcal{W}_\ell(C) \cdot (1 - \langle v_i, v_j \rangle) \geq t \quad \forall \ell \in [k] \\
& \|v_i\|^2 = 1 \quad \text{for } i = 1, \dots, n
\end{array}$$

Figure 8.13: Semidefinite Program (SDP) for minimum approximation Simultaneous MAX-CUT

SDP, a vector solution that assigns to the three vertices of the cycle three vectors such that  $\langle v_i, v_j \rangle = -1/2$  for  $i \neq j$  gives a constant objective value for all three instances. ■

**Claim 8.8.2.** *For every fixed  $k$ , there exists  $k$ -instances of MAX-CUT where the SDP relaxation has value  $1 - \Omega\left(\frac{1}{k^2}\right)$ , while the maximum simultaneous cut has value only  $\frac{1}{2}$ . Moreover, the random hyperplane rounding for a good vector solution for this instance, returns a simultaneous cut of value 0.*

*Proof.* Let  $k$  be odd. We define  $k$  graphs on  $kn$  vertices. Partition the vertex set into  $S_0, S_1, \dots, S_{k-1}$ , each of size  $n$ . Graph  $G_i$  has only edges  $(x, y)$  such that  $x \in S_i$  and  $y \in S_{(i+1) \bmod k}$ , each of weight  $1/n^2$ . The optimal cut must contain exactly half the number of vertices from each partition, giving a simultaneous cut value of  $1/2$ . Whereas, the following SDP vectors achieve a simultaneous objective of  $(1 - O(\frac{1}{k^2}))$ : For all vertices in  $S_i$ , we assign the vector  $(\cos \frac{i}{k}\pi, \sin \frac{i}{k}\pi)$ . It is straightforward to see that applying the hyperplane rounding algorithm to this vector solution gives (with probability 1) a simultaneous cut value of 0. ■

### 8.8.2 SDP for Simultaneous MAX-CSP

For MAX-CSP, we will be interested in the regime where the optimum assignment satisfies at least a  $(1 - \varepsilon)$  fraction of the constraints in each of the instances.

Given a MAX-2-CSP instance, we use the standard reduction to transform it into a MAX-2-SAT instance: We reduce each constraint of the 2-CSP instance with a set of

at most 4 2-SAT constraints such that for any fixed assignment, the 2-CSP constraint is satisfied iff all the 2-SAT constraints are satisfied, and if the 2-CSP constraint is not satisfied, then at least one of the 2-SAT constraint is not satisfied. *e.g.* We replace  $x_1 \wedge x_2$  with  $x_1 \vee x_2, \overline{x_1} \vee x_2$ , and  $x_1 \vee \overline{x_2}$ . Similarly, we replace  $x_1 \neq x_2$  with  $x_1 \vee x_2$  and  $\overline{x_1} \vee \overline{x_2}$ . We distribute the weight of the 2-CSP constraint equally amongst the 2-SAT constraints.

Given  $k$  simultaneous MAX-2-CSP instances, we apply the above reduction to each of the instances to obtain  $k$  simultaneous MAX-2-SAT instances. The above transformation guarantees the following:

- **Completeness** If there was an assignment of variables that simultaneously satisfied all the constraints in each of the MAX-2-CSP instances, then the same assignment satisfies all the constraints in each of the MAX-2-SAT instances.
- **Soundness** If no assignment of variables simultaneously satisfied more than  $(1-\varepsilon)$  weighted fraction of the constraints in each of the MAX-2-CSP instances, then no assignment simultaneously satisfies more than  $(1 - \varepsilon/4)$  weighted fraction of the constraints in each of the MAX-2-SAT instances.

From now on, we will assume that we have  $k$  simultaneous MAX-2-SAT instances on the set of variables  $\{x_1, \dots, x_n\}$ , specified by the associated weight functions  $\mathcal{W}_1, \dots, \mathcal{W}_k$ . As before  $\mathcal{C}$  denotes the set of all possible 2-SAT constraints on  $V$ . We assume that for each  $\ell \in [k]$ ,  $\sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) = 1$ . Following Charikar *et al.* [CMM06], the semi-definite programming relaxation for such an instance is described in Fig. 8.14.

For convenience, we replace each negation  $\overline{x_i}$  with a new variable  $x_{-i}$ , that is equal to  $\overline{x_1}$  by definition. For each variable  $x_i \in V$ , the SDP relaxation will have a vector  $v_i$ . We define  $v_{-i} = -v_i$ . We will also have a unit vector  $v_0$  that is intended to represent the value 1. For a subset  $S$  of variables and a partial assignment  $h : S \rightarrow \{0, 1\}$ , we write the following SDP for the simultaneous MAX-2-SAT optimization problem:

$$\begin{array}{ll}
& \text{maximize} \quad t \\
\text{s.t.} & \sum_{\substack{C \in \mathcal{C} \\ C = x_i \vee x_j}} \mathcal{W}_\ell(C) \cdot \left( \|v_0\|^2 - \frac{1}{4} \langle v_i - v_0, v_j - v_0 \rangle \right) \geq t \quad \forall \ell \in [k] \\
& \langle v_i - v_0, v_j - v_0 \rangle \geq 0 \quad \forall \text{ constraints } x_i \vee x_j \\
& \|v_i\|^2 = 1 \quad \text{for } i = -n, \dots, n \\
& v_i = -v_{-i} \quad \text{for } i = 1, \dots, n \\
& v_i = v_0 \quad \forall i \in S \text{ s.t. } h(i) = 1 \\
& v_j = -v_0 \quad \forall j \in S \text{ s.t. } h(j) = 0
\end{array}$$

Figure 8.14: Semidefinite Program (SDP) with a partial assignment  $h : S \rightarrow \{0, 1\}$  for Simultaneous MAX-2-SAT

We first observe that for an optimal partial assignment  $h$ , the optimum of the above SDP is at least the optimum of the simultaneous maximization problem, by picking the solution  $v_i = v_0$  if  $x_i = \text{TRUE}$ , and  $v_i = -v_0$  otherwise. For this vector solution, we have  $1/4 \cdot \left( \|v_0\|^2 - \langle v_i - v_0, v_j - v_0 \rangle \right) = 1$  if the constraint  $x_1 \vee x_2$  is satisfied by the assignment, and 0 otherwise. Since  $\sum_{C \in \mathcal{C}} \mathcal{W}_\ell(C) = 1$  for all  $\ell$ , the optimum of the SDP lies between 0 and 1.

Note that the rounding algorithm defined in [CMM06] does not depend on the structure of the vectors in the SDP solution. Thus, the following theorem that was proved without a partial assignment in [CMM06] also applies to above SDP.

**Theorem 8.8.3.** *Given a single MAX-2-SAT instance ( $k = 1$ ), there is an efficient randomized rounding algorithm such that, if the optimum of the above SDP is  $1 - \varepsilon$ , for  $n$  large enough, it returns an assignment such that the weight of the constraints satisfied is at least  $1 - O(\sqrt{\varepsilon})$  in expectation.*

Now, using Markov's inequality, we can prove the following corollary.

**Corollary 8.8.4.** *For  $k$  simultaneous instances of MAX-2-SAT, there is an efficient randomized rounding algorithm such that if the optimum of the above SDP is  $1 - \varepsilon$ , for  $n$*

large enough, it returns an assignment that simultaneously satisfies at least  $1 - O(k\sqrt{\varepsilon})$  fraction of the constraints in each instance with probability 0.9.

*Proof.* We use the rounding algorithm given by Theorem 8.8.3 to round a solution to the SDP for the  $k$  simultaneous instances that achieves an objective value of  $1 - \varepsilon$ . Observe that this solution is also a solution for the SDP for each of the instances by itself with the same objective value. Thus, by Theorem 8.8.3, for each of the instances, we are guaranteed to find an assignment such that the weight of the constraints satisfied is at least  $1 - c_0\varepsilon$  in expectation, for some constant  $c > 0$ . Since, for any instance, the maximum weight an assignment can satisfy is at most 1, with probability at least  $1 - 1/10 \cdot k$  for each instance, we get an assignment such that the weight of the constraints satisfied is at least  $1 - 10ck \cdot \sqrt{\varepsilon}$ . Thus, applying a union bound, with probability at least  $1 - 1/10$ , we obtain an assignment such that the weight of the satisfied constraints in *all* the  $k$  instances is at least  $1 - 10ck \cdot \sqrt{\varepsilon}$ . ■

Combining the above corollary with the reduction from any MAX-2-CSP to MAX-2-SAT, and the completeness of the SDP, we get a proof of Lemma 8.7.1.

### Concentration inequalities

**Lemma 8.8.5** (McDiarmid's Inequality). *Let  $X_1, X_2, \dots, X_m$  be independent random variables, with  $X_i$  taking values in a set  $A_i$  for each  $i$ . Let  $\text{score} : \prod A_i \rightarrow \mathbb{R}$  be a function which satisfies:*

$$|\text{score}(x) - \text{score}(x')| \leq \alpha_i$$

*whenever the vector  $x$  and  $x'$  differ only in the  $i$ -th co-ordinate. Then for any  $t > 0$*

$$\Pr[|\text{score}(X_1, X_2, \dots, X_m) - \mathbf{E}[\text{score}(X_1, X_2, \dots, X_m)]| \geq t] \leq 2 \exp\left(\frac{-2t^2}{\sum_i \alpha_i^2}\right)$$

### The need for perturbing OPT

We construct 2 simultaneous instances of MAX-1-SAT. Suppose the algorithm will picks at most  $r$  influential variables. Construct the two instances on  $r + 1$  variables, with the weights of the variables decreasing geometrically, say, with ratio  $1/3$ . The

first instance requires all of them to be TRUE, where as the second instance requires all of them to be FALSE. Under a reasonable definition of “influential variables”, the only variable left behind should be the vertex with the least weight. We consider the Pareto optimal solution that assigns TRUE to all but the last variable. If we pick the optimal assignment for the influential variables, and then randomly assign the rest of the variables, with probability  $1/2$ , we get zero on the second instance.

## Chapter 9

### Simultaneous Max-Cut

#### 9.1 Introduction

In this chapter, we give near-optimal approximation algorithms for the simultaneous MAX-CUT problem. Here we are given a collection of weighted graphs  $G_1, G_2, \dots, G_k$  on the same vertex set  $V$  of size  $n$ . Our goal is to find a partition of the vertex set  $V$  into two parts, such that in *every* graph, the total weight of edges going between the two parts is large. The  $k = 1$  case is the classical MAX-CUT problem, and the approximability of this problem has been extensively studied [FL92, GW95, Hås01, KKMO07, MOO05, OW08]. This chapter studies the approximability of this problem for constant  $k$ .

We fix some convenient notation. Let the weighted graphs  $G_1, \dots, G_k$  be given by weight functions  $\mathcal{E}_1, \dots, \mathcal{E}_k$ , which assign to each pair in  $\binom{V}{2}$  a weight in  $[0, 1]$ . We assume that for each  $i \in [k]$ , the total weight of all edges under  $\mathcal{E}_i$  equals 1. Let  $f : V \rightarrow \{0, 1\}$  be a function, which we view as a partition of the vertex set. We define  $\text{val}(f, \mathcal{E}_i)$  to be the total weight (under  $\mathcal{E}_i$ ) of the edges cut by the partition  $f$ . Given this setup, we can formally state the notions of approximation that we consider.

- **$\alpha$ -minimum approximation:** Let  $c$  be the maximum, over all partitions  $f^* : V \rightarrow \{0, 1\}$ , of the quantity  $\min_{i \in [k]} \text{val}(f^*, \mathcal{E}_i)$ . The goal is to output an  $f : V \rightarrow \{0, 1\}$  such that  $\min_{i \in [k]} \text{val}(f, \mathcal{E}_i) \geq \alpha \cdot c$ .
- **$\alpha$ -Pareto approximation:** Let  $c_1, c_2, \dots, c_k$  be given such that there exists  $f^* : V \rightarrow \{0, 1\}$  with  $\text{val}(f^*, \mathcal{E}_i) \geq c_i$  for each  $i \in [k]$ . The goal is to output an  $f : V \rightarrow \{0, 1\}$  such that  $\text{val}(f, \mathcal{E}_i) \geq \alpha \cdot c_i$  for all  $i \in [k]$ .

For  $k = 1$ , there is a celebrated polynomial time  $\alpha_{GW} = 0.8786\dots$  factor (Pareto)

approximation algorithm by Goemans and Williamson [GW95]. This approximation is in both the minimum and Pareto senses. Furthermore, it is Unique-Games hard to achieve a better approximation factor than this [KKMO07], and the entire polynomial time “approximation curve” is also known.

For larger (but constant)  $k$ , far less is understood. Clearly, the hardness results from the  $k = 1$  case carry over, and thus it is Unique-Games hard to approximate this to a factor better than  $\alpha_{GW}$ . [ABG06] gave a polynomial time 0.439-Pareto approximation algorithm for this problem for the case  $k = 2$ . Subsequently, [BKS15] gave a polynomial time  $(1/2 - \varepsilon)$ -Pareto approximation algorithm for this problem. For the case of unweighted graphs<sup>1</sup>, [BKS15] showed that there is a polynomial time  $(1/2 + \Omega(1/k^2))$ -minimum approximation algorithm. Furthermore, [BKS15] gave a matching integrality gap of  $(1/2 + O(1/k^2))$  for a natural SDP relaxation of the minimum approximation problem.

Our main result is a polynomial time 0.8782-factor Pareto approximation algorithm for simultaneous MAX-CUT for arbitrary constant  $k$ .

**Main Theorem:** For all constant  $k$ , there is a polynomial time algorithm which computes a 0.8782-factor Pareto approximation (and hence min approximation) to the simultaneous MAX-CUT problem with  $k$  instances.

**Remark 9.1.1.** *We assume that the edge weights are lower bounded by  $\exp(-|V|^c)$  for some constant  $c > 0$ . We are interested in an algorithm which runs in time polynomial in  $|V|$  and hence it is natural to assume the edge weights are lower bounded by  $\exp(-|V|^c)$  as otherwise the bit complexity of the input will be super polynomial in  $|V|$ .*

We give a brief overview of ideas involved in our algorithm next. The main ingredients of the algorithm are: a sum-of-squares hierarchy SDP relaxation, a generalization of the [RT12], [ABG12] approach to rounding such relaxations, and some ideas from [BKS15].

---

<sup>1</sup>We call an instance of simultaneous MAX-CUT *unweighted* if for any  $i$ , all the nonzero weight edges under  $\mathcal{E}_i$  have the same weight.

## 9.2 Overview of the algorithm

We begin by considering the unweighted case; later we will discuss how to remove this restriction. One crucial observation about the unweighted case is that if there are enough number of edges in every graph, compared to  $k$ , then by second moment argument there exists a cut which cuts a constant fraction of edges from each graph. Thus, we can always assume that each target value is  $c_i = \Omega_k(1)$ , which is a constant for a constant  $k$ .

There is a natural SDP relaxation for the simultaneous MAX-CUT problem, generalizing the Goemans-Williamson SDP for the  $k = 1$  case. If we solve this SDP and round the resulting vector solution using the Goemans-Williamson hyperplane rounding procedure, this gives us a distribution of partitions of the vertex set  $V$ , such that for *each*  $i \in [k]$ , the total weight of edges cut in instance  $i$  is at least  $\alpha_{GW}$  times the corresponding SDP cut value. However, unlike in the  $k = 1$  case, this does not guarantee the existence of a single partition of  $V$  which achieves a large cut value for all the  $k$  instances simultaneously! This distinction between distributions of solutions which are good in expectation for each instance and single solutions that are simultaneously good for all instances is the heart of the difficulty in designing simultaneous approximation algorithms.

One of the basic ingredients underlying mathematical programming relaxation hierarchies for combinatorial optimization problems is the idea of expanding the search space, from the discrete space of pure assignments to the continuous space of distributions over assignments. For simultaneous approximation of MAX-CUT beyond a factor  $1/2$ , this idea alone is not enough. An example from [BKS15] shows that there are cases of simultaneous MAX-CUT on  $k$ -instances, for which there is a distribution of partitions of  $V$  cutting  $(1 - \frac{1}{k})$ -fraction of edges *in expectation* for each instance, but for which any single partition of  $V$ , there is an instance  $i \in [k]$ , such that at most  $1/2$  of the edges in instance  $i$  are cut by the partition. This is where the sum-of-squares SDP hierarchy comes in – even though it is also modeled on the idea of expanding the search space to distributions of assignments – it allows us to *condition* on partial assignments

and impose a constraint that the SDP cut value is large in expectation for each instance and for every possible conditioning on a small number of variables. This is what allows us to overcome the aforementioned obstacle.

Having formulated the SDP relaxation, we now discuss the rounding procedure. The motivating observation is this: if the rounding procedure is such that for each instance the expected cut value is large, and the cut value is concentrated around its expectation with high probability, then by a union bound, the rounding procedure will produce a cut that is simultaneously good for all instances. The rounding procedure we will use will be closely related to the Goemans-Williamson rounding (but different – it was found by computer search given various technical conditions required by the rest of the algorithm). Our algorithm now tries to improve the concentration of the cut-value produced by the rounding procedure, via a beautiful information-theoretic approach of Raghavendra-Tan [RT12]. If the cut-value for a certain instance turns out to be not concentrated under the rounding procedure, then it must be because of high correlation between many pairs of edges of that instance (more precisely, correlation between the events that the edge is cut). This in turn means that conditioning on the variables in a random edge should significantly decrease the amount of entropy of the rounded cut. Iterating this several times, and using the fact that the initial entropy is not too large, we conclude that conditioning on a small number of variables leads to good concentration for the rounding procedure. The key point is that the sum-of-squares SDP relaxation we use gives us access to a vector solution for the conditioned SDP, with the promise that the SDP cut-value (and hence the expected integral cut-value) is still large. By the concentration property and a union bound, we get a simultaneously good cut. This completes the description of the algorithm in the unweighted case.

To handle the general weighted case, we essentially need to overcome few technical obstacles. Following [BKS15], we add a preprocessing and postprocessing phase. The preprocessing phase identifies “wild” instances, i.e. those instances with an abnormally large number of high (weighted-)degree vertices (which would increase the variance of the cut value of that instance under random rounding). Then the SDP based algorithm described above is run only on the “tame” instances.

With conditioning on constantly many variables, arguably we can only manage to bring the variance down to arbitrarily small constant. Hence, in order to use second moment method to get concentration, we would need a good lower bound on the expected value of a cut given by our rounding procedure. If the graphs are weighted then it is not necessarily true that the simultaneous cut value is large for all instances. One important property of the *tame* instances we used is that they have a good simultaneous MAX-CUT value. We crucially use this property while formulating the SDP for tame instances.

Finally in the postprocessing phase, we find suitable assignment to the high degree vertices of the wild instances to ensure that those instance have a large cut value (without spoiling the large cut value of the tame instances that the SDP guaranteed) – this uses a new and much simpler perturbation argument compared to [BKS15].

This concludes the high-level description of the algorithm.

### 9.2.1 Note about the rounding procedure

We mentioned earlier that our SDP solution after conditioning on a small number of variables is rounded by a rounding algorithm similar to the Goemans-Williamson rounding algorithm, but is different. We expound upon these conditions here and compare with the previous results that used similar rounding procedure.

The SDP solution induces a consistent local distribution on every set of variables of size at most some constant  $r$ , and we define the *sdp-bias* of a variable as the bias with respect to this local distribution. For a given rounding procedure, we define the *rounding-bias* of a variable as the difference in the probability of the corresponding vertex being assigned to each side of the cut. Note that in the original hyperplane rounding of Goemans-Williamson, rounding-bias of each vertex is 0.

In the rounding procedure for the MAX-BISECTION from [RT12], the rounding-bias induced by the rounding procedure is the same as the sdp-bias. Their algorithm gave a 0.85 approximation for MAX-BISECTION, and using the same bias function for the rounding in our case gives us a 0.85 approximation for simultaneous MAX-CUT as well. The approximation factor given by [RT12] was subsequently improved in [ABG12] to

0.8776, where they used techniques to relax the restriction on the choice of the bias function, but they were constrained by the nature of the MAX-BISECTION problem and therefore had to ensure that the rounding-bias of all variables sum up to 0 to maintain the balance of the cut. As we do not want equal sized partition of the vertex set, we have more freedom in our rounding procedure with respect to rounding-bias; we only have to ensure that when the bias of a variable is high, the side of the cut it falls on is almost fixed. This helps us achieve an improved approximation factor of 0.8782. The rounding function we come up with was arrived at by trial and error method with the constraint that the rounding bias goes to  $\pm 1$  as the sdp-bias goes to  $\pm 1$ .

The approximation ratio for our rounding procedure is proved by a computer assisted prover, where the techniques are similar to the ones used in [Sjo09] and [ABG12].

### 9.2.2 Other related work

The simultaneous MAX-CUT problem is a special case of the simultaneous approximation problem for general constraint satisfaction problems. This general problem was studied in [BKS15], where it was shown that there is a polynomial time constant factor Pareto approximation algorithm for every simultaneous CSP (with approximation factor independent of  $k$ ). The algorithm there was based on understanding the structure of CSP instances whose value is highly concentrated under a random assignment to the variables, in addition to linear-programming. It was also observed that there are CSPs for which the best polynomial time approximation factor for the simultaneous version (with  $k > 1$ ) is *different* from the best polynomial time approximation factor achievable in the standard  $k = 1$  case (assuming  $P \neq NP$ ). This makes the study of simultaneous approximation factors very interesting.

The simultaneous MAXSAT problem was studied in [GRW11], where a  $1/2$ -Pareto approximation algorithm was given. For bounded width MAXSAT, the approximation factor was improved to  $(3/4 - \varepsilon)$  in [BKS15].

### 9.3 Preliminaries

#### 9.3.1 Simultaneous MAX-CUT

Let  $V$  be a vertex set with  $|V| = n$ . We use the set  $[n]$  for the vertex set  $V$  for convenience. We are given  $k$  graphs  $G_1, \dots, G_k$  on the vertex set  $V$ . Let  $\mathcal{E}_\ell : [n] \times [n] \rightarrow \mathbb{R}^{\geq 0}$  denote the edge weights of graph  $G_\ell$  where the edge weights are normalized such that total weight of edges in each instance is 1. We'll use  $\mathcal{E}_\ell$  to denote the edge set of graph  $G_\ell$  and also the distribution of the edges based on the weights. For each instance  $\ell$ , we are given a target cut value  $c_\ell$  that we would like to achieve.

**Definition 9.3.1** ( $\alpha$ -approximation of simultaneous MAX-CUT). *A partition  $(U, \bar{U})$  of  $V$  is said to be an  $\alpha$ -approximation if for each instance  $G_\ell$ ,*

$$\text{Cut}_\ell(U, \bar{U}) \geq \alpha \cdot c_\ell$$

#### 9.3.2 Information Theory

In this section, we define and state some facts about entropy and mutual information between random variables.

**Definition 9.3.2** (Entropy). *Let  $X$  be a random variable taking values in  $[q]$  then, entropy of  $X$  is defined as:*

$$H(X) := \sum_{i \in [q]} \Pr[X = i] \log \frac{1}{\Pr[X = i]}$$

**Definition 9.3.3** (Conditional Entropy). *Let  $X, Y$  be jointly distributed random variable taking values in  $[q]$  then, the conditional entropy of  $X$  conditioned on  $Y$  is defined as:*

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

The following observations can be made about entropy of a collection of random variables.

Entropy of a collection of random variables cannot exceed the sum of their entropies.

**Fact 9.3.4.**  $H(X_1, X_1, \dots, X_n) \leq \sum_{i=1}^n H(X_i)$

Entropy never decreases on adding more random variables to the collection.

**Fact 9.3.5.**  $H(X_1, X_2|Y) \geq H(X_1|Y)$

Conditioning can only decrease the entropy.

**Fact 9.3.6.**  $H(X|Y) - H(X|Y, Z) \geq 0$

**Definition 9.3.7** (Mutual Information). *Let  $X, Y$  be jointly distributed random variable taking values in  $[q]$  then, the mutual information between  $X$  and  $Y$  is defined as:*

$$I(X; Y) := \sum_{i,j \in [q]} \Pr[X = i, Y = j] \log \frac{\Pr[X = i, Y = j]}{\Pr[X = i] \Pr[Y = j]}$$

**Theorem 9.3.8.** (Data Processing Inequality) *If  $X, Y, W, Z$  are random variables such that  $X$  is fully-determined by  $W$  and  $Y$  is fully-determined by  $Z$ , then*

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

## 9.4 Algorithm for simultaneous weighted MAX-CUT

In this section, we give our approximation algorithm for simultaneous weighted MAX-CUT and the analysis.

### 9.4.1 Notation

We use the same notation as in [BKS15], which we reproduce here. Let  $\mathcal{E} = \binom{V}{2}$  be the set of all possible edges. Given an edge  $e$  and a vertex  $v$ , we say  $v \in e$  if  $v$  appears in the edge  $e$ . For an edge  $e$ , let  $e_1, e_2$  denote the endpoints of  $e$  (arbitrary order). Let  $f : V \rightarrow \{0, 1\}$  be an assignment. For an edge  $e \in \mathcal{E}$ , define  $e(f)$  to be 1 if the edge  $e$  is cut by the assignment  $f$ , and define  $e(f) = 0$  otherwise. Note that an assignment cuts an edge if it assigns different values to the end points. Then, we have the following expression for the cut value of the assignment:

$$\text{val}(f, \mathcal{E}) \stackrel{\text{def}}{=} \sum_{e \in \mathcal{E}} \mathcal{E}(e) \cdot e(f).$$

A partial assignment  $h : S \rightarrow \{0, 1\}$  is an assignment to  $S$  where  $S \subseteq V$ . We say an edge is *active* with respect to  $S$  if at least one of the end vertices is not in  $S$ . We

denote by  $\text{Active}(S)$  the set of all edges which are active with respect to  $S$ . For two edges  $e, e' \in \mathcal{E}$ , we say  $e \sim_S e'$  if they share a vertex that is contained in  $V \setminus S$ . Note that if  $e \sim_S e'$ , then  $e, e'$  are both in  $\text{Active}(S)$ . Let  $\text{actdist}_S(\ell)$  denotes the distribution  $\mathcal{E}_\ell$  conditioned on an edge being active w.r.t  $S$ .

Define the active degree given  $S$  of a variable  $v \in V \setminus S$  for instance  $\ell$  by:

$$\text{activedegree}_S(v, \ell) \stackrel{\text{def}}{=} \sum_{e \in \text{Active}(S), e \ni v} \mathcal{E}_\ell(e).$$

We then define the active degree of the whole instance  $\ell$  given  $S$ :

$$\text{activedegree}_S(\ell) \stackrel{\text{def}}{=} \sum_{v \in V \setminus S} \text{activedegree}_S(v, \ell).$$

Note that we count weight of an active edge in  $\text{activedegree}_S(\ell)$  at most twice. For a partial assignment  $h : S \rightarrow \{0, 1\}$ , we define

$$\text{val}(h, \mathcal{E}_\ell) \stackrel{\text{def}}{=} \sum_{\substack{e \in \mathcal{E} \\ e \notin \text{Active}(S)}} \mathcal{E}_\ell(e) \cdot e(h)$$

which is the total weight of non-active edges cut by the partial assignment  $h$ . Thus, for an assignment  $g : V \setminus S \rightarrow \{0, 1\}$ , to the remaining set of variables, we have the equality:

$$\text{val}(h \cup g, \mathcal{E}_\ell) - \text{val}(h, \mathcal{E}_\ell) = \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e) \cdot e(h \cup g).$$

#### 9.4.2 Algorithm

In [Figure 9.1](#), we give the algorithm for Simultaneous MAX-CUT. The input to the algorithm consists of an integer  $k \geq 1$ ,  $\varepsilon \in (0, 2/5]$ ,  $k$  instances of MAX-CUT, specified by weight functions  $\mathcal{E}_1, \dots, \mathcal{E}_k$ , and  $k$  target objective values  $c_1, \dots, c_k$ .

#### 9.4.3 Analysis of the Algorithm

The algorithm broadly proceeds in 3 sections, the pre-processing step, the SDP step and the post processing step. The pre-processing step consists of identifying a small subset  $S \subseteq V$  carefully. We then attempt all assignments to vertices in  $S$  by brute force iteratively and use SDP with the partial assignment followed by a rounding to assign

**Input:**  $k$  instances of MAX-CUT, with weights defined by  $\mathcal{E}_1, \dots, \mathcal{E}_k$  on the set of variables  $V$ , target objective values  $c_1, \dots, c_k$ , and  $\varepsilon \in (0, 2/5]$ .

**Output:** An assignment to  $V$ .

**Parameters:**  $\delta_0 = \frac{1}{10k}$ ,  $\varepsilon_0 = \frac{\varepsilon}{2}$ ,  $\gamma = \frac{\varepsilon_0^2 \delta_0}{16}$ ,  $t = \frac{2k}{\gamma} \cdot \log\left(\frac{11}{\gamma}\right)$ ,  $\tau = \varepsilon$ .

**Pre-processing:**

1. Initialize  $S \leftarrow \emptyset$ .
2. For each instance  $\ell \in [k]$ , initialize  $\text{count}_\ell \leftarrow 0$  and  $\text{flag}_\ell \leftarrow \text{TRUE}$ .
3. Repeat the following until for every  $\ell \in [k]$ , either  $\text{flag}_\ell = \text{FALSE}$  or  $\text{count}_\ell = t$ :
  - (a) For each  $\ell \in [k]$ , compute  $\text{Uvar}_\ell = \sum_{e \sim_S e'} \mathcal{E}_\ell(e) \mathcal{E}_\ell(e')$ .
  - (b) For each  $\ell \in [k]$  compute  $\text{Lmean}_\ell \stackrel{\text{def}}{=} \tau \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e)$ .
  - (c) For each  $\ell \in [k]$ , if  $\text{Uvar}_\ell \geq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2$ , then set  $\text{flag}_\ell = \text{TRUE}$ , else set  $\text{flag}_\ell = \text{FALSE}$ .
  - (d) Choose any  $\ell \in [k]$ , such that  $\text{count}_\ell < t$  AND  $\text{flag}_\ell = \text{TRUE}$  (if any):
    - i. Find  $v \in V$  such that  $\text{activedegree}_S(v, \ell) \geq \gamma \cdot \text{activedegree}_S(\ell)$ .
    - ii. Set  $S \leftarrow S \cup \{v\}$ . We say that  $v$  was brought into  $S$  because of instance  $\ell$ .
    - iii. Set  $\text{count}_\ell \leftarrow \text{count}_\ell + 1$ .
4. After exiting the loop, all instances for which  $\text{flag}_\ell$  is set to FALSE are labelled “low-variance” instances and all instances for which  $\text{count}_\ell = t$  are labelled “high-variance” instances.

**Main algorithm:**

- 5 For each possible partial fixing  $h : S \rightarrow \{0, 1\}$  do the following
  - (a) Run the Lasserre version of  $\text{SDP}^*(h)$  mentioned in Figure 9.2 on the low variance instances. (Refer Section 9.4.3)
  - (b) Follow the procedure in Figure 9.4 to make the solution locally independent. (Refer Section 9.4.3)
  - (c) Round the solution based on the rounding procedure described in Figure 9.5 to get a partial assignment  $g : V \setminus S \rightarrow \{0, 1\}$ . (Refer Section 9.4.3)
  - (d) For every assignment  $h' : S \rightarrow \{0, 1\}$ , compute  $\min_{\ell} \frac{\text{val}(h' \cup g, \mathcal{E}_{\ell})}{c_{\ell}}$  and return the assignment  $h' \cup g$  that maximizes this. (**Post-processing step**)

Figure 9.1: Algorithm ALG-SIM-MAXCUT for approximating weighted simultaneous MAX-CUT

vertices in  $V \setminus S$ . The post-processing step involves perturbing the assignments to the vertices in  $S$ , the need for which is explained in detail in [Section 9.4.3](#).

In what follows, we stick to the following notation. Let  $S^*$  denote the final set  $S$  that we get at the end of [Step 3](#). of ALG-SIM-MAXCUT. Let  $f^* : V \rightarrow \{0, 1\}$  be the assignment that achieves  $\text{val}(f^*, \mathcal{E}_\ell) \geq c_\ell$  for all  $\ell \in [k]$  and  $h^*$  be the restriction of  $f^*$  to the set  $S^*$ .

### Pre-processing: Low and High variance instances

**Definition 9.4.1** ( $\tau$ -smooth distribution). *A distribution  $D$  on  $\{0, 1\}$  is called  $\tau$ -smooth if*

$$\Pr_{x \sim D}[x = 1] \geq \tau, \quad \Pr_{x \sim D}[x = 0] \geq \tau.$$

Let  $h : S \rightarrow \{0, 1\}$  be an arbitrary partial assignment to the vertices in  $S$ . Let  $g : V \setminus S \rightarrow \{0, 1\}$  be the random assignment such that each of the marginals  $g(v)$  is  $\tau$ -smooth. For an instance  $\ell$ , define the random variable

$$Y_\ell \stackrel{\text{def}}{=} \text{val}(h \cup g, \mathcal{E}_\ell) - \text{val}(h, \mathcal{E}_\ell) = \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e) \cdot e(h \cup g).$$

$Y_\ell$  measures the total active edge weight cut by the assignment in the instance  $\ell$ .

Consider the two quantities defined in [Step 3](#). of the algorithm. They depend only on  $S$  (and importantly, not on  $h$ ), which will be useful in controlling the expectation and variance of  $Y_\ell$ . The first quantity is an upper bound on  $\text{Var}[Y_\ell]$ :

$$\text{Uvar}_\ell \stackrel{\text{def}}{=} \sum_{e \sim_S e'} \mathcal{E}_\ell(e) \mathcal{E}_\ell(e')$$

The second quantity is a lower bound on  $\mathbf{E}[Y_\ell]$ :

$$\text{Lmean}_\ell \stackrel{\text{def}}{=} \tau \cdot \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e)$$

**Lemma 9.4.2.** *Let  $S \subseteq V$  be a subset of vertices and  $h : S \rightarrow \{0, 1\}$  be an arbitrary partial assignment to  $S$ . Let  $Y_\ell, \text{Uvar}_\ell, \text{Lmean}_\ell$  be as above.*

1. *If  $\text{Uvar}_\ell \leq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2$ , then  $\Pr[Y_\ell < (1 - \varepsilon_0) \mathbf{E}[Y_\ell]] < \delta_0$ .*

2. If  $\text{Uvar}_\ell \geq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2$ , then there exists  $v \in V \setminus S$  such that

$$\text{activedegree}_S(v, \ell) \geq \frac{1}{4} \tau^2 \varepsilon_0^2 \delta_0 \cdot \text{activedegree}_S(\ell).$$

We defer the formal proof to [Section 9.5](#). The first part is a simple application of the Chebyshev inequality. For the second part, we use the assumption that  $\text{Uvar}_\ell$  is large, to deduce that there exists an edge  $e$  such that the total weight of edges adjacent to the vertex/vertices in  $e$  that belong to  $V \setminus S$ , *i.e.*,  $\sum_{e_2 \sim_S e} \mathcal{E}(e_2)$ , is large. It then follows that at least one variable  $v \in e$  must have large active degree given  $S$ .

The above lemma ([Lemma 9.4.2](#)) ensures that [Step 3.\(d\)i](#) in the algorithm always succeeds in finding a variable  $v$ . Next, we note that [Step 3.](#) always terminates. Indeed, whenever we find an instance  $\ell \in [k]$  in [Step 3.d](#) such that  $\text{count}_\ell < t$  and  $\text{flag}_\ell = \text{TRUE}$ , we increment  $\text{count}_\ell$ . This can happen only  $tk$  times before the condition  $\text{count}_\ell < t$  fails for all  $\ell \in [k]$ . Thus the loop must terminate within  $tk$  iterations.

To analyze the approximation guarantee of the algorithm, we classify instances according to how many vertices were brought into  $S^*$  because of them.

**Definition 9.4.3** (Low and high variance instances). *At the completion of [Step 3.d](#) in Algorithm ALG-SIM-MAXCUT, if  $\ell \in [k]$  satisfies  $\text{count}_\ell = t$ , we call instance  $\ell$  a high variance instance. Otherwise we call instance  $\ell$  a low variance instance.*

The next section describes the  $\text{SDP}^*$  that we formulate and solve for just the low variance instances. The claim that [Step 0d](#) of the algorithm shown in [Figure 9.1](#) handles the high variance instances is discussed and proved in [Section 9.4.3](#).

### Basic SDP formulation for simultaneous MAX-CUT

We write the  $\text{SDP}^*$  for simultaneous MAX-CUT problem, after the partial fixing given by pre-processing step, as in [Figure 9.2](#). Let  $\mathcal{L}$  denote the set of indices of the low variance instances. We have vectors  $\mathbf{v}_{T,\alpha}$  for all  $T$  and  $\alpha$  where  $T$  is a subset of  $V$  of size at most 2, and  $\alpha$  is an assignment to the vertices in  $T$ .

If we consider the  $\text{SDP}^*$  without the constraint [\(9.4.2\)](#), it is easy to see that this is a relaxation. Given a partition  $(U, \bar{U})$  of  $V$  that achieves a simultaneous optimum, we

$$\sum_{e=\{i,j\} \in \mathcal{E}_\ell} \mathcal{E}_\ell(e) (\|\mathbf{v}_{\{i,j\},(0,1)}\|_2^2 + \|\mathbf{v}_{\{i,j\},(1,0)}\|_2^2) \geq (1 - 3\varepsilon)c_\ell \quad \forall \ell \in [k], \quad (9.4.1)$$

$$\sum_{e=\{i,j\} \in \text{Active}(S^*)} \mathcal{E}_\ell(e) (\|\mathbf{v}_{\{i,j\},(0,1)}\|_2^2 + \|\mathbf{v}_{\{i,j\},(1,0)}\|_2^2) \geq \varepsilon/3 \cdot \text{activedegree}_{S^*}(\ell) \quad \forall \ell \in \mathcal{L} \quad (9.4.2)$$

$$\begin{aligned} \langle \mathbf{v}_{\{i,0\}}, \mathbf{v}_{\{i,1\}} \rangle &= 0 & \forall i \in [n], \\ \|\mathbf{v}_{\{i,j\},(b_1,b_2)}\|^2 &= \langle \mathbf{v}_{\{i,b_1\}}, \mathbf{v}_{\{j,b_2\}} \rangle & \forall i, j \in [n] \\ & & \text{and } b_1, b_2 \in \{0, 1\} \\ \|\mathbf{v}_{\{T,\alpha\}}\|^2 &= \langle \mathbf{v}_{\{T,\alpha\}}, \mathbf{v}_\emptyset \rangle & \forall T \subset V, |T| \leq 2, \alpha \in \{0, 1\}^{|T|} \\ \mathbf{v}_{\{i,b\}} &= \mathbf{v}_\emptyset & \forall i \in S^*, b = h(i) \end{aligned}$$

Figure 9.2:  $\text{SDP}^*(h : S^* \rightarrow \{0, 1\})$  for simultaneous MAX-CUT with partial fixing

can set vectors  $\mathbf{v}_{T,\alpha} = \mathbf{v}_\emptyset$  if the pair  $(T, \alpha)$  is consistent with  $1_U$  (i.e.  $1_U$  assigns  $\alpha$  to  $T$ ) and  $\mathbf{v}_{T,\alpha} = 0$  otherwise.  $\mathbf{v}_\emptyset$  can be viewed as a vector that denotes 1.

A part of our analysis require that for every low variance instance, the expected weighted fraction of active edges that we cut is at least a constant fraction of its active degree. An optimal SDP solution without constraint (9.4.2) may not guarantee this condition (for the rounding procedure we choose). Hence, we force the SDP solution to satisfy this property by adding constraint (9.4.2). We need to relax constraint (9.4.1) to make sure that there is a solution that satisfies all the constraints.

We now prove that  $\text{SDP}^*$ , in its present form, has feasible solutions.

**Lemma 9.4.4.**  $\text{SDP}^*(h^*)$  shown in Figure 9.2 has a feasible solution.

*Proof.* To show that  $\text{SDP}^*$  has a feasible solution, it suffices to show that there exists an integral solution that satisfies the constraints.

Fix an optimal assignment  $f^* : V \rightarrow \{0, 1\}$  to the simultaneous instance.  $f^*$  satisfies  $\forall \ell \in [k], \text{val}(f^*, \mathcal{E}_\ell) \geq c_\ell$ . Consider the following random assignment: For all  $v \in V \setminus S^*$

$$r(v) = \begin{cases} f^*(v) & \text{with probability } (1 - \varepsilon) \\ \overline{f^*(v)} & \text{otherwise} \end{cases}$$

and for  $v \in S^*$ , set  $r(v) = f^*(v)$ . Now, for any  $\ell \in \mathcal{L}$ , let  $Y_\ell$  denote the random variable

$$Y_\ell = \sum_{e \in \text{Active}(S^*)} \mathcal{E}_\ell(e) \cdot e(r)$$

We have  $\mathbf{E}[e(r)] \geq \varepsilon$ , hence  $\mathbf{E}[Y_\ell] \geq \varepsilon/2 \cdot \text{activedegree}_{S^*}(\ell)$ . Also,

$$\begin{aligned} \mathbf{E}_r[\text{val}(r, \mathcal{E}_\ell)] &\geq \sum_{e \notin \text{Active}(S^*)} \mathcal{E}_\ell(e) \cdot \mathbf{E}[e(r)] + \sum_{\substack{e \in \text{Active}(S^*), \\ e(f^*)=1}} \mathcal{E}_\ell(e) \cdot \mathbf{E}[e(r)] \\ &= \sum_{e \notin \text{Active}(S^*)} \mathcal{E}_\ell(e) \cdot e(f^*) + \sum_{\substack{e \in \text{Active}(S^*), \\ e(f^*)=1}} \mathcal{E}_\ell(e) \cdot ((1 - \varepsilon)^2 + \varepsilon^2) \\ &\geq (1 - 2\varepsilon) \sum_{e: e(f^*)=1} \mathcal{E}_\ell(e) = (1 - 2\varepsilon) \text{val}(f^*, \mathcal{E}_\ell) \geq (1 - 2\varepsilon) c_\ell \end{aligned}$$

Thus, we have,

1.  $\mathbf{E}[Y_\ell] \geq \varepsilon/2 \cdot \text{activedegree}_{S^*}(\ell)$ .
2.  $\mathbf{E}_r[\text{val}(r, \mathcal{E}_\ell)] \geq (1 - 2\varepsilon) c_\ell$

Recall that the  $\text{SDP}^*$  involves only the low variance instances. Also, the assignment  $r$  is  $\varepsilon$ -smooth on the set  $V \setminus S^*$ . Therefore, we have concentration guarantees as given by point 1 of [Lemma 9.4.2](#).

$$\Pr[Y_\ell \leq (1 - \varepsilon_0) \mathbf{E}[Y_\ell]] \leq \delta_0$$

$$\Pr[\text{val}(r, \mathcal{E}_\ell) \leq (1 - \varepsilon_0) \mathbf{E}[\text{val}(r, \mathcal{E}_\ell)]] \leq \delta_0$$

Hence, with probability at least  $1 - 2\delta_0$ , we have  $Y_\ell \geq (1 - \varepsilon/2) \cdot \varepsilon/2 \cdot \text{activedegree}_{S^*}(\ell) \geq \varepsilon/3 \cdot \text{activedegree}_{S^*}(\ell)$  and  $\text{val}(r, \mathcal{E}_\ell) \geq (1 - \varepsilon/2)(1 - 2\varepsilon) c_\ell \geq (1 - 3\varepsilon) c_\ell$ .

Now we do union bound over all low variance instances, we get with a probability at least  $1 - 2 \cdot \delta_0 \cdot k = 4/5$ , all the SDP constraints are satisfied by integral solution  $r$ . Thus, there exists an *integral* solution which satisfies all  $\text{SDP}^*(h^*)$  constraints and hence is feasible.  $\blacksquare$

### Lasserre Hierarchy SDP formulation

The  $r^{\text{th}}$  level Lasserre SDP for the SDP in Figure 9.2 can be written as follows: The SDP formulation has vectors  $\mathbf{v}_{\{T, \alpha\}}$  for all  $T \subseteq V$  such that  $|T| \leq r$  and  $\alpha \in \{0, 1\}^{|T|}$ . In terms of local distribution, the SDP solution consists of *consistent local distribution* on every set  $T$  of size at most  $r$  (denoted by  $\mu_T$ ). The random variable corresponding to set  $T$  is denoted by  $X_T$  distributed over  $\{0, 1\}^{|T|}$ . The vector solution and the local distribution are related as follows: Suppose  $T$  and  $U$  are subsets of  $V$  such that  $|T \cup U| \leq r$  and the assignments  $\alpha \in \{0, 1\}^{|T|}$  and  $\beta \in \{0, 1\}^{|U|}$  are consistent on  $T \cap U$  then

$$\langle \mathbf{v}_{T, \alpha}, \mathbf{v}_{U, \beta} \rangle = \Pr_{\mu_{T \cup U}} (X_T = \alpha, X_U = \beta)$$

To ensure the consistency among local distributions, we have to add the constraints 9.4.5 and 9.4.6 to the SDP in Figure 9.3. Here if  $\alpha \in \{0, 1\}^{|S|}$  is an assignment to the vertices in  $S$ , and if  $S' \subset S$ ,  $\alpha_{|S'} \in \{0, 1\}^{|S'|}$  denotes the assignment  $\alpha$  restricted to the vertices in  $S'$ . Also, if  $\alpha$  and  $\beta$  are assignments to sets  $S$  and  $T$  agreeing on  $S \cap T$ , then we denote  $\alpha \circ \beta$  an assignment to  $S \cup T$ . We also add the set of constraints (Equation 9.4.7 in Figure 9.3) to capture the partial assignment  $h : S^* \rightarrow \{0, 1\}$  given by pre-processing.

With these definitions and constraints, the objective is to ensure that for all  $\ell \in [k]$ ,

$$\sum_{e=\{i,j\} \in \mathcal{E}_\ell} \mathcal{E}_\ell(e) (\Pr (X_{\{i,j\}} = (0, 1) \vee X_{\{i,j\}} = (1, 0))) \geq (1 - 3\varepsilon)c_\ell$$

A simple way to capture this would be to write the objective of the SDP solution similar to the basic SDP formulation, as follows.

$$\sum_{e=\{i,j\} \in \mathcal{E}_\ell} \mathcal{E}_\ell(e) (\|\mathbf{v}_{\{(i,j), (0,1)\}}\|_2^2 + \|\mathbf{v}_{\{(i,j), (1,0)\}}\|_2^2) \geq (1 - 3\varepsilon)c_\ell$$

$$\begin{aligned}
& \sum_{e=\{i,j\} \in \mathcal{E}_\ell} ( \mathcal{E}_\ell(e) ( \| \mathbf{v}_{\{\mathbf{S} \cup \{\mathbf{i}, \mathbf{j}\}, \alpha \circ (\mathbf{0}, \mathbf{1})\}} \|_2^2 & \forall S \subseteq V, |S| \leq r-2, \alpha \in \{0,1\}^{|S|}, \\
& \quad + \| \mathbf{v}_{\{\mathbf{S} \cup \{\mathbf{i}, \mathbf{j}\}, \alpha \circ (\mathbf{1}, \mathbf{0})\}} \|_2^2 ) & \forall \ell \in [k] \\
& \geq (1 - 3\varepsilon) c_\ell \| \mathbf{v}_{\{\mathbf{S}, \alpha\}} \|^2 & (9.4.3)
\end{aligned}$$

$$\begin{aligned}
& \sum_{e=\{i,j\} \in \text{Active}(S^*)} ( \mathcal{E}_\ell(e) ( \| \mathbf{v}_{\{\mathbf{S} \cup \{\mathbf{i}, \mathbf{j}\}, \alpha \circ (\mathbf{0}, \mathbf{1})\}} \|_2^2 & \forall S \subseteq V, |S| \leq r-2, \alpha \in \{0,1\}^{|S|}, \\
& \quad + \| \mathbf{v}_{\{\mathbf{S} \cup \{\mathbf{i}, \mathbf{j}\}, \alpha \circ (\mathbf{1}, \mathbf{0})\}} \|_2^2 ) & \forall \ell \in \mathcal{L} \\
& \geq \varepsilon/3 \cdot \text{activedegree}_{S^*}(\ell) & (9.4.4)
\end{aligned}$$

$$\begin{aligned}
\langle \mathbf{v}_{\{\mathbf{S}, \alpha\}}, \mathbf{v}_{\{\mathbf{T}, \beta\}} \rangle &= \| \mathbf{v}_{\{\mathbf{S} \cup \mathbf{T}, \alpha \circ \beta\}} \|_2^2 & \forall S, T \subseteq V, |S \cup T| \leq r, \\
& & \alpha \in \{0,1\}^{|S|}, \beta \in \{0,1\}^{|T|}, \\
& & (9.4.5)
\end{aligned}$$

$$\begin{aligned}
\langle \mathbf{v}_{\mathbf{S}, \alpha}, \mathbf{v}_{\mathbf{T}, \beta} \rangle &= 0 & \forall S, T \subseteq V, |S \cup T| \leq r, \\
& & \alpha \in \{0,1\}^{|S|}, \beta \in \{0,1\}^{|T|}, \\
& \text{s.t. } \alpha_{|S \cap T} \neq \beta_{|S \cap T} & (9.4.6)
\end{aligned}$$

$$\begin{aligned}
\| \mathbf{v}_{\{\mathbf{T}, \alpha\}} \|^2 &= \langle \mathbf{v}_{\{\mathbf{T}, \alpha\}}, \mathbf{v}_\emptyset \rangle & \forall T \subseteq V, |T| \leq r, \alpha \in \{0,1\}^{|T|} \\
\langle \mathbf{v}_{\{\mathbf{S}, \alpha\}}, \mathbf{v}_{\{\mathbf{i}, \mathbf{b}\}} \rangle &= \langle \mathbf{v}_{\{\mathbf{S}, \alpha\}}, \mathbf{v}_\emptyset \rangle & \forall S \subseteq V, |S| \leq r-1, \alpha \in \{0,1\}^{|S|} \\
& & \forall i \in S^*, b = h(i) & (9.4.7)
\end{aligned}$$

Figure 9.3: Lasserre version of  $\text{SDP}^*(h : S^* \rightarrow \{0,1\})$  for simultaneous MAX-CUT with partial fixing

However, in order to make the solution locally independent, we will need to condition based on the local distribution (Refer [Section 9.4.3](#)). Therefore, we need to re-write the objective so that it is satisfied (w.r.t the conditioned local distribution) even after conditioning on at most  $r$  variables, as shown in [Equation 9.4.3](#) in the SDP formulation.

Also, similar to the previous case, we need to ensure that the solution post-conditioning still cuts at least a constant fraction of the active edges, which is ensured by adding the set of constraints specified in [Equation 9.4.4](#) in the SDP.

We observe that solving the SDP using ellipsoid method can result in a small additive error, and if  $\text{activedegree}_{S^*}(\ell)$  is small compared to this additive error, the error would be significant. This will not cause any issues and we elaborate on this more. We can solve the SDP using ellipsoid method with an error of  $\varepsilon$  in time polynomial in  $n$  and  $\log(1/\varepsilon)$ . Therefore, we can take  $\varepsilon$  to be  $\exp(-\text{poly}(n))$  and still solve the SDP time polynomial in  $n$ . We assumed that the edge weights are at least  $\exp(-n^c)$  for some constant  $c > 0$ . Therefore, if the active degree is non-zero, it is at least  $\exp(-n^c)$ . If we take  $\varepsilon = \exp(-n^{c'})$  for  $c' \gg c$ , we can solve the SDP in time polynomial in  $n$  and get a vector solution which satisfies all the constraints upto additive error  $\varepsilon$  which is upto multiplicative factor of  $(1 + o(1))$ . This will not have a major effect on our analysis and hence we assume from here that the vector solution that we get satisfies the all the constraints exactly.

### Obtaining independent local solution

The notion of independent solution (which is formalized below in [Definition 9.4.5](#)) that we need is different from [\[RT12\]](#). Following procedure in [Figure 9.4](#) is used to achieve the kind of independence we need.

**Definition 9.4.5.** *A Lasserre solution is  $\delta$ -independent if it satisfies the following condition.*

$$\forall \ell \in \mathcal{L}, \quad \mathbf{E}_{a,b \sim \text{actdist}_{S^*}(\ell)} \left[ \sum_{i,j \in \{1,2\}} I(X_{a_i}; X_{b_j}) \right] \leq \delta.$$

**Lemma 9.4.6.** *For all  $\delta > 0$ , there exists  $t \leq 2k/\delta$ , there exists  $e^1, e^2, \dots, e^t \in \mathcal{E}$  such that*

$$\forall \ell \in \mathcal{L}, \quad \mathbf{E}_{a,b \sim \text{actdist}_{S^*}(\ell)} [I(X_{a_1}, X_{a_2}; X_{b_1}, X_{b_2} | X_{e_1^1}, X_{e_2^1}, \dots, X_{e_1^t}, X_{e_2^t})] \leq \delta \quad (9.4.8)$$

*Proof.* Consider the following potential function,

$$\phi = \sum_{\ell \in \mathcal{L}} \mathbf{E}_{a \in \text{actdist}_{S^*}(\ell)} H(X_{a_1}, X_{a_2}).$$

**Input:**  $r + 2$  round Lasserre solution of a given simultaneous MAX-CUT instance,

$$\delta := \frac{8 \cdot k^3}{r-2}$$

**Output:**  $\delta$ -independent 2-round Lasserre solution.

1. Sample  $\ell_1, \dots, \ell_r \in \mathcal{L}$  each u.a.r. Sample edges  $e^i \in \text{actdist}_{S^*}(\ell_i)$  u.a.r. for all  $i \in [r]$ .
2. Do the following until the solution becomes  $\delta$ -independent. Set  $t = 1$ .
  - Sample  $(X_{e_1^t}, X_{e_2^t})$  from the marginal distribution after previous  $t - 1$  fixings.
  - Condition the SDP solution on  $(X_{e_1^t}, X_{e_2^t})$ .
  - $t = t + 1$

Figure 9.4: Making locally independent solution

As entropy of a bit is at most 1, clearly  $\phi \leq 2k$ . We have the following identity for each  $\ell \in \mathcal{L}$  which follows from conditional entropy and linearity of expectation

$$\begin{aligned} \mathbf{E}_{a,b \in \text{actdist}_{S^*}(\ell)} [H(X_{a_1}, X_{a_2} | X_{b_1}, X_{b_2})] &= \mathbf{E}_{a \in \text{actdist}_{S^*}(\ell)} [H(X_{a_1}, X_{a_2})] - \\ &\quad \mathbf{E}_{a,b \in \text{actdist}_{S^*}(\ell)} I(X_{a_1}, X_{a_2}; X_{b_1}, X_{b_2}). \end{aligned}$$

This identity suggests that if for some  $\ell \in \mathcal{L}$ ,  $\mathbf{E}_{a,b \in \text{actdist}_{S^*}(\ell)} I(X_{a_1}, X_{a_2}; X_{b_1}, X_{b_2}) > \delta$  then there exists a conditioning which reduces the potential function by at least  $\delta$ . Thus, either the current conditioned solution satisfies (9.4.8) in which case we are done or there exists an edge  $b$  such that if we condition the SDP solution based on the value of its endpoints  $(b_1, b_2)$  according to the local distribution then the potential function decreases by at least  $\delta$ . So, if we fail to achieve (9.4.8) then  $\phi$  decreases by at least  $\delta$ . As entropy is always non-negative and conditioning never increases entropy (Fact 9.3.6), this process cannot go beyond  $2k/\delta$  conditioning. Thus, before at most

$2k/\delta$  conditioning, we are guaranteed to achieve (9.4.8). ■

The following fact follows from the data processing inequality (Theorem 9.3.8)

**Fact 9.4.7.** *If  $X_1, X_2, Y_1$  and  $Y_2$  are random variables then for  $i, j \in \{1, 2\}$ , we have*

$$I(X_i; Y_j) \leq I(X_1, X_2; Y_1, Y_2)$$

The following corollary follows from Lemma 9.4.6 and Fact 9.4.7.

**Corollary 9.4.8.** *For all  $\delta > 0$ , there exists  $t \leq \frac{2k}{\delta}$ , and edges  $e^1, e^2, \dots, e^t \in \mathcal{E}$ ,*

$$\forall \ell \in \mathcal{L}, \quad \mathbf{E}_{a,b \in \text{actdist}_{S^*}(\ell)} \left[ \sum_{i,j \in \{1,2\}} I(X_{a_i}; X_{b_j} | X_{e_1^1}, X_{e_2^1}, \dots, X_{e_1^{t-1}}, X_{e_2^{t-1}}) \right] \leq 4\delta$$

**Lemma 9.4.9.** *There exists a fixing of at most  $\frac{2k}{\delta/4}$  variables such that the conditioned solution is  $\delta$  independent as well as satisfies all constraints from  $\text{SDP}^*(h^*)$ .*

*Proof.*  $\delta$  independence follows from Corollary 9.4.8 and Fact 9.4.7. We now prove the later part.

As the conditioning maintains the marginal distribution of variables and because of the the Inequality (9.4.3) and (9.4.4), the constraints about the SDP cut value as well as the fraction of active edges that are cut remain valid in the conditioned solution. Hence, from Lemma 9.4.4  $\text{SDP}^*(h^*)$  remains feasible. ■

## Rounding Procedure

In this section, we describe the rounding procedure for variables in  $V \setminus S^*$ . The input to this procedure is 2 round Lasserre solution which is  $\delta$ -independent. We use a slight variation of GW rounding procedure to round the SDP vector solution. In particular, we want to maintain the bias of heavily biased random variable in our rounding procedure.

SDP gives the vector solution  $\mathbf{v}_{i,0}, \mathbf{v}_{i,1}$  for all  $i \in [n]$ . Let  $\mu_i = \mathbf{E}[X_i]$ , the expectation is according to the local distribution. Define  $\mathbf{v}_i = \mathbf{v}_{i,1} - \mathbf{v}_{i,0}$ . These  $\mathbf{v}_i$  are the unit

vectors (as  $\|\mathbf{v}_i\|^2 = \|\mathbf{v}_{i,1} - \mathbf{v}_{i,0}\|^2 = \|\mathbf{v}_{i,1}\|^2 + \|\mathbf{v}_{i,0}\|^2 - 2\langle \mathbf{v}_{i,1}, \mathbf{v}_{i,0} \rangle = \Pr[X_i = 0] + \Pr[X_i = 1] - 0 = 1$ ). Let  $\mathbf{w}_i$  be component of  $\mathbf{v}_i$  orthogonal to  $\mathbf{v}_\emptyset$  ( $\mathbf{v}_i = \mu_i \mathbf{v}_\emptyset + \mathbf{w}_i$ ),  $\|\mathbf{w}_i\|_2 = \sqrt{1 - \mu_i^2}$ . Let  $\bar{\mathbf{w}}_i$  be the normalized unit vector of  $\mathbf{w}_i$ . The rounding procedure is applied on vectors  $\bar{\mathbf{w}}_i$  along with the “bias” of each variable  $\langle \mathbf{v}_i, \mathbf{v}_\emptyset \rangle$ . The rounding procedure is shown in Figure 9.5.

**Input:**  $\delta$ -independent 2 round Lasserre solution, biases  $u_i \in [-1, +1]$  and a function  $f_R : [-1, 1] \rightarrow [-1, 1]$  which is bounded by above and below with some constant degree polynomials

**Output:** A partition of  $V$ .

1. Pick a random Gaussian vector  $\mathbf{g}$  orthogonal to  $\mathbf{v}_\emptyset$  with each co-ordinate distributed as  $\mathcal{N}(0, 1)$ .
2. For each  $i \in [n]$ 
  - Calculate  $\xi_i = \langle \mathbf{g}, \bar{\mathbf{w}}_i \rangle$ .
  - Let  $r_i \leftarrow f_R(\mu_i)$
  - Set  $y_i = 1$  if  $\xi_i \leq \Phi^{-1}(r_i/2 + 1/2)$ , otherwise set  $y_i = -1$ . (Here,  $\Phi$  is the Gaussian CDF)

Figure 9.5: Rounding procedure

### Analysis of the rounding procedure

We use the notation  $\text{poly}(x)$  to denote a fixed constant degree polynomial in  $x$  such that  $\text{poly}(x) \rightarrow 0$  as  $x \rightarrow 0$ .

Note that if we simply use the rounding function  $f_R(x) = x$  as used in [RT12] the we get for each instance, in expectation the cut produced by the rounding procedure is at least 0.85 times the SDP value (and hence eventually 0.85 approximation for simultaneous MAX-CUT). Here, we leverage the fact that the constraints on what rounding

functions are good for us are mild compared to [RT12] as explained in Section 9.2.1.

**Lemma 9.4.10.** *For a fixed instance, in expectation, the rounding procedure described in Figure 9.5 gives an approximation ratio 0.8782 for the following  $f_R$ ,*

$$\begin{aligned} f_R(x) &= (0.79 + 0.1 \cdot x^2)x && \text{if } |x| < 0.47 \\ f_R(x) &= (0.815 + 0.185 \cdot x^6)x && \text{otherwise} \end{aligned}$$

*Proof.* The proof of this lemma is numerical. We arrive at a informal approximate value for the bound using Matlab code (0.8782) and verify it using computer assisted techniques. The program works in a recursive fashion, by continuously splitting the cube into sub-cubes. In each sub-cube, the program checks if either across all points in the region, the lower bound on  $\alpha$  exceeds the approximation ratio we try to prove or if the upper bound on  $\alpha$  is lower than the approximation ration we try to prove. It proceeds with further division into smaller sub-cubes until one of the above is satisfied. If the latter is true at any point, the code returns a failure, and it returns a success if the entire region can be proved to come under the former case. The prover was adapted from [ABG12] and modified to suit our rounding procedure. For more details on the workings of the prover, refer [ABG12]. ■

**Lemma 9.4.11** ([RT12]). *Let  $\mathbf{v}_i$  and  $\mathbf{v}_j$  be the unit vectors,  $\mathbf{w}_i$  and  $\mathbf{w}_j$  be the components of  $\mathbf{v}_i$  and  $\mathbf{v}_j$  that are orthogonal to  $\mathbf{v}_\emptyset$ . Then  $|\langle \mathbf{w}_i, \mathbf{w}_j \rangle| \leq 2I(X_i; X_j)$ .*

Above lemma along with Lemma 9.4.9 implies that if we sample edge  $(i_1, i_2), (j_1, j_2) \sim \text{actdist}_{S^*}(\ell)$  then we have on average,

$$|\langle \mathbf{w}_{i_1}, \mathbf{w}_{j_1} \rangle| + |\langle \mathbf{w}_{i_1}, \mathbf{w}_{j_2} \rangle| + |\langle \mathbf{w}_{i_2}, \mathbf{w}_{j_1} \rangle| + |\langle \mathbf{w}_{i_2}, \mathbf{w}_{j_2} \rangle| \leq \delta$$

The rounding procedure is assigning values  $\pm 1$  to variables  $y_i$  where  $y_i$  is the variable for vertex  $i \in V$  and its value decides on which side of cut the vertex  $i$  is present in the final solution. Thus  $y_i$  is a random variable taking values in  $\{+1, -1\}$ . We now wish to prove similar guarantee as the following lemma from [RT12], which relates the mutual information between the pair of rounded variables with the inner product of the corresponding vectors  $w$ .

**Lemma 9.4.12** ([RT12]). *For  $f_R$  such that  $f_R(x) = x$ , if  $|\langle \mathbf{w}_i, \mathbf{w}_j \rangle| \leq \delta$  then  $I(y_i; y_j) \leq \delta^{1/3}$ .*

In our case, we need that the mutual information between the events that a pair of edges are cut is small on average. Thus, our notion of local independence will be useful in proving this guarantee about mutual information.

**Lemma 9.4.13.** *Fix  $f_R$  to be the rounding function given by Lemma 9.4.10. For a pair of edges  $(i_1, i_2)$  and  $(j_1, j_2)$ , suppose the vectors  $w$  corresponding to their endpoints satisfy the following condition,*

$$|\langle \mathbf{w}_{i_1}, \mathbf{w}_{j_1} \rangle| + |\langle \mathbf{w}_{i_1}, \mathbf{w}_{j_2} \rangle| + |\langle \mathbf{w}_{i_2}, \mathbf{w}_{j_1} \rangle| + |\langle \mathbf{w}_{i_2}, \mathbf{w}_{j_2} \rangle| \leq \delta$$

*then  $I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2}) \leq \text{poly}(\delta)$ .*

*Proof.* Since  $\bar{\mathbf{w}}_i$  is a normalized vector of  $\mathbf{w}_i$  and  $\|\mathbf{w}_i\| = \sqrt{1 - \mu_i^2}$ , we have

$$\left. \begin{aligned} & \sqrt{1 - \mu_{i_1}^2} \cdot \sqrt{1 - \mu_{j_1}^2} \cdot |\langle \bar{\mathbf{w}}_{i_1}, \bar{\mathbf{w}}_{j_1} \rangle| \\ & + \sqrt{1 - \mu_{i_1}^2} \cdot \sqrt{1 - \mu_{j_2}^2} \cdot |\langle \bar{\mathbf{w}}_{i_1}, \bar{\mathbf{w}}_{j_2} \rangle| \\ & + \sqrt{1 - \mu_{i_2}^2} \cdot \sqrt{1 - \mu_{j_1}^2} \cdot |\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_1} \rangle| \\ & + \sqrt{1 - \mu_{i_2}^2} \cdot \sqrt{1 - \mu_{j_2}^2} \cdot |\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_2} \rangle| \end{aligned} \right\} \leq \delta \quad (9.4.9)$$

Since the total sum is bounded and each quantity is non-negative, at least one of the three quantities in each summand is at most  $\delta^{1/3}$ . We use two crucial properties of the rounding procedure:

- For the heavily biased variable according to the local distribution, the rounding procedure also keeps the rounded value heavily biased and
- If two vectors  $\mathbf{w}_i$  and  $\mathbf{w}_j$  are nearly orthogonal, the corresponding rounded values  $y_i$  and  $y_j$  are nearly independent.

We need following claim which we prove in Section 9.5.

**Claim 9.4.14.** *If all these quantities  $|\langle \bar{\mathbf{w}}_{i_1}, \bar{\mathbf{w}}_{j_1} \rangle|, |\langle \bar{\mathbf{w}}_{i_1}, \bar{\mathbf{w}}_{j_2} \rangle|, |\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_1} \rangle|, |\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_2} \rangle|$  are upper bounded by  $\delta^{1/3}$ , then we can upper bound  $I(y_{i_1}, y_{i_2}; (y_{j_1}, y_{j_2})) \leq \text{poly}(\delta)$*

We now formally prove the upper bound on  $I(y_{i_1}y_{i_2}; y_{j_1}y_{j_2})$  by case analysis. We use the following upper bound which follows from data processing inequality.

$$I(y_{i_1}y_{i_2}; y_{j_1}y_{j_2}) \leq I(y_{i_1}, y_{i_2}; (y_{j_1}, y_{j_2}))$$

We now bound the right hand side based on following case analysis.

- Case 1: If all these quantities  $|\langle \bar{\mathbf{w}}_{i_1}, \bar{\mathbf{w}}_{j_1} \rangle|, |\langle \bar{\mathbf{w}}_{i_1}, \bar{\mathbf{w}}_{j_2} \rangle|, |\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_1} \rangle|, |\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_2} \rangle|$  are upper bounded by  $\delta^{1/3}$  then using [Claim 9.4.14](#), we can upper bound  $I(y_{i_1}, y_{i_2}; (y_{j_1}, y_{j_2})) \leq \text{poly}(\delta)$ .
- Case 2: Consider the case when both the endpoints of an edge (w.l.o.g. of  $(i_1, i_2)$ ) have large bias i.e.  $\sqrt{1 - \mu_{i_1}^2} \leq \delta^{1/3}, \sqrt{1 - \mu_{i_2}^2} \leq \delta^{1/3}$ . It implies,

$$\min(|1 - \mu_{i_1}|, |1 + \mu_{i_1}|) \leq \delta^{2/3}, \quad \min(|1 - \mu_{i_2}|, |1 + \mu_{i_2}|) \leq \delta^{2/3}.$$

Assume both  $\mu_{i_1}, \mu_{i_2} > 0$  (there cases can be handled in a similar way). Then we have,  $1 - \mu_{i_1} \leq \delta^{2/3}$  and  $1 - \mu_{i_2} \leq \delta^{2/3}$ . Since the rounding procedure maintains the bias of a variable for a heavily biased variables, up to some constant polynomial factor, we have,

$$\begin{aligned} I(y_{i_1}, y_{i_2}; (y_{j_1}, y_{j_2})) &\leq H(y_{i_1}, y_{i_2}) \leq H(y_{i_1}) + H(y_{i_2}) \\ &= O(-(1 - \text{poly}(\mu_{i_1})) \log(1 - \text{poly}(\mu_{i_1}))) + \\ &\quad O(-(1 - \text{poly}(\mu_{i_2})) \log(1 - \text{poly}(\mu_{i_2}))) \\ &\leq \text{poly}(\delta). \end{aligned}$$

- Case 3: Consider the case when exactly two non-endpoints of an edge (w.l.o.g. of  $(i_1, j_i)$ ) have large bias. This implies that  $\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_2} \rangle \leq \delta^{1/3}$ . Using the analysis of the previous case we have  $H(y_{i_1}), H(y_{j_1}) \leq \text{poly}(\delta)$ . Mutual information can

be bounded as follows:

$$\begin{aligned}
I((y_{i_1}, y_{i_2}); (y_{j_1}, y_{j_2})) &\leq H((y_{i_1}, y_{i_2})) - H((y_{i_1}, y_{i_2})|(y_{j_1}, y_{j_2})) \\
&\leq H(y_{i_1}) + H(y_{i_2}) - H(y_{i_2}|(y_{j_1}, y_{j_2})) \\
&= H(y_{i_1}) + I((y_{j_1}, y_{j_2}); y_{i_2}) \tag{9.4.10}
\end{aligned}$$

$$= \text{poly}(\delta) + I((y_{j_1}, y_{j_2}); y_{i_2}) \tag{9.4.11}$$

Now,

$$\begin{aligned}
I((y_{j_1}, y_{j_2}), y_{i_2}) &= H((y_{j_1}, y_{j_2})) - H((y_{j_1}, y_{j_2})|y_{i_2}) \\
&\leq H(y_{j_1}) + H(y_{j_2}) - H(y_{j_2}|y_{i_2}) \\
&= H(y_{j_1}) + I(y_{j_2}; y_{i_2}) = \text{poly}(\delta) + I(y_{j_2}; y_{i_2})
\end{aligned}$$

Therefore, we have

$$I(y_{i_1}y_{i_2}; y_{j_1}y_{j_2}) \leq \text{poly}(\delta) + I(y_{j_2}; y_{i_2})$$

From [Claim 9.4.14](#),  $I(y_{j_2}; y_{i_2})$  is bounded above by  $\text{poly}(\delta)$  as  $\langle \bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_2} \rangle \leq \delta^{1/3}$

- Case 4: Consider the only remaining case in which exactly one variable, say  $X_{i_1}$ , has a large bias i.e.  $\sqrt{1 - \mu_{i_1}^2} \leq \delta^{1/3}$ . From [\(9.4.9\)](#), it implies that pairwise inner products of  $\bar{\mathbf{w}}_{i_2}, \bar{\mathbf{w}}_{j_1}$  and  $\bar{\mathbf{w}}_{j_2}$  are at most  $\delta^{1/3}$ . Hence by [Claim 9.4.14](#), we have  $I(y_{i_2}; (y_{j_1}, y_{j_2})) \leq \text{poly}(\delta)$ . As before from [\(9.4.10\)](#),

$$I(y_{i_1}, y_{i_2}; (y_{j_1}, y_{j_2})) \leq H(y_{i_1}) + I((y_{j_1}, y_{j_2}); y_{i_2}) \leq \text{poly}(\delta)$$

■

We can now upper bound the variance of a cut produced by the randomized rounding in graph  $\ell \in \mathcal{L}$ . Define  $Y_\ell$  to be a random variable which is equal to the total weight of active edges cut by the rounding procedure.

$$Y_\ell = \sum_{C \in \text{Active}(S^*)} \mathcal{E}_\ell(C) e(g).$$

**Lemma 9.4.15.** *Fix a rounding function  $f_R$  given in [Lemma 9.4.10](#) and let the SDP solution is  $\delta$  independent then*

$$\text{Var}(Y_\ell) \leq \frac{\text{poly}(\delta)}{\varepsilon^2} \mathbf{E}[Y_\ell]^2.$$

*Proof.* Let  $\alpha := 0.8782$ . Note that by [Lemma 9.4.10](#), we have for an active edge  $e(i, j)$ ,

$$\Pr[e(i, j) \text{ is cut}] \geq \alpha \cdot \frac{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle}{2}. \quad (9.4.12)$$

We now lower bound the expected value of  $Y_\ell$ .

$$\begin{aligned} \mathbf{E}[Y_\ell] &= \sum_{e \in \text{Active}(S^*)} \mathcal{E}_\ell(e) \cdot \Pr[e(i, j) \text{ is cut}] \\ (\text{ from (9.4.12) }) &\geq \alpha \sum_{e \in \text{Active}(S^*)} \mathcal{E}_\ell(e) \cdot \frac{1 - \langle \mathbf{v}_i, \mathbf{v}_j \rangle}{2} \\ &= \alpha \cdot \sum_{e \in \text{Active}(S^*)} \mathcal{E}_\ell(e) (\|\mathbf{v}_{\{(i,j), (0,1)\}}\|_2^2 + \|\mathbf{v}_{\{(i,j), (1,0)\}}\|_2^2) \\ (\text{ from (9.4.2) }) &\geq \alpha \cdot \varepsilon/3 \cdot \text{activedegree}_{S^*}(\ell), \end{aligned}$$

We can now bound the variance as follows:

$$\begin{aligned} \text{Var}(Y_\ell) &= \sum_{i,j \sim \text{Active}(S^*)} \mathcal{E}_\ell(i) \mathcal{E}_\ell(j) [\text{Cov}(y_{i_1} y_{i_2}, y_{j_1} y_{j_2})] \\ &\leq \sum_{i,j \sim \text{Active}(S^*)} \mathcal{E}_\ell(i) \mathcal{E}_\ell(j) [O(\sqrt{I(y_{i_1} y_{i_2}; y_{j_1} y_{j_2})})] \\ (\text{from Lemma 9.4.13}) &\leq \sum_{i,j \sim \text{Active}(S^*)} \mathcal{E}_\ell(i) \mathcal{E}_\ell(j) \cdot \text{poly} \left( |\langle \mathbf{w}_{i_1}, \mathbf{w}_{j_1} \rangle| + |\langle \mathbf{w}_{i_1}, \mathbf{w}_{j_2} \rangle| + \right. \\ &\quad \left. |\langle \mathbf{w}_{i_2}, \mathbf{w}_{j_1} \rangle| + |\langle \mathbf{w}_{i_2}, \mathbf{w}_{j_2} \rangle| \right) \\ (\text{ from Lemma 9.4.11}) &\leq \sum_{i,j \sim \text{Active}(S^*)} \mathcal{E}_\ell(i) \mathcal{E}_\ell(j) \text{poly} \left( \mathbf{E}_{\substack{a \sim \{i_1, i_2\}, \\ b \sim \{j_1, j_2\}}} [I(X_a; X_b)] \right) \\ &\leq \text{activedegree}_{S^*}(\ell)^2 \mathbf{E}_{i,j \sim \text{actdist}_{S^*}(\ell)} \text{poly} \mathbf{E}_{\substack{a \sim \{i_1, i_2\}, \\ b \sim \{j_1, j_2\}}} [I(X_a; X_b)] \\ &\leq \text{activedegree}_{S^*}(\ell)^2 \text{poly} \left( \mathbf{E}_{\substack{(i_1, i_2), \\ (j_1, j_2) \sim \text{actdist}_{S^*}(\ell)}} \mathbf{E}_{\substack{a \sim \{i_1, i_2\}, \\ b \sim \{j_1, j_2\}}} [I(X_a; X_b)] \right) \\ &\leq \text{poly}(\delta) \cdot \text{activedegree}_{S^*}(\ell)^2 \end{aligned}$$

Thus, we have

$$\text{Var}(Y_\ell) \leq \frac{\text{poly}(\delta)}{\varepsilon^2} \mathbf{E}[Y_\ell]^2.$$

■

**Corollary 9.4.16.** *If we set  $r := \text{poly}(k, 1/\varepsilon)$  then for every low variance instance  $\ell \in [k]$ , with probability at least  $1 - 1/10k$  we have  $\text{val}(h^* \cup g) \geq (0.8782 - 4\varepsilon)c_\ell$ .*

*Proof.* Choosing  $r$  a large constant, by Lemma 9.4.15 and application of Chebyshev's Inequality, we can deduce that with probability at least  $1 - 1/10k$ , we have  $Y_\ell \geq (1 - \varepsilon) \mathbf{E}[Y_\ell]$ . Thus, with probability at least  $1 - 1/10k$ , we have,

$$\begin{aligned} \text{val}(h^* \cup g, \mathcal{E}_\ell) &= \text{val}(h^*, \mathcal{E}_\ell) + Y_\ell \geq \text{val}(h^*, \mathcal{E}_\ell) + (1 - \varepsilon) \mathbf{E}[Y_\ell] \\ &\geq (1 - \varepsilon) \cdot \mathbf{E}[\text{val}(h^*, \mathcal{E}_\ell) + Y_\ell] = (1 - \varepsilon) \cdot \mathbf{E}[\text{val}(h^* \cup g, W_\ell)] \\ &\geq (1 - \varepsilon) \cdot 0.8782 \cdot (1 - 3\varepsilon) \cdot c_\ell \geq (0.8782 - 4\varepsilon) \cdot c_\ell, \end{aligned}$$

where we have used Lemma 9.4.10 for the lower bound  $\mathbf{E}[\text{val}(h^* \cup g, W_\ell)] \geq 0.8782 \cdot (1 - 3\varepsilon)c_\ell$ , ■

## Post-Processing

**Lemma 9.4.17.** *For all high variance instances  $\ell \in [k]$ , we have*

1.  $\text{activedegree}_{S^*}(\ell) \leq 2(1 - \gamma)^t$ .
2. *For each of the first  $t/2$  variables that were brought inside  $S^*$  because of instance  $\ell$ , the total weight of constraints incident on each of that variable and totally contained inside  $S^*$  is at least  $10 \cdot \text{activedegree}_{S^*}(\ell)$ .*

*Proof.* Consider any high variance instance  $\ell \in [k]$ . Initially, when  $S = \emptyset$ , we have  $\text{activedegree}_\emptyset(\mathcal{E}_\ell) \leq 2$  since the weight of every edge is counted at most twice, once for each of the 2 active vertices of the edge, and  $\sum_{e \in \mathcal{E}} \mathcal{E}_\ell(e) = 1$ . For every  $v$ , note that  $\text{activedegree}_{S_2}(v, \mathcal{E}_\ell) \leq \text{activedegree}_{S_1}(v, \mathcal{E}_\ell)$  whenever  $S_1 \subseteq S_2$ .

Let  $u$  be one of the vertices that ends up in  $S^*$  because of instance  $\ell$ . Let  $S_u$  denote the set  $S \subseteq S^*$  just before  $u$  was brought into  $S^*$ . When  $u$  is added to  $S_u$ , we know that  $\text{activedegree}_{S_u}(u, \mathcal{E}_\ell) \geq \gamma \cdot \text{activedegree}_{S_u}(\ell)$ . Hence,  $\text{activedegree}_{S_u \cup \{u\}}(\ell) \leq \text{activedegree}_{S_u}(\ell) - \text{activedegree}_{S_u}(u, \mathcal{E}_\ell) \leq (1 - \gamma) \cdot \text{activedegree}_{S_u}(\ell)$ . Since  $t$  vertices were brought into  $S^*$  because of instance  $\ell$ , and initially  $\text{activedegree}_\emptyset(\ell) \leq 2$ , we get  $\text{activedegree}_{S^*}(\ell) \leq 2(1 - \gamma)^t$ .

Now, let  $u$  be one of the first  $t/2$  vertices that ends up in  $S^*$  because of instance  $\ell$ . Since at least  $t/2$  vertices are brought into  $S^*$  because of instance  $\ell$ , after  $u$ , as above, we get  $\text{activedegree}_{S^*}(\ell) \leq (1-\gamma)^{t/2} \cdot \text{activedegree}_{S_u}(\ell)$ . Combining with  $\text{activedegree}_{S_u}(u, \mathcal{E}_\ell) \geq \gamma \cdot \text{activedegree}_{S_u}(\ell)$ , we get  $\text{activedegree}_{S_u}(u, \mathcal{E}_\ell) \geq \gamma(1-\gamma)^{-t/2} \text{activedegree}_{S^*}(\ell)$ , which is at least  $11 \cdot \text{activedegree}_{S^*}(\ell)$ , by the choice of parameters. Since any edge incident on a vertex in  $V \setminus S^*$  contributes its weight to  $\text{activedegree}_{S^*}(\ell)$ , the total weight of edges incident on  $u$  and totally contained inside  $S^*$  is at least  $10 \cdot \text{activedegree}_{S^*}(\ell)$  as required.  $\blacksquare$

We now describe a procedure **PERTURB** (see [Figure 9.6](#)) which takes  $h^* : S^* \rightarrow \{0, 1\}$  and  $g : V \setminus S^* \rightarrow \{0, 1\}$ , and produces a new  $h : S^* \rightarrow \{0, 1\}$  such that for all (low variance as well as high variance) instances  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell)$  is not much smaller than  $\text{val}(h^* \cup g, \mathcal{E}_\ell)$ , and furthermore, for all high variance instances  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell)$  is large. The procedure works by picking a special vertex in  $S^*$  for every high variance instance and perturbing the assignment of  $h^*$  to these special vertices. The partial assignment  $h$  is what we will be using to argue that [Step 0d](#) of the algorithm produces a good Pareto approximation. More formally, we have the following Lemma.

**Lemma 9.4.18.** *For the assignment  $h$  obtained from Procedure **PERTURB** (see [Figure 9.6](#)), for each  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^* \cup g, \mathcal{E}_\ell)$ . Furthermore, for each high variance instance  $\mathcal{E}_\ell$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq 4 \cdot \text{activedegree}_{S^*}(\ell)$ .*

*Proof.* Consider the special vertex  $v_\ell$  that we choose for *high variance* instance  $\ell \in [k]$ . Since  $v_\ell \notin B$ , the edges incident on  $v_\ell$  only contribute at most a  $\varepsilon/2k$  fraction of the objective value in each instance. Thus, changing the assignment  $v_\ell$  can reduce the value of any instance by at most a  $\frac{\varepsilon}{2k}$  fraction of their current objective value. Also, we pick different special variables for each *high variance* instance. Hence, the total effect of these perturbations on any instance is that it reduces the objective value (given by  $h^* \cup g$ ) by at most  $1 - (1 - \frac{\varepsilon}{2k})^k \leq \frac{\varepsilon}{2}$  fraction. Hence for all instances  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^* \cup g, \mathcal{E}_\ell)$ .

For a *high variance* instance  $\ell \in [k]$ , since  $v_\ell \in U$ , the vertex  $v_\ell$  must be one of the first  $t/2$  variables brought into  $S^*$  because of  $\ell$ . Hence, by [Lemma 9.4.17](#) the total

**Input:**  $h^* : S^* \rightarrow \{0, 1\}$  and  $g : V \setminus S^* \rightarrow \{0, 1\}$

**Output:** A perturbed assignment  $h : S^* \rightarrow \{0, 1\}$ .

1. Initialize  $h \leftarrow h^*$ .
2. For  $\ell = 1, \dots, k$ , if instance  $\ell$  is a high variance instance case (i.e.,  $\text{count}_\ell = t$ ), we pick a special variable  $v_\ell \in S^*$  associated to this instance as follows:
  - (a) Let  $B = \{v \in V \mid \exists \ell \in [k] \text{ with } \sum_{e \in \mathcal{E}, e \ni v} \mathcal{E}_\ell(e) \cdot e(h \cup g) \geq \frac{\varepsilon}{2k} \cdot \text{val}(h \cup g, \mathcal{E}_\ell)\}$ .  
 Since the weight of each edge is counted at most twice, we know that  $|B| \leq \frac{4k^2}{\varepsilon}$ .
  - (b) Let  $U$  be the set consisting of the first  $t/2$  vertices brought into  $S^*$  because of instance  $\ell$ .
  - (c) Since  $t/2 > |B| + k$ , there exists some  $u \in U$  such that  $u \notin B \cup \{v_1, \dots, v_{\ell-1}\}$ . We define  $v_\ell$  to be  $u$ .
  - (d) By [Lemma 9.4.17](#), the total  $\mathcal{E}_\ell$  weight of edges that are incident on  $v_\ell$  and only containing vertices from  $S^*$  is at least  $10 \cdot \text{activedegree}_{S^*}(\ell)$ . We update  $h$  by setting  $h(v_\ell)$  to be that value from  $\{0, 1\}$  such that at least half of the  $\mathcal{E}_\ell$  weight of these edges is satisfied.
3. Return the assignment  $h$ .

Figure 9.6: Procedure PERTURB for perturbing the optimal assignment

weight of edges that are incident on  $v_\ell$  and entirely contained inside  $S^\star$  is at least  $10 \cdot \text{activedegree}_{S^\star}(\ell)$ . Hence, there is an assignment to  $v_\ell$  that satisfies at least half the weight of these MAX-CUT constraints in  $\ell$ . At the end of the iteration when we pick an assignment to  $v_\ell$ , we have  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq 5 \cdot \text{activedegree}_{S^\star}(\ell)$ . Since the later perturbations do not affect value of this instance by more than  $\varepsilon/2$  fraction, we get that for the final assignment  $h$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq (1 - \varepsilon/2) \cdot 5 \cdot \text{activedegree}_{S^\star}(\ell) \geq 4 \cdot \text{activedegree}_{S^\star}(\ell)$ .  $\blacksquare$

**Theorem 9.4.19.** *Suppose we're given  $\varepsilon \in (0, 2/5]$ ,  $k$  simultaneous MAX-CUT instances  $\mathcal{E}_1, \dots, \mathcal{E}_k$  on  $n$  variables, and target objective value  $c_1, \dots, c_k$  with the guarantee that there exists an assignment  $f^\star$  such that for each  $\ell \in [k]$ , we have  $\text{val}(f^\star, \mathcal{E}_\ell) \geq c_\ell$ . Then, the algorithm ALG-SIM-MAXCUT runs in time  $\exp(k^3/\varepsilon^2 \log(k/\varepsilon^2)) \cdot n^{\text{poly}(k)}$ , and with probability at least 0.9, outputs an assignment  $f$  such that for each  $\ell \in [k]$ , we have,  $\text{val}(f, \mathcal{E}_\ell) \geq (0.8782 - 5\varepsilon) \cdot c_\ell$ .*

*Proof.* Let  $\alpha := 0.8782$ . By Corollary 9.4.16 and a union bound, with probability at least 0.9, over the choice of  $g$ , we have that for every low variance instance  $\ell \in [k]$ ,  $\text{val}(h^\star \cup g, \mathcal{E}_\ell) \geq (\alpha - 4\varepsilon) \cdot c_\ell$ . Henceforth we assume that the assignment  $g$  sampled in Step 0c of the algorithm is such that this event occurs. Let  $h$  be the output of the procedure PERTURB given in Figure 9.6 for the input  $h^\star$  and  $g$ . By Lemma 9.4.18,  $h$  satisfies

1. For every instance  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^\star \cup g, \mathcal{E}_\ell)$ .
2. For every high variance instance  $\ell \in [k]$ ,  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq 4 \cdot \text{activedegree}_{S^\star}(\ell)$ .

We now show that the desired Pareto approximation behavior is achieved when  $h$  is considered as the partial assignment in Step 0d of the algorithm. We analyze the guarantee for low and high variance instances separately.

For any low variance instance  $\ell \in [k]$ , from property 1 above, we have  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq (1 - \varepsilon/2) \cdot \text{val}(h^\star \cup g, \mathcal{E}_\ell)$ . Since we know that  $\text{val}(h^\star \cup g, \mathcal{E}_\ell) \geq (\alpha - 4\varepsilon) \cdot c_\ell$ , we have  $\text{val}(h \cup g, \mathcal{E}_\ell) \geq (\alpha - 5\varepsilon) \cdot c_\ell$ .

For every high variance instance  $\ell \in [k]$ , since  $h^* = f^*|_{S^*}$ , for any  $g$  we must have,

$$\text{val}(h^* \cup g, \mathcal{E}_\ell) \geq \text{val}(f^*, \mathcal{E}_\ell) - \text{activedegree}_{S^*}(\ell) \geq c_\ell - \text{activedegree}_{S^*}(\ell).$$

Combining this with properties 1 and 2 above, we get,

$$\begin{aligned} \text{val}(h \cup g, \mathcal{E}_\ell) &\geq (1 - \varepsilon/2) \cdot \max\{c_\ell - \text{activedegree}_{S^*}(\ell), 4 \cdot \text{activedegree}_{S^*}(\ell)\} \\ &\geq (\alpha - \varepsilon) \cdot c_\ell. \end{aligned}$$

Thus, for all instances  $\ell \in [k]$ , we get  $\text{val}(h \cup g) \geq (\alpha - 5\varepsilon) \cdot c_\ell$ . Since we are taking the best assignment  $h \cup g$  at the end of the algorithm ALG-SIM-MAXCUT, the theorem follows. ■

## 9.5 Deferred Proofs

### 9.5.1 Proof of Claim 9.4.14

We need following bounds on the gaussian random variables.

**Claim 9.5.1.** *For all  $x > 0$ ,  $\Pr_{g \sim \mathcal{N}(0,1)}[|g| > x] \leq e^{-x^2/2}$*

**Claim 9.5.2.** *For all  $1 > x > 0$ ,  $\Pr_{g \sim \mathcal{N}(0,1)}[|g| < x] \leq x$*

#### Random process $\mathcal{P}$ :

Let  $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4 \in \mathbb{R}^4$  be unit vectors and  $\mu_1, \mu_2, \mu_3, \mu_4$  be any real numbers. Consider the following random variables  $(y_1, y_2, y_3, y_4)$  where  $y_i \in \{-1, +1\}$  which are sampled as follows: Pick a random vector  $\mathbf{g} := (g_1, g_2, g_3, g_4) \in \mathbb{R}^4$  with each entry distributed as  $\mathcal{N}(0, 1)$ . Set

$$\begin{aligned} y_i &= -1 && \text{if } \langle \mathbf{g}, \mathbf{w}_i \rangle \leq \mu_i \\ &= +1 && \text{otherwise} \end{aligned}$$

The following lemmas gives sufficient conditions when  $I(y_1, y_2; y_3, y_4)$  is *small*.

**Lemma 9.5.3.** Suppose  $|\langle \mathbf{w}_i, \mathbf{w}_j \rangle| \leq \delta$  for all  $i, j \in [4]$  and  $i \neq j$ , then for all  $\mathbf{b} \in \{-1, +1\}^4$ , we have

$$\left| \Pr[(y_1, y_2, y_3, y_4) = \mathbf{b}] - \prod_{1 \leq i \leq 4} \Pr[y_i = b_i] \right| = O(\delta^{1/4}),$$

where  $y_i$  are sampled according to the random process  $\mathcal{P}$ . In fact, joint distribution on any subset of variables is close to its product distribution pointwise with an additive error of at most  $O(\delta^{1/4})$ .

*Proof.* Assume that  $0 < \delta < 1/100$  (Otherwise, the lemma is trivial). Let  $\mathbf{e}_i$  is a unit vector with 1 in the  $i^{\text{th}}$  coordinate. By rotational symmetry, we can assume that  $\langle \mathbf{w}_i, \mathbf{e}_i \rangle \geq 1 - 20\delta$  for all  $i$ . We can write vector  $\mathbf{w}_i = \sqrt{1 - \delta_i} \mathbf{e}_i + \sqrt{\delta_i} \eta_i$  where  $\eta_i$  is a unit vector. The conditions on inner products therefore imply each  $\delta_i < 40\delta$ . We will prove the lemma for  $\mathbf{b} = (-1, -1, -1, -1)$  (all other cases are similar). We have,

$$\begin{aligned} \Pr[y_i = -1, \forall i \in [4]] &= \Pr[\forall i, \langle \mathbf{g}, \mathbf{w}_i \rangle \leq \mu_i] \\ &= \Pr[\forall i, \sqrt{1 - \delta_i} g_i + \sqrt{\delta_i} \langle \mathbf{g}, \eta_i \rangle \leq \mu_i] \end{aligned}$$

Let  $B$  be the following event,

$B$  : There exists  $1 \leq i \leq 4$ , such that  $|\langle \mathbf{g}, \eta_i \rangle| \geq 1/\delta^{1/4}$ .

By union bound,

$$\Pr[B] = \sum_i \Pr[|\langle \mathbf{g}, \eta_i \rangle| \geq 1/\delta^{1/4}] \leq 4 \cdot \Pr[|\langle \mathbf{g}, \eta_1 \rangle| \geq 1/\delta^{1/4}] = 4 \cdot \Pr_{g \sim \mathcal{N}(0,1)}[|g| \geq 1/\delta^{1/4}] \leq 4e^{-\frac{1}{2\sqrt{\delta}}}$$

Now,

$$\begin{aligned} \Pr[y_i = -1, \forall 1 \leq i \in [4]] &= \Pr[B] \cdot \Pr[y_i = -1, \forall i \in [4] | B] + \Pr[\overline{B}] \cdot \Pr[y_i = -1, \forall i \in [4] | \overline{B}] \\ &\leq 4e^{-\frac{1}{2\sqrt{\delta}}} + \Pr[y_i = -1, \forall i \in [4] | \overline{B}], \end{aligned} \tag{9.5.1}$$

where last inequality uses [Claim 9.5.1](#). We now estimate the probability conditioned

on event  $\overline{B}$ .

$$\begin{aligned}
\Pr[y_i = -1, \forall i \in [4] | \overline{B}] &= \Pr[\forall i, \sqrt{1 - \delta_i} g_i + \sqrt{\delta_i} \langle \mathbf{g}, \eta_i \rangle \leq \mu_i | \overline{B}] \\
&\leq \Pr[\forall i, \sqrt{1 - \delta_i} g_i \leq \mu_i + \sqrt{\delta_i} \cdot \frac{1}{\delta^{1/4}}] \\
(g_i \text{ are independent}) \quad &= \prod_i \Pr[\sqrt{1 - \delta_i} g_i \leq \mu_i + \sqrt{\delta_i} \cdot \frac{1}{\delta^{1/4}}] \\
(\delta_i \leq 40\delta) \quad &\leq \prod_i \Pr[\sqrt{1 - \delta_i} g_i \leq \mu_i + \sqrt{40\delta^{1/4}}] \\
(\delta_i \leq 1/2) \quad &\leq \prod_i \Pr[g_i \leq (1 + \delta_i)(\mu_i + \sqrt{40\delta^{1/4}})] \\
(\delta_i \leq 1/2) \quad &\leq \prod_i \Pr[g_i \leq \mu_i + \delta_i \mu_i + 3/2 \cdot \sqrt{40\delta^{1/4}}] \\
&\leq \prod_i \Pr[g_i \leq (\mu_i + \delta_i \mu_i + 15\delta^{1/4})]
\end{aligned}$$

We now analyse the above probability in cases, and try to show the following.

$$\Pr[g_i \leq \mu_i + \delta_i \mu_i + 15\delta^{1/4}] \leq \prod_i \Pr[g_i \leq \mu_i] + O(\delta^{1/4}) \quad (9.5.2)$$

Notice that

$$\begin{aligned}
\prod_i \Pr[g_i \leq \mu_i + c\delta^{1/4}] &\leq \prod_i \Pr[g_i \leq \mu_i] + \Pr[|g_i| \leq c\delta^{1/4}] \\
(\text{Claim 9.5.2}) \quad &\leq \left( \prod_{1 \leq i \leq 4} \Pr[y_i = b_i] + c\delta^{1/4} \right) \\
&\leq \prod_{1 \leq i \leq 4} \Pr[y_i = b_i] + O(\delta^{1/4}) \quad (9.5.3)
\end{aligned}$$

- Case 1:  $\mu_i < 0$ .

In this case, we can directly say the following.

$$\prod_i \Pr[g_i \leq \mu_i + \delta_i \mu_i + 15\delta^{1/4}] \leq \prod_i \Pr[g_i \leq \mu_i + 15\delta^{1/4}]$$

- Case 2:  $\mu_i \leq \frac{10}{\delta^{3/4}}$  We can say the following because  $\delta_i < 40\delta$ .

$$\prod_i \Pr[g_i \leq \mu_i + \delta_i \mu_i + 15\delta^{1/4}] \leq \prod_i \Pr[g_i \leq \mu_i + O(\delta^{1/4})]$$

- Case 3:  $\mu_i > \frac{10}{\delta^{3/4}}$  In this case, since  $\mu_i$  is large, we have the following from [Claim 9.5.1](#).

$$\prod_i \Pr[g_i \leq \mu_i] \geq 1 - o(\delta^{1/4})$$

$$\prod_i \Pr[g_i \leq \mu_i + \delta_i \mu_i + 15\delta^{1/4}] \leq 1 = \prod_i \Pr[g_i \leq \mu_i] + o(\delta^{1/4})$$

Form (9.5.1), (9.5.2) and (9.5.3) we get

$$\Pr[(y_1, y_2, y_3, y_4) = \mathbf{b}] - \prod_{1 \leq i \leq 4} \Pr[y_i = b_i] \leq O(\delta^{1/4})$$

The other direction can be shown in an analogous way. ■

We can now bound the Mutual information between  $(y_1, y_2)$  and  $(y_3, y_4)$  if the vectors  $\mathbf{w}_i$  satisfy the condition from Lemma 9.5.3

**Lemma 9.5.4.** *Suppose  $|\langle \mathbf{w}_i, \mathbf{w}_j \rangle| \leq \delta$  for all  $i, j \in [4]$  and  $i \neq j$ , then  $I((y_1, y_2); (y_3, y_4)) \leq \text{poly}(\delta)$ , where  $y_i$  are sampled according to the random process  $\mathcal{P}$ .*

*Proof.* The lemma follows from Lemma 9.5.3 as the distribution is *close* to the product distribution.

To formally prove the lemma, first we assume that each of the random variables  $y_i$  is not heavily biased i.e.  $\Pr[y_i = -1] \in [\delta^{1/100}, 1 - \delta^{1/100}]$ . Using the definition of mutual information,

$$I((y_1, y_2); (y_3, y_4)) = \sum_{b_1, b_2, b_3, b_4 \in \{-1, 1\}} \Pr[\mathbf{y} = \mathbf{b}] \cdot \log \frac{\Pr[\mathbf{y} = \mathbf{b}]}{\Pr[(y_1, y_2) = (b_1, b_2)] \cdot \Pr[(y_3, y_4) = (b_3, b_4)]} \quad (9.5.4)$$

Form Lemma 9.5.3, we have

$$\Pr[(y_1, y_2) = (b_1, b_2)] \geq \Pr[y_1 = b_1] \Pr[y_2 = b_2] - O(\delta^{1/4})$$

$$\Pr[(y_3, y_4) = (b_3, b_4)] \geq \Pr[y_3 = b_3] \Pr[y_4 = b_4] - O(\delta^{1/4})$$

Plugging any simplifying in (9.5.4), we get

$$I((y_1, y_2); (y_3, y_4)) \leq \sum_{b_1, b_2, b_3, b_4 \in \{-1, 1\}} \Pr[\mathbf{y} = \mathbf{b}] \cdot \log \frac{\prod_{1 \leq i \leq 4} \Pr[y_i = b_i] + O(\delta^{1/4})}{\prod_{1 \leq i \leq 4} \Pr[y_i = b_i] - O(\delta^{1/4})} \quad (9.5.5)$$

As each variable is not heavily biased, we have  $\prod_{1 \leq i \leq 4} \Pr[y_i = b_i] \geq \delta^{1/25}$  and hence the log in the above expression can be upper bounded by  $\log \frac{\delta^{1/25} + O(\delta^{1/4})}{\delta^{1/25} - O(\delta^{1/4})}$  which is at most  $\log(1 + O(\delta^{1/10})) \leq O(\delta^{1/10})$ . Hence we have

$$I((y_1, y_2); (y_3, y_4)) \leq O(\delta^{1/10})$$

If a variable is heavily biased, suppose say  $y_1$  has large bias, then we can claim  $I((y_1, y_2); (y_3, y_4)) \leq \text{poly}(\delta) + I(y_2; (y_3, y_4))$  using derivation similar to (9.4.11) and then proceed by upper bounding  $I(y_2; (y_3, y_4))$  in a similar fashion as above. ■

**Proof of Claim 9.4.14:** The proof follows from Lemma 9.5.4 noting the fact that the upper bound is independent of  $\mu_i$ .

### 9.5.2 Proof of Lemma 9.4.2

*Proof.* Item 1 of the lemma follows from Chebyshev's inequality. We now focus on the proof of Item 2. We have

$$\text{Uvar}_\ell \geq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2 \Rightarrow \sum_{e \sim_S e'} \mathcal{E}_\ell(e) \mathcal{E}_\ell(e') \geq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2$$

Let  $e_0$  be an edge in  $\text{Active}(S)$  that maximizes  $\sum_{e \sim_S e_0} \mathcal{E}_\ell(e)$ . We can now upper bound the expression on the left as follows

$$\sum_{e \sim_S e'} \mathcal{E}_\ell(e) \mathcal{E}_\ell(e') \leq \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \cdot \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e)$$

Therefore, we have

$$\begin{aligned} \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \cdot \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e) &\geq \delta_0 \varepsilon_0^2 \cdot \text{Lmean}_\ell^2 \geq \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \left( \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e) \right)^2 \\ &\Rightarrow \sum_{e \sim_S e_0} \mathcal{E}_\ell(e) \geq \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e) \end{aligned}$$

Let  $v$  be the end vertex of  $e_0$  that has greater weight of active edges adjacent to it,  $v \in V \setminus S$ . We can say the following

$$\text{activedegree}_S(v, \ell) \geq \frac{1}{2} \cdot \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e)$$

From the definition of  $\text{activedegree}_S(\ell)$ , we can say the following

$$\text{activedegree}_S(\ell) \leq 2 \cdot \sum_{e \in \text{Active}(S)} \mathcal{E}_\ell(e)$$

as each edge could contribute at most twice to the sum, once for each end vertex. This gives us the following required result.

$$\text{activedegree}_S(v, \ell) \geq \frac{1}{4} \cdot \delta_0 \varepsilon_0^2 \cdot \tau^2 \cdot \text{activedegree}_S(\ell)$$

■

## References

- [ABG06] Eric Angel, Evripidis Bampis, and Laurent Gourvès. Approximation algorithms for the bi-criteria weighted max-cut problem. *Discrete Applied Mathematics*, 154(12):1685 – 1692, 2006.
- [ABG12] Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: A 0.8776-approximation for max bisection. *CoRR*, abs/1205.0458, 2012.
- [ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 563–572. IEEE, 2010.
- [AFWZ95] Noga Alon, Uriel Feige, Avi Wigderson, and David Zuckerman. Derandomized graph products. *Computational Complexity*, 5(1):60–75, 1995.
- [AKK<sup>+</sup>08] Sanjeev Arora, Subhash A Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K Vishnoi. Unique games on expanding constraint graphs are easy. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 21–28. ACM, 2008.
- [ALM<sup>+</sup>98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.
- [AM09] Per Austrin and Elchanan Mossel. Approximation resistant predicates from pairwise independence. *Comput. Complexity*, 18(2):249–271, 2009. (Preliminary version in *23rd IEEE Conference on Computational Complexity*, 2008).
- [AMS11] Christoph Ambühl, Monaldo Mastrolilli, and Ola Svensson. Inapproximability results for maximum edge biclique, minimum linear arrangement, and sparsest cut. *SIAM Journal on Computing*, 40(2):567–596, 2011.
- [AS97] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 485–495. ACM, 1997.
- [AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *J. ACM*, 45(1):70–122, January 1998.
- [BCN89] A.E. Brouwer, A.M. Cohen, and A. Neumaier. *Distance-regular graphs*. *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, 1989.

- [BDL17] Amey Bhangale, Irit Dinur, and Inbal Livni. Cube vs Cube Low Degree Test. *to appear in ITCS*, 2017.
- [BG14] Mark Braverman and Ankit Garg. Small value parallel repetition for general games. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:95, 2014. To appear in STOC 2015.
- [BGH<sup>+</sup>12] Boaz Barak, Parikshit Gopalan, Johan Håstad, Raghu Meka, Prasad Raghavendra, and David Steurer. Making the long code shorter. In *Proc. 53rd IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 370–379, 2012.
- [BGH<sup>+</sup>16] Amey Bhangale, Rajiv Gandhi, Mohammad Taghi Hajiaghayi, Rohit Khandekar, and Guy Kortsarz. Bicovering: Covering Edges With Two Small Subsets of Vertices. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55, pages 6:1–6:12, 2016.
- [BGS98] Mihir Bellare, Oded Goldreich, and Madhu Sudan. Free Bits, PCPs, and Nonapproximability—Towards Tight Results. *SIAM Journal on Computing*, 27(3):804–915, 1998.
- [BHV15] Amey Bhangale, Prahladh Harsha, and Girish Varma. A Characterization of Hard-to-cover CSPs. In *30th Conference on Computational Complexity (CCC 2015)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 280–303, 2015.
- [BK09] Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *Foundations of Computer Science, 2009. FOCS’09. 50th Annual IEEE Symposium on*, pages 453–462. IEEE, 2009.
- [BK10] Nikhil Bansal and Subhash Khot. Inapproximability of hypergraph vertex cover and applications to scheduling problems. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Proc. 37th International Colloq. of Automata, Languages and Programming (ICALP), Part I*, volume 6198 of *LNCS*, pages 250–261. Springer, 2010.
- [BKK<sup>+</sup>16] Amey Bhangale, Subhash Khot, Swastik Kopparty, Sushant Sachdeva, and Devanathan Thiruvengatachari. Near-optimal approximation algorithm for simultaneous Max-Cut. *Manuscript*, 2016.
- [BKS15] Amey Bhangale, Swastik Kopparty, and Sushant Sachdeva. Simultaneous approximation of constraint satisfaction problems. In *International Colloquium on Automata, Languages, and Programming*, pages 193–205. Springer, 2015.
- [BKT16] Amey Bhangale, Subhash Khot, and Devanathan Thiruvengatachari. Improved Dictatorship Test with Perfect Completeness. *Manuscript*, 2016.
- [BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap\*. *Combinatorica*, 26(5):495–519, 2006.

- [BLR90] M. Blum, M. Luby, and R. Rubinfeld. Self-testing/correcting with applications to numerical problems. pages 73–83, 1990.
- [BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *FOCS*, pages 472–481, 2011.
- [BS02] Thang Nguyen Bui and Lisa C. Strite. An ant system algorithm for graph bisection. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO '02*, pages 43–51, 2002.
- [BS04] B. Bollobás and A. D. Scott. Judicious partitions of bounded-degree graphs. *Journal of Graph Theory*, 46(2):131–143, 2004.
- [BSVV15] Amey Bhangale, Ramprasad Saptharishi, Girish Varma, and Rakesh Venkat. On Fortification of Projection Games. In *APPROX/RANDOM 2015*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 497–511, 2015.
- [Cha13] Siu On Chan. Approximation resistance from pairwise independent subgroups. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing, STOC '13*, pages 447–456. ACM, 2013.
- [CMM06] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Note on MAX-2SAT. *Electronic Colloquium on Computational Complexity (ECCC)*, 13(064), 2006.
- [Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.
- [DG08] Irit Dinur and Elazar Goldenberg. Locally testing direct product in the low error range. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 613–622. IEEE, 2008.
- [DG13] Irit Dinur and Venkatesan Guruswami. PCPs via low-degree long code and hardness for constrained hypergraph coloring. In *Proc. 54th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 340–349, 2013.
- [Dia11] Ilias Diakonikolas. *Approximation of Multiobjective Optimization Problems*. PhD thesis, Columbia University, 2011.
- [DK13] Irit Dinur and Gillat Kol. Covering CSPs. In *2013 IEEE Conference on Computational Complexity*, pages 207–218. IEEE, 2013.
- [DMR09] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM Journal on Computing*, 39(3):843–873, 2009.
- [DR06] Irit Dinur and Omer Reingold. Assignment testers: Towards combinatorial proofs of the PCP theorem. *SIAM Journal on Computing*, 36(4):975–1024, 2006. Special issue on Randomness and Computation.

- [DS14a] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633, 2014.
- [DS14b] Irit Dinur and David Steurer. Direct product testing. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 188–196. IEEE, 2014.
- [EH08] Lars Engebretsen and Jonas Holmerin. More efficient queries in PCPs for NP and improved approximation hardness of maximum CSP. *Random Structures & Algorithms*, 33(4):497–514, 2008.
- [Fei91] Uriel Feige. On the success probability of the two provers in one-round proof systems. In *Structure in Complexity Theory Conference, 1991., Proceedings of the Sixth Annual*, pages 116–123, Jun 1991.
- [Fei02] Uriel Feige. Relations between average case complexity and approximation complexity. In *Proceedings of the thirty-fourth annual ACM symposium on Theory of computing*, pages 534–543. ACM, 2002.
- [FGL<sup>+</sup>96] Uriel Feige, Shafi Goldwasser, László Lovász, Shmuel Safra, and Mario Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996.
- [FK94] Uriel Feige and Joe Kilian. Two prover protocols: low error at affordable rates. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 172–183, 1994.
- [FK04] Uriel Feige and Shimon Kogan. *Hardness of approximation of the balanced complete bipartite subgraph problem*, 2004.
- [FL92] Uriel Feige and László Lovász. Two-prover one-round proof systems: Their power and their problems (extended abstract). In *Proceedings of the Twenty-fourth Annual ACM Symposium on Theory of Computing, STOC '92*, pages 733–744, New York, NY, USA, 1992. ACM.
- [For89] Lance Jeremy Fortnow. *Complexity-theoretic aspects of interactive proof systems*. PhD thesis, Massachusetts Institute of Technology, 1989.
- [GHH<sup>+</sup>14] Venkat Guruswami, Prahladh Harsha, Johan Håstad, Srikanth Srinivasan, and Girish Varma. Super-polylogarithmic hypergraph coloring hardness via low-degree long codes. In *Proc. 46th ACM Symp. on Theory of Computing (STOC)*, pages 614–623, 2014.
- [GHM<sup>+</sup>11] Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. Beating the random ordering is hard: Every ordering csp is approximation resistant. *SIAM Journal on Computing*, 40(3):878–914, 2011.
- [GHS02] Venkatesan Guruswami, Johan Håstad, and Madhu Sudan. Hardness of approximate hypergraph coloring. *SIAM J. Computing*, 31(6):1663–1686, 2002. (Preliminary Version in *41st FOCS*, 2000).

- [GKSW06] Rajiv Gandhi, Samir Khuller, Aravind Srinivasan, and Nan Wang. Approximation algorithms for channel allocation problems in broadcast networks. *Networks*, 47(4):225–236, 2006.
- [GL15] Venkatesan Guruswami and Euiwoong Lee. Strong inapproximability results on balanced rainbow-colorable hypergraphs. In *Proc. 26th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)*, pages 822–836, 2015.
- [GRW11] Christian Glaßer, Christian Reitwießner, and Maximilian Witek. Applications of discrepancy theory in multiobjective approximation. *CoRR*, abs/1107.0634, 2011.
- [GS97] Oded Goldreich and Shmuel Safra. A combinatorial consistency lemma with application to proving the PCP theorem. In *RANDOM: International Workshop on Randomization and Approximation Techniques in Computer Science*. LNCS, 1997.
- [GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *FOCS*, pages 482–491, 2011.
- [GW93] Michel X. Goemans and David P. Williamson. A new  $\frac{3}{4}$ -approximation algorithm for MAX SAT. In *IPCO*, pages 313–321, 1993.
- [GW95] Michel X. Goemans and David P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, November 1995.
- [Hås01] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001. (Preliminary version in *29th STOC*, 1997).
- [Hås14] Johan Håstad. On the NP-hardness of Max-Not-2. *SIAM Journal on Computing*, 43(1):179–193, 2014.
- [HK05] Johan Håstad and Subhash Khot. Query Efficient PCPs with Perfect Completeness. *Theory of Computing*, 1(1):119–148, 2005.
- [Hol09] Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(1):141–172, 2009.
- [Hua13] Sangxia Huang. Approximation resistance on satisfiable instances for predicates with few accepting inputs. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 457–466. ACM, 2013.
- [HW03] Johan Håstad and Avi Wigderson. Simple analysis of graph tests for linearity and PCP. *Random Structures & Algorithms*, 22(2):139–160, 2003.

- [IKW12] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query PCPs. *SIAM Journal on Computing*, 41(6):1722–1768, 2012.
- [IP01] Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367 – 375, 2001.
- [IPZ01] Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, December 2001.
- [Kar72] Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.
- [Kho02a] S. Khot. On the power of unique 2-prover 1-round games. pages 767–775, 2002.
- [Kho02b] Subhash Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *Proc. 43rd IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 23–32, 2002.
- [Kho06] Subhash Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025–1071, 2006.
- [KKMO07] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O’Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, April 2007.
- [KO07] Daniela Kühn and Deryk Osthus. Maximizing several cuts simultaneously. *Comb. Probab. Comput.*, 16(2):277–283, March 2007.
- [KR08] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within  $2 - \epsilon$ . *Journal of Computer and System Sciences*, 74(3):335–349, 2008.
- [KS06] Subhash Khot and Rishi Saket. A 3-query non-adaptive PCP with perfect completeness. In *21st Annual IEEE Conference on Computational Complexity (CCC’06)*, pages 11–pp. IEEE, 2006.
- [KS14] Subhash Khot and Rishi Saket. Hardness of coloring 2-colorable 12-uniform hypergraphs with  $2^{(\log n)^{\Omega(1)}}$  colors. In *Proc. 55th IEEE Symp. on Foundations of Comp. Science (FOCS)*, pages 206–215, 2014.
- [KSTW01] Sanjeev Khanna, Madhu Sudan, Luca Trevisan, and David P. Williamson. The approximability of constraint satisfaction problems. *SIAM J. Comput.*, 30(6):1863–1920, December 2001.
- [KTW14] Subhash Khot, Madhur Tulsiani, and Pratik Worah. A characterization of strong approximation resistance. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 634–643. ACM, 2014.

- [Lev73] Leonid Anatolevich Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.
- [Mar13] Dániel Marx. Slides : CSPs and fixed-parameter tractability. <http://www.cs.bme.hu/~dmarx/papers/marx-bergen-2013-csp.pdf>, June 2013.
- [MM12] Konstantin Makarychev and Yury Makarychev. Approximation algorithm for non-boolean max k-csp. In Anupam Gupta, Klaus Jansen, Jos© Rolim, and Rocco Servedio, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, volume 7408 of *Lecture Notes in Computer Science*, pages 254–265. Springer Berlin Heidelberg, 2012.
- [MOO05] Elchanan Mossel, Ryan O’Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences invariance and optimality. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS ’05*, pages 21–30, Washington, DC, USA, 2005. IEEE Computer Society.
- [Mos10] Elchanan Mossel. Gaussian bounds for noise correlation of functions. *Geom. Funct. Anal.*, 19(6):1713–1756, 2010. (Preliminary version in *49th FOCS*, 2008).
- [Mos14] Dana Moshkovitz. Parallel repetition from fortification. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 414–423, 2014.
- [Mos15] Dana Moshkovitz. Parallel repetition from fortification. <http://people.csail.mit.edu/dmoshkov/papers/par-rep/final3.pdf>, 2015.
- [MR08] Dana Moshkovitz and Ran Raz. Sub-constant error low degree test of almost-linear size. *SIAM J. Computing*, 38(1):140–180, 2008.
- [Ole03] Krzysztof Oleszkiewicz. On a nonsymmetric version of the khinchine-kahane inequality. In *Stochastic inequalities and applications*, pages 157–168. Springer, 2003.
- [OW08] Ryan O’Donnell and Yi Wu. An optimal sdp algorithm for max-cut, and equally optimal long code tests. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, STOC ’08*, pages 335–344, New York, NY, USA, 2008. ACM.
- [OW09a] Ryan O’Donnell and Yi Wu. 3-bit dictator testing: 1 vs. 5/8. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 365–373. Society for Industrial and Applied Mathematics, 2009.
- [OW09b] Ryan O’Donnell and Yi Wu. Conditional hardness for satisfiable 3-CSPs. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 493–502. ACM, 2009.

- [PRS02] Michal Parnas, Dana Ron, and Alex Samorodnitsky. Testing basic Boolean Formulae. *SIAM Journal on Discrete Mathematics*, 16(1):20–46, 2002.
- [PY00] Christos H. Papadimitriou and Mihalis Yannakakis. On the approximability of trade-offs and optimal access of web sources. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 86–92, 2000.
- [Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th annual ACM symposium on Theory of computing*, STOC '08, pages 245–254, New York, NY, USA, 2008. ACM.
- [Rao11] Anup Rao. Parallel repetition in projection games and a concentration bound. *SIAM J. Comput.*, 40(6):1871–1891, 2011.
- [Raz98] Ran Raz. A parallel repetition theorem. *SIAM J. Computing*, 27(3):763–803, June 1998. (Preliminary version in *27th STOC*, 1995).
- [Raz11] Ran Raz. A counterexample to strong parallel repetition. *SIAM J. Comput.*, 40(3):771–777, June 2011.
- [RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.
- [RS97] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 475–484. ACM, 1997.
- [RS09] Prasad Raghavendra and David Steurer. How to round any CSP. In *In Proc. 50th IEEE Symp. on Foundations of Comp. Sci.*, 2009.
- [RT00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM J. Discrete Math.*, 13(1):2–24, 2000.
- [RT12] Prasad Raghavendra and Ning Tan. Approximating csps with global cardinality constraints using sdp hierarchies. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*, pages 373–387. SIAM, 2012.
- [RVW00] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 3–13, 2000.
- [Sch78] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, STOC '78, pages 216–226, New York, NY, USA, 1978. ACM.

- [Sjo09] Henrik Sjogren. Rigorous analysis of approximation algorithms for max 2-csp. 2009.
- [ST00] Alex Samorodnitsky and Luca Trevisan. A PCP characterization of NP with optimal amortized query complexity. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 191–199. ACM, 2000.
- [ST09] Alex Samorodnitsky and Luca Trevisan. Gowers uniformity, influence of variables, and PCPs. *SIAM Journal on Computing*, 39(1):323–360, 2009.
- [Sve10] Ola Svensson. Conditional hardness of precedence constrained scheduling on identical machines. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 745–754. ACM, 2010.
- [Tre98] L. Trevisan. Parallel approximation algorithms by positive linear programming. *Algorithmica*, 21(1):72–88, 1998.
- [TY15] Suguru Tamaki and Yuichi Yoshida. A query efficient non-adaptive long code test with perfect completeness. *Random Structures & Algorithms*, 47(2):386–406, 2015.
- [Var14] Girish Varma. A note on reducing uniformity in Khot-Saket hypergraph coloring hardness reductions. (manuscript), 2014.
- [vdWANB49] van der Waerden, E. Artin, E. Noether, and F. Blum. *Modern Algebra*. Number v. 1. Frederick Ungar Publishing, 1949.
- [Ver96] Oleg Verbitsky. Towards the parallel repetition conjecture. *Theor. Comput. Sci.*, 157(2):277–282, May 1996.
- [Wen13] Cenny Wenner. Circumventing  $d$ -to-1 for approximation resistance of satisfiable predicates strictly containing parity of width at least four. *Theory Comput.*, 9(23):703–757, 2013. (Preliminary version in *APPROX*, 2012).
- [Wol07] Pawel Wolff. Hypercontractivity of simple random variables. *Studia Mathematica*, 180(3):219–236, 2007.
- [Zwi97] Uri Zwick. Approximation Algorithms for Constraint Satisfaction Problems Involving at Most Three Variables per Constraint. In *In Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1997.