Inapproximability Reductions and Integrality Gaps

by

Preyas Popat

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

New York University

May 2013

Professor Subhash Khot

To the Courant administrative staff: for always being friendly, efficient and affectionate.

Acknowledgements

First and foremost, I thank my advisor Subhash Khot, who was the perfect mentor for me in every way I can think of.

I thank my committee members: Assaf Naor, Richard Cole, Joel Spencer and Oded Regev for their valuable time and input, especially Oded for his detailed comments which significantly improved this thesis. In addition, I thank my excellent collaborators: Rishi Saket, Venkatesan Guruswami, Ryan O'Donnell, Madhur Tulsiani, Yi Wu and Nisheeth Vishnoi for the pleasure of working with them.

I thank the Courant administrative staff, especially Rosemary, for being on top of things and shielding us PhD students from all kinds of bureaucracy.

Finally I thank my parents, brother, sister-in-law and friends for their continuous support throughout the PhD.

Abstract

In this thesis we prove intractability results for several well studied problems in combinatorial optimization.

Closest Vector Problem with Pre-processing (CVPP): We show that the pre-processing version of the well known CLOSEST VECTOR PROBLEM is hard to approximate to an almost polynomial factor ($2^{\log^{1-\epsilon} n}$) unless NP is in quasi polynomial time. The approximability of CVPP is closely related to the security of lattice based cryptosystems.

Pricing Loss Leaders: We show hardness of approximation results for the problem of maximizing profit from buyers with *single minded valuations* where each buyer is interested in bundles of at most k items, and the items are allowed to have negative prices ("Loss Leaders"). For $k = 2$, we show that assuming the UNIQUE GAMES CONJECTURE, it is hard to approximate the profit to any constant factor. For $k \geq 2$, we show the same result assuming $P \neq NP$.

Integrality gaps: We show Semi-Definite Programming (SDP) integrality gaps for UNIQUE GAMES and 2-TO-1 GAMES. Inapproximability results for these problems imply inapproximability results for many fundamental optimization problems. For the first problem, we show "approximate" integrality gaps for super constant rounds of the powerful Lasserre hierarchy. For the second problem we show integrality gaps for the basic SDP relaxation with perfect completeness.

Contents

List of Figures

Chapter 1

Introduction

In this chapter, we describe the problems of interest to us, motivation for studying them, related previous research and progress made in this thesis. The first few sections below describe the various problems while the last section summarizes the contributions of this thesis.

1.1 Closest Vector Problem with Pre-processing

An integer lattice B is a set of vectors $\{\sum_{i=1}^{n} \alpha_i b_i \mid \alpha_i \in \mathbb{Z}\}$, where $b_1, b_2, \ldots, b_n \in \mathbb{Z}^m$ are linearly independent vectors, called the *basis* of the lattice. Given (the basis of) an integer lattice B and a target vector t in $\mathbb{Z}^m$, the Closest Vector Problem (CVP) asks for the vector in B nearest to t under the ℓ_p norm. All norms $p \geq 1$ are interesting although the case $p = 2$ has received the most attention. An important variant of CVP is the pre-processing version of the problem where the lattice B is known in advance and the algorithm is allowed arbitrary pre-processing on B before the input t is revealed. This is known as the Closest Vector Problem with Pre-processing (CVPP).

A related problem is the Nearest Codeword Problem (NCP) where the input is a generator matrix C of a linear code over $\mathbb{F}_2$ and a target vector t. The goal is to find the codeword nearest to t in Hamming distance. Again, if C is known in advance and arbitrary pre-processing is allowed on it, the problem is known as the Nearest Codeword Problem with Pre-processing (NCPP).

The approximation version for all these problems with approximation factor K asks for a lattice vector (or a codeword) whose distance from the target vector t is within factor K of the minimal distance. The approximation version is interesting from both algorithmic and hardness perspective, namely designing an efficient K-approximation algorithm as well as showing that K-approximation is computationally infeasible under a reasonable complexity hypothesis, for specific approximation factors K.

Pre-processing problems arise in cryptography and coding theory where, typically, a publicly known lattice (or a linear error-correcting code) is used to transmit messages across a faulty channel. The decrypting or decoding of the received word amounts to solving an instance of CVP for this lattice. The security of the cryptosystem relies on the assumption that CVP is hard to solve even up to fairly large approximation factors. Since the lattice is known publicly and an adversary may carry out arbitrary pre-computation, it is important to understand whether the pre-computation might compromise the security of the cryptographic protocol (see [FM04, Reg04] for more details). From this perspective, it is desirable to have an inapproximability result showing that CVP remains hard to approximate even after revealing the lattice in advance.

Potentially, the pre-computed information could make CVPP much easier to approximate than CVP. Indeed, using the so-called Korkine-Zolotarev basis, Lagarias *et al.* [LLS90] designed an $O(n^{1.5})$ approximation algorithm for CVPP, which is significantly better than the best known almost-exponential $2^{O(n \log\log n/\log n)}$ approximation known for CVP [MV10, Sch87]. This was further improved to $O(n)$ by Regev [Reg04] and subsequently to $O(\sqrt{n/\log n})$ by Aharonov and Regev [AR05].

On the inapproximability side, CVP is known to be inapproximable within an almost polynomial factor [ABSS97] (i.e. factor $2^{\log^{1-\epsilon} n}$ for any constant $\epsilon > 0$. Dinur *et al.* [DKRS03] obtain an even stronger hardness factor $n^{O(1/\log\log n)}$). Obtaining inapproximability results for CVPP has been a more challenging task. Feige and Micciancio [FM04] proved a $\frac{5}{3} - \epsilon$ factor NP-hardness for NCPP for any constant $\epsilon > 0$. This was improved to $3 - \epsilon$ by Regev [Reg04]. These authors observed that a factor K hardness for NCPP implies a factor $K^{1/p}$ hardness for CVPP under the ℓ_p norm for any $1 \leq p < \infty$. Also, a hardness result in the ℓ_2 case implies essentially the same hardness result in the ℓ_p case for any $p \geq 1$ as shown by Regev and Rosen [RR06] via the norm-embedding technique.

The inapproximability results were improved in [AKKV05] who proved a factor K NP-hardness for CVPP and NCPP for any constant K and a hardness of $(\log n)^{\delta}$ for some constant $\delta > 0$ under the assumption that NP $\not\subseteq$ DTIME($2^{\mathrm{poly}(\log n)}$). They also gave another reduction which achieves a hardness factor of $(\log n)^{1-\epsilon}$ for NCPP for any constant $\epsilon > 0$. The latter reduction was under a certain hypothesis about the pre-processing version of the PCP Theorem.[1] A similar hypothesis was later proved in [FJ12] who also initiated a systematic study of various pre-processing problems. The hypothesis needed by [AKKV05] can be deduced from the work of [FJ12].

[1]The authors claimed to have a proof, but did not include it in the paper.

1.2 Pricing Loss Leaders

Consider the problem of pricing n items under an unlimited supply with m buyers. Each buyer is interested in a bundle or subset of the n items. These buyers are single minded, which means each of them has a budget and they will either buy all the items if the total price is within their budget or they will buy none of the items. The goal is to price each item with profit margin $p_1, p_2, \ldots, p_n$ so as to maximize the overall profit.

There is a flurry of work on understanding the approximability of this problem (e.g., see [GHK$^+$05, HK05, BB06, GVLSU06, BK06, DHFS06, ESZ07, BBCH07, KKMS09, ERRS09, GvLU10, GS10, GR11, Wu11]). The best approximation algorithm is an $O(\log n + \log m)$-approximation given by Guruswami, Hartline, Karlin, Kempe, Kenyon and McSherry [GHK$^+$05]. The best hardness of approximation result is a factor of $(\log n)^\epsilon$ for some $\epsilon > 0$ assuming NP $\not\subseteq$ BPTIME(2^{n^δ}) for some $0 \leq \delta \leq 1$ given by Demaine, Feige, Hajiaghayi and Salavatipour [DHFS06].

We think of the items as vertices on a graph and buyers as hyper-edges, and denote the problem described above as Vertex Pricing. The special case where each buyer is interested in bundles of at most k items is denoted as Vertex Pricing$_k$. When $k = 2$, the problem is also known in the literature as the Graph Vertex Pricing problem. Another special case is the Highway Pricing problem when the items (toll-booths) are arranged linearly on a line and each buyer (as a driver) is interested in paying for a path that consists of consecutive items. For Vertex Pricing$_k$ Balcan and Blum [BB06] give an algorithm with approximation ratio $O(k)$. In particular, for Graph Vertex Pricing, their algorithm gives a 4-approximation. On the hardness side, it is known that even the simple Graph Vertex Pricing problem is APX-hard [GHK$^+$05] and Unique Games-hard to get better than 2-approximation by Khandekar, Kimbrel, Makarychev and Sviridenko [KKMS09].[2] This UG hardness result even holds when the underlying graph is bipartite, which is tight since [BB06] give a 2-approximation for bipartite graphs. Note that a factor c hardness for $k = l$ also translates to a factor c hardness for all $k > l$ since we allow bundles with *at most* k items. The HIGHWAY PRICING problem is known to be strongly NP-hard by Elbassioni, Raman, Ray and Sitters [ERRS09] and very recently a PTAS is obtained by Grandoni and Rothvoß [GR11].

All of the above results assume the seller always prices each item with a positive profit margin. Much less is known for the problem when the seller is allowed to price some of the items below their margin cost. The motivation for pricing certain item below cost is to stimulate the sales of other more profitable items. Such a pricing strategy is also widely used in practice and these items sold at price below cost are called *"loss leaders"*. For example, a printer company may sell the printer at a low price (as the loss leader) to make more profits from selling the ink cartridges. Consider the following concrete example: there are three items A, B, C and three customers: one values $\{A\}$ at \$10 above the margin cost, one values $\{C\}$ at \$10 above the cost and one values $\{A, B, C\}$ at \$10 above the cost. By pricing A, C at \$10 above the cost and B at \$10 below the cost, the seller makes a total profit of \$30. On the contrary, if no item is allowed to price below its cost, it is easy to verify that the maximum profit of the seller is at most \$20. .

To formally study the problem of pricing loss leaders, Balcan, Blum, Chan and Hajiaghayi [BBCH07] proposed two reasonable theoretical models: the *discount model* and the *coupon model*. The *discount model* is the most direct one: it assumes that the profit the seller collects from a bundle of items is the sum of the profit

[2] UNIQUE GAMES-hard or UG-hard means NP-hard assuming the UNIQUE GAMES CONJECTURE. We refer the reader to Section 1.3.1 for details about the UNIQUE GAMES CONJECTURE

margin on each item in the bundles. One drawback of the discount model is that it does not make sense to assign a negative profit margin when the margin cost of each item is 0 (e.g, for the HIGHWAY PRICING problem). To address this, the authors also propose the *coupon model* which assumes that the profit on each bundle is at least 0; i.e., if the sum of the profit margin on each item in the bundle is negative, then the seller has profit 0 on that bundle. This model also assumes that a customer is interested in a particular set of items and will not purchase a superset even if it is cheaper, which is true for problems such as HIGHWAY PRICING where the driver is only interested in travelling a particular path and would not like to travel additional stretches to save tolls. It is shown in [BBCH07, BB06] that the maximum profit under either the coupon model or discount model can be as large as $\Omega(\log n)$ times the maximum profit when only positive profit margin prices are allowed. Such a gap of $\Omega(\log n)$ holds even for the HIGHWAY PRICING problem when all the drivers have the same valuation (budget). Given the possibility of making more profit, understanding the approximability of pricing loss leaders for GRAPH VERTEX PRICING, HIGHWAY PRICING as well as the general item pricing problem are formulated as open problems in [BB06,BBCH07]. It was shown in [Wu11] that VERTEX PRICING$_3$ is UG-hard to approximate to any constant factor when Loss Leaders are allowed.

1.3 Integrality gaps

Semi-definite Programming (SDP) has played a central role in designing approximation algorithms since it was first used in this context by [GW95]. The current best approximation to many natural computational problems is achieved by solving an SDP relaxation for the problem and rounding the vectors so obtained to an integral solution. On the other hand, existence of an *integrality gap* instance is taken as evidence that an algorithm based on LP/SDP relaxation is unlikely to give a good approximation. An integrality gap instance is a specific instance (or a family of instances) where the optimum of the LP/SDP relaxation differs significantly from the integral (i.e. true) optimum. Hence, constructing integrality gaps for optimization problems against a certain class of SDPs can serve as an important indicator for the hardness of the problem.

In this thesis we consider the problem of constructing integrality gaps for two variants of the well studied Label Cover problem which are significant for proving hardness of approximation results. We introduce the problems and review related work in the next two subsections.

1.3.1 Unique Games Conjecture

Since its introduction in 2002, the *Unique Games Conjecture* (UGC) of Khot [Kho02b] has proved highly influential and powerful in the study of probabilistically checkable proofs (PCPs) and approximation algorithms. Assuming the UGC yields many strong — and often, optimal — hardness of approximation results that we have been unable to obtain assuming only P $\neq$ NP. Perhaps the acme of this line of research so far is the work of Raghavendra [Rag08], who showed the following result:

Theorem 1.3.1. *([Rag08], informally.) Let $\mathcal{C}$ be any bounded-arity constraint satisfaction problem (CSP). Assume the* Unique Games Conjecture*. Then for a certain semidefinite programming (SDP) relaxation of $\mathcal{C}$, the SDP gap for $\mathcal{C}$ is the same as the optimal polynomial-time approximability gap for $\mathcal{C}$, up to an* *additive constant $\epsilon > 0$ which can be arbitrarily small.*

We first introduce a couple of definitions to state the UGC formally.

Definition 1.3.2. *(*LABEL COVER*) An instance $\mathcal{L}\left(G(U,V,E),[L],[K],\{\pi_e\}_{e\in E}\right)$ of* LABEL COVER *is given by a bipartite graph $G = (U,V,E)$ and for each edge $e = (u,v) \in E$, a projection $\pi_e : [L] \mapsto [K]$. A labeling to the graph consists of an assignment $A : U \to L, V \to [K]$. An edge $e = (u,v)$ is said to be satisfied by an assignment A if $\pi_e(A(u)) = A(v)$. The value of an instance is the maximum fraction of edges that can be satisfied by any assignment. We call L as the "label size" of the* LABEL COVER *instance, and denote the value of $\mathcal{L}$ as $\mathrm{Opt}(L)$.*

Definition 1.3.3. *(*UNIQUE LABEL COVER*) An instance of* LABEL COVER *is called an instance of* UNIQUE LABEL COVER *if every projection π_e corresponding to an edge e is a 1-to-1 mapping, i.e. a permutation. In this case, we can take $L = K$.*

Notice that given an instance of UNIQUE LABEL COVER with value 1, it is easy to find a labeling satisfying every edge in polynomial time. For historical reasons, the UNIQUE LABEL COVER problem is also known in the literature as the UNIQUE GAMES problem.

Conjecture 1.3.4. *(*UNIQUE GAMES CONJECTURE*) [Kho02b]*

For every $\epsilon > 0$ there is a $K = K(\epsilon)$ large enough such that given a UNIQUE LABEL COVER *instance with label size K it is NP-hard to distinguish between the cases,*

- *The value of the instance is at least $1 - \epsilon$*
- *The value of the instance is at most ϵ*

Despite significant work, the status of the Unique Games Conjecture is unresolved. Several approximation algorithms for Unique Games have been developed in an attempt to refute the conjecture [Kho02b, Tre05, GT06, CMM06]. All these algorithms are based on LP or SDP relaxation and find a near satisfying assignment to a Unique Games instance if there exists one. However their performance deteriorates as the number of labels and/or the size of the instance grows, and therefore they fall short of disproving the UGC. On the other hand, Khot and Vishnoi [KV05] give a strong integrality gap for a basic SDP relaxation of the Unique Games problem (the algorithmic result of Charikar, Makarychev, and Makarychev [CMM06] essentially matches this integrality gap).

In addition to providing evidence towards the validity of the UGC, SDP gaps for Unique Games have served another important role: they are the starting points for strong SDP gaps for other important optimization problems. A notable example of this comes in the work of Khot and Vishnoi [KV05] who used the UG gap instance to construct a super-constant integrality gap for the Sparsest Cut-SDP with triangle inequalities, thereby refuting the Goemans-Linial conjecture that the gap was bounded by $O(1)$. They also used this approach to show that the integrality gap of the Max-Cut SDP remains 0.878 when triangle inequalities are added. Indeed the approach via Unique Games remains the *only known* way to get such strong gaps for Max Cut. Recently, even stronger gaps for Max-Cut were shown using this framework in [KS09, RS09]. Another example of a basic problem for which a SDP gap construction is only known via the reduction from Unique Games is Maximum Acyclic Subgraph [GMR08].

In view of these results, it is fair to say that SDP gaps for Unique Games are significant unconditionally, regardless of the truth of the UGC. Thus, it is worthwhile to investigate whether stronger LP/SDP relaxations help for problems like Unique Games, Maximum Cut or Sparsest Cut. One can obtain stronger relaxations by adding (say polynomially many) natural constraints that an integral solution must satisfy.

Natural families of constraints considered in literature include the Lovász-Schrijver LP and SDP heirarchies, the Sherali-Adams LP heirarchy, and Lasserre SDP heirarchy. Instead of attempting a complete survey of known results, we refer the reader to the relevant papers [ABLT06, STT07b, STT07a, GMPT07, Sch08, CMM09, RS09, KS09]. and focus on the results pertaining to the Sherali-Adams and Lasserre heirarchies. The t-round Sherali-Adams LP hierarchy enforces the existence of local distributions over integral solutions. Specifically, a solution to such an LP gives a distribution over assignments to every set of at most t variables and the distributions over pairwise intersecting sets are consistent on the intersection. Strong lower bounds have been obtained by Charikar, Makarychev, and Makarychev [CMM09] for up to n^{δ} rounds of Sherali-Adams relaxation for the Maximum Cut problem. Their result shows $2-\epsilon$ gap for Maximum Cut, and since the gap of the basic SDP relaxation is at most α_{GW}^{-1}, their result shows that even a *large* number of rounds of the Sherali-Adams hierarchy fail to capture the power of the basic SDP. In recent work, Raghavendra and Steurer [RS09] have obtained integrality gaps for a combination of a basic SDP and $(\log\log n)^{\Omega(1)}$ rounds of the Sherali-Adams LP: they obtain a strong gap for Unique Games, $\alpha_{GW}^{-1}-\epsilon$ for Maximum Cut and $(\log\log n)^{\Omega(1)}$ for Sparsest Cut. Simultaneously, Khot and Saket [KS09] also obtained similar but quantitatively weaker results.

One may also consider the t-round Lasserre SDP hierarchy [Las01] which introduces a SDP vector for every subset of variables of size at most t and each integral assignment to that subset. Appropriate consistency and orthogonality constraints are also added. As it turns out, a vector solution to the t-round Lasserre SDP also yields a solution to the t-round Sherali-Adams LP, and therefore the Lasserre SDP is at least as powerful as the Sherali-Adams LP.

Currently, we know very few integrality gap results for the Lassere hierarchy. Schoenebeck [Sch08] obtained Lasserre integrality gap for Max-3-Lin and Tulsiani extended it to Max-k-CSP, and also obtained a gap of 1.36 for Vertex Cover. However, we already know corresponding NP-hardness results, e.g. Håstad's [Hås01] hardness result for Max-3-Lin and Dinur and Safra's 1.36 hardness result for Vertex Cover. Indeed Tulsiani's integrality gap for Vertex Cover follows by *simulating* the Dinur-Safra reduction. It would be very interesting to have Lasserre gaps where we only know UGC-based hardness results, e.g. $2-\epsilon$ for Vertex Cover, $\alpha_{GW}^{-1}-\epsilon$ for Maximum Cut, and a superconstant gap for Sparsest Cut. Currently, such gaps are not known even for the fifth level of Lassere hierarchy, leaving open the tantalizing possibility that a constant round Lasserre SDP relaxation might give better approximations to these problems, and consequently disprove the UGC.

1.3.2 2-to-1 Games Conjecture

As mentioned in Section 1.3.1, the Unique Games Conjecture has strong implications for the approximability of many fundamental optimization problems. Unfortunately, because of the additive ϵ term in Theorem 1.3.1, Raghavendra's work is not applicable (even granting the UGC or any related conjecture) for the important case of completely *satisfiable* CSPs; equivalently, PCPs with *perfect completeness*. A good example of this comes from *coloring problems*; e.g., the very well known problem of coloring 3-colorable graphs. The UGC does not help in deducing any hardness result for such problems. Indeed the first strong hardness result for it, due to Dinur, Mossel, and Regev [DMR09], used instead certain variants of UGC which have perfect completeness, namely, the "2-to-1 Conjecture", the "2-to-2 Conjecture", and the "α-Constraint Conjecture". An instance of Label-Cover with α-constraints was also implicit in the result of Dinur and Safra [DS05] on the hardness of approximating minimum vertex cover. Recently, several more works have needed to use these alternate conjectures with perfect completeness: e.g., O'Donnell and Wu [OW09] and Tang [Tan09] on Max-3CSP, Guruswami and Sinop [GS09] on Max-k-Colorable-Subgraph.

We describe the 2-TO-1 GAMES CONJECTURE below, the other two conjectures will be described in Chapter 4.

Definition 1.3.5. *(*2-TO-1 LABEL COVER*) An instance of* LABEL COVER *is called an instance of* 2-TO-1 LABEL COVER *if every projections π_e corresponding to an edge e is a* 2*-to-*1 *mapping. In this case we can take $L = 2K$.*

Similar to UNIQUE LABEL COVER, the 2-TO-1 LABEL COVER problem is also known in the literature as the 2-TO-1 GAMES problem.

Conjecture 1.3.6. *(*2-TO-1 GAMES CONJECTURE*) [Kho02b]*

For every $\epsilon > 0$ there is a $K = K(\epsilon)$ large enough such that given a 2-TO-1 LABEL COVER *instance with label size K it is NP-hard to distinguish between the cases,*

- *The value of the instance is* 1 *i.e. there is a labeling satisfying all edges*
- *The value of the instance is at most ϵ*

Given the importance of 2-to-1 and related conjectures in reductions to satisfiable CSPs and other problems like coloring where perfect completeness is crucial, SDP gaps for 2-TO-1 LABEL COVER and variants are worthy of study even beyond the motivation of garnering evidence towards the associated conjectures on their inapproximability.

1.4 Contributions of this Thesis

1.4.1 Hardness of Approximating the Closest Vector Problem with Pre-processing

In this thesis we prove that unless NP is in quasi-polynomial time, the CVPP problem is hard to approximate to an almost polynomial factor. This appears as a joint work with Subhash Khot and Nisheeth Vishnoi [KPV12] and the proof is given in Chapter 2.

Theorem 1.4.1. *Unless* $NP \subseteq DTIME(2^{\log^{O(1/\epsilon)} n})$, NCPP *and* CVPP *are hard to approximate to a factor within* $2^{\log^{1-\epsilon} n}$ *for an arbitrarily small constant* $\epsilon > 0$.

This improves on the previous hardness factor of $(\log n)^\delta$ for some $\delta > 0$ due to [AKKV05] and essentially matches the almost polynomial factor inapproximability of Dinur *et al.* [DKRS03] for CVP. We emphasize that unlike the case of CVP where the best approximation algorithm achieves a factor of $2^{n \log \log n / \log n}$, the best approximation algorithm for CVPP achieves an approximation factor of $O(\sqrt{n/\log n})$.

1.4.2 Hardness of pricing Loss Leaders

In this thesis we prove strong hardness of approximation results for Vertex Pricing$_k$ and Highway Pricing problems. The results are stated in the next few paragraphs. These appear as a joint work with Yi Wu [PW12] and the proofs are given in Chapter 3.

For the Graph Vertex Pricing problem (aka Vertex Pricing$_2$) we prove the following theorem.

Theorem 1.4.2. Graph Vertex Pricing *under the coupon model is UG-hard to approximate to any constant factor, even when the graph is bipartite.*

Next, we prove a simple lemma which relates the approximability of Highway Pricing to the approximability of Graph Vertex Pricing on a bipartite graph.

lemma 1.4.3. *If* Graph Vertex Pricing *on bipartite graphs is hard to approximate to factor α under the coupon model, then* Highway Pricing *is also hard to approximate to factor α under the coupon model.*

Combining Theorem 1.4.2 and Lemma 1.4.3 we get the following hardness of approximation result for the Highway Pricing problem.

Corollary 1.4.4. Highway Pricing *under the coupon model is UG-hard to approximate to any constant factor.*

For the Vertex Pricing$_3$ problem we prove the following theorem.

Theorem 1.4.5. Vertex Pricing$_3$ *under the coupon or the discount model is NP-hard to approximate to factor $\Omega(\log\log\log n)$.*

It is instructive to compare our hardness results with the known approximation algorithms for the corresponding problem using positive profit margin prices only. For the general pricing problem, there is a 4-approximation algorithm when $k = 2$ and $\frac{1}{3e}$-approximation algorithm for $k = 3$ [BB06]. As for the highway problem, there exists a PTAS [GR11]. All of the three problems have (at least) a constant approximation algorithm for positive profit margin prices while our corresponding hardness results for pricing loss leaders are (at least) super-constant. Conceptually, our results indicate that the problem of pricing loss leaders is substantially harder.

1.4.3 Integrality gap for 2-to-1 Label Cover

In this thesis, we show the following theorem on the limitations of the basic semidefinite programming relaxation for 2-to-1 Label Cover.

Theorem 1.4.6. *There are instances of* 2-to-1 Label Cover *with alphabet size* K *and optimum value* $O(1/\sqrt{\log K})$ *on which the SDP has value* 1. *The instances have size* $2^{\Omega(K)}$.

This appears as a joint work with Venkatesan Guruswami, Subhash Khot, Ryan O'Donnell, Madhur Tulsiani and Yi Wu [GKO+10] and the theorem is proved in Chapter 4. We also prove similar results for 2-to-2 Label Cover and α Label Cover which are stated and proved in Chapter 4.

We note that if we only require the SDP value to be $1 - \epsilon$ instead of 1, then integrality gaps for all these problems easily follow from gaps for Unique Games constructed by Khot and Vishnoi [KV05] (by duplicating labels appropriately to modify the constraints). However, the motivation behind these conjectures is applications where it is important that the completeness is 1. Another difference between the 2-TO-1 LABEL COVER and the UNIQUE LABEL COVER is the fact that for 2-TO-1 LABEL COVER instances, it is consistent with known algorithmic results of [CMM06] that OPT be as low as K^{-c} for some $c > 0$ independent of ϵ, when the SDP value is $1 - \epsilon$. It is an interesting question if OPT can indeed be this low even when the SDP value is 1. Our constructions do not address this question, as we only show $\mathrm{OPT} = O(1/\sqrt{\log K})$.

We also point out that our integrality gaps are for special cases of the LABEL COVER problem where the constraints can be expressed as difference equations over $\mathbb{F}_2$-vector spaces. For example, for 2-TO-2 LABEL COVER, each constraint ϕ_e is of the form $x - y \in \{\alpha, \alpha + \gamma\}$ where $\alpha, \gamma \in \mathbb{F}_2^k$ are constants. For such constraints, the problem of deciding whether an instance is completely satisfiable ($\mathrm{OPT} = 1$) or not ($\mathrm{OPT} < 1$) is in fact in P. To see this, one can treat the coordinates $(x_1, \ldots, x_k)$ and $(y_1, \ldots, y_k)$ as separate boolean variables and introduce an auxiliary boolean variable z_e for each constraint. We can then rewrite the constraint as a conjunction of linear equations over $\mathbb{F}_2$: $\left[\bigwedge_{i=1}^{k} (x_i - y_i - z_e \cdot \gamma_i = \alpha_i)\, .\right]$ Here $x_i, y_i, \alpha_i, \gamma_i$ denote the i^{th} coordinates of the corresponding vectors. Deciding whether a system of linear equations is completely satisfiable is of course in P. Alternatively, one can note that constraints $x - y \in \{\alpha, \alpha + \gamma\} \bmod \mathbb{F}_2^k$ are *Mal'tsev constraints*, and hence deciding satisfiability of CSPs based on them is in P by the work of Bulatov and Dalmau [BD06].

Despite this tractability, the SDPs fail badly to decide satisfiability. This situation is similar to the very strong SDP gaps known for problems such as 3-XOR (see [Sch08], [Tul09]) for which deciding complete satisfiability is easy.

1.4.4 Approximate Lasserre integrality gap for Unique Label Cover

In this thesis, we make partial progress towards constructing Lasserre integrality gaps for Unique Label Cover. This appears as a joint work with Subhash Khot and Rishi Saket [KPS10]. We describe our result informally in the next few paragraphs and give a formal statement and proof in Chapter 5.

We show that if the constraints of a t-round Lasserre SDP are allowed to have a tiny but non-zero error $\delta > 0$, then a strong integrality gap exists for the Unique Games problem. Using standard reductions from Unique Games, similar integrality gaps can be obtained for Max-Cut, Vertex Cover etc. (we omit the details in this thesis). In fact the error can be made as small as desired independent of other parameters (except the size of the instance). All recent integrality gap constructions involving Sherali-Adams LP (see [CMM09, RS09, KS09]) first construct such approximate solutions followed by an error-correction step. However correcting Lasserre vector solution seems challenging (due to a global constraint of positive definiteness) and we leave this as an open problem. On the other hand, our result does demonstrate that a Lasserre SDP relaxation will not give good approximation if it is insensitive to a tiny perturbation of the vector solution. At the time of publication of our work, all known algorithms fell into this category. Subsequent to our work, an algorithm which uses the full power of the Lasserre hierarchy was developed by [GS11].

Our integrality gap instance is based on the integrality gap instance of [KV05]. Subsequent to our work, it was shown in [BBH$^+$12] that such instances cannot be used to construct an exact integrality gap for super constant rounds of the Lasserre hierarchy.

Chapter 2

Hardness of Approximating the Closest Vector Problem with Pre-processing

In this chapter we prove Theorem 1.4.1, restated below.

Theorem. *(Theorem 1.4.1 Restated) Unless* $NP \subseteq DTIME(2^{\log^{O(1/\epsilon)} n})$, NCPP *and* CVPP *are hard to approximate to a factor within* $2^{\log^{1-\epsilon} n}$ *for an arbitrarily small constant* $\epsilon > 0$.

In the next section, we provide a high level overview of our reduction.

2.1 Overview of the proof

We show a hardness factor of $2^{\log^{1-\epsilon} n}$ for NCPP. As mentioned in Section 1.1, a factor C hardness for NCPP implies a factor $\sqrt{C}$ hardness for CVPP under the l_2 norm hence this suffices to prove the theorem. For the sake of presentation, we find it more convenient to consider the Minimum Weight Solution Problem with Pre-processing (MWSPP) which is simply a reformulation of NCPP. The input to this problem consists of a set of *fixed* linear forms described by $B_f \in \mathbb{F}_2^{l \times N}$, a set of *variable* linear forms $B_v \in \mathbb{F}_2^{l' \times N}$ and a target vector $t \in \mathbb{F}_2^l$. The goal is to find a solution $x \in \mathbb{F}_2^N$ to the system $B_f x = t$, which minimizes the Hamming weight of the vector $B_v x$. We allow arbitrary pre-processing on all parts of the input except the vector t. The equivalence of MWSPP with NCPP is shown in Section 2.2.1. We henceforth focus on the MWSPP problem.

The authors in [AKKV05] present two reductions to MWSPP. The first one is a direct reduction from the hyper-graph vertex cover problem from [DGKR05] whereas the second one is a reduction from the pre-processing version of the PCP Theorem (proved later in [FJ12] as mentioned before). Our reduction builds on the second one.

We first quickly elaborate on the PCP Theorem and its pre-processing version. The PCP Theorem [FGL+96, AS98, ALM+98] is a fundamental result in the area of inapproximability and serves as a starting point for almost all hardness reductions. The theorem may be stated from two equivalent viewpoints, a *proof checking viewpoint* and a *reduction viewpoint*. The equivalence between the two viewpoints, thought not difficult to see, has led to many illuminating insights and exciting research over the last two decades. From the proof checking viewpoint, the PCP Theorem states that there is a way to write a proof for an NP-statement such that its correctness can be verified by a probabilistic verifier that uses logarithmic randomness and a constant number of queries to the proof. The proof system is *complete* in the sense that a correct proof of a correct statement is accepted with probability 1 and is also *sound* in the sense that any proof of an incorrect statement is accepted with probability at most s for some constant s strictly less than 1 (this probability can be reduced to any small constant by running the verifier $O(1)$ times). From the reduction viewpoint, the PCP Theorem states that there is a polynomial time reduction from any NP-complete language to $\mathsf{GapCSP}_{1,s}$, a promise problem with a set of boolean variables and constraints such that every constraint depends on $O(1)$ variables and the instance is guaranteed to be either satisfiable or no assignment satisfies more than a fraction s of the constraints.

When the PCP Theorem is stated as NP-hardness of GapCSP, the constraints may be taken as 3SAT constraints or, as will be convenient for us, quadratic equations over $\mathbb{F}_2$, with each equation involving exactly three variables. Let the variables be $z_1, \ldots, z_n$ and the constraints be $C_1, \ldots, C_m$ where the j^{th} constraint is of the form $p_j(z_{j_1}, z_{j_2}, z_{j_3}) = v_j$. Here p is a quadratic polynomial over $\mathbb{F}_2$ and $v_j \in \mathbb{F}_2$. In the pre-processing version of the CSP, arbitrary pre-processing is allowed on the polynomials $\{p_j\}_{j=1}^m$ and only the *right hand sides* of the equations, namely $\{v_j\}_{j=1}^m$ is the actual input. The pre-processing version of the PCP Theorem then states that even the pre-processing version of $\mathsf{GapCSP}_{1,s}$ is NP-hard. To be more precise, there is a polynomial time reduction from any NP-complete language L that maps input x to an instance $\{(p_j, v_j)\}_{j=1}^m$ of the $\mathrm{GapCSP}_{1,s}$ such that the set of polynomials $\{p_j\}_{j=1}^m$ depends only on the size of x and not on x itself (the right hand sides $\{v_j\}_{j=1}^m$ of course depend on x).

As we said, the PCP Theorem serves as a starting point for almost all hardness reductions. A vast majority of these reductions begin by reducing the GapCSP instance given by the PCP Theorem to a problem called Label Cover (defined originally in [ABSS97]) and then amplifying the hardness of Label Cover via Raz's Parallel Repetition Theorem [Raz98]. An instance $G = (V, W, E, [R], [S], \{\pi_e\}_{e \in E})$

of LABEL COVER is given by a bipartite graph $G(V, W, E)$ and for each edge $e = (v, w) \in E$, a function $\pi_e : [R] \to [S]$. A labeling to the graph consists of an assignment $A : V \to [R],\ W \to [S]$. An edge $e = (v, w)$ is said to be satisfied by an assignment A if $\pi_e(A(v)) = A(w)$. The value of an instance is the maximum fraction of edges that can be satisfied by any labeling. The PCP Theorem and the Parallel Repetition Theorem together imply that for every constant R, given an instance of LABEL COVER it is NP-hard to distinguish whether the value of the instance is 1 or at most $R^{-\gamma}$ (called the *soundness*) for some absolute constant $\gamma > 0$.

When the pre-processing version of the PCP Theorem is used as a starting point, one obtains the hardness of approximation for the LABEL COVER PROBLEM WITH PRE-PROCESSING (LCPP), defined in [AKKV05]. In the LCPP problem, the label set $[R]$ for each vertex $v \in V$ comes with a designated partition, and an *allowable set* from the partition. The vertices in V are required to receive labels only from their respective allowable sets. Pre-processing is allowed on all parts of the LCPP instance except for (the choice of) the allowable set for each vertex $v \in V$.

The authors in [AKKV05] present a reduction from LCPP to MWSPP. The reduction uses constructions of LABEL COVER with an additional property called *smoothness*. An instance of LABEL COVER (with or without pre-processing) is called δ-smooth if any two labels $i \neq i'$ of $v \in V$ map to different labels of $w \in W$ with probability at least $1 - \delta$ over the choice of a neighbor w of v. The smoothness property was introduced in [Kho02a] and has been used for several hardness of approximation reductions [FGRW09, GRSW10, KS11]. The hardness factor achieved by the reduction from LCPP to MWSPP is upper bounded by $1/\delta$

and $1/s$ where δ is the smoothness parameter and s is the soundness of the LCPP instance. The reduction of [AKKV05] fails to give a hardness factor better than $(\log n)^{1-\epsilon}$ for MWSPP because they use construction of Label Cover which requires size $n^{\Omega(1/\delta)}$ to ensure δ-smoothness.

To get a better hardness factor using this reduction, we require hardness of LCPP with *very good* smoothness and soundness simultaneously (relative to the size of the instance). This is exactly our main technical contribution, except that we are able to show this only for a hyper-graph variant of label cover (the underlying structure is a multi-layered hyper-graph instead of a two-layered, i.e. bipartite, graph). This is sufficient for our purpose since the reduction from LCPP to MWSPP can easily be modified so as to start with the hyper-graph variant. This new variant is named as the Hyper-graph Label Cover with Preprocessing (HLCPP). It is a labeling problem just as LCPP but differs from the latter in the following respects (see Definition 2.3.5 for a formal description).

- The vertex set is multi-layered instead of two-layered.
- The constraints are given by hyper-edges rather than edges. A hyper-edge e in itself contains several edges between pairs of variables inside e and the constraint associated with the hyper-edge e is a boolean AND of constraints on all the edges inside it.
- The constraints associated to edges are more general *many-to-many* constraints instead of the *many-to-one* (projection) constraints as in LCPP.

Our reduction is a reworking of the (original algebraic) proof of the PCP Theorem. More specifically, the proof of the PCP Theorem (see for instance Arora's thesis [Aro94]) can be broken into three phases where the number of queries the verifier makes is successively reduced to $\text{poly}(\log n)$, $\text{poly}(\log\log\log n)$ and finally to $O(1)$. In the first phase, one rewrites an instance of an NP-complete language, say 3SAT, as an algebraic CSP (say quadratic equations over a finite field), the PCP proof consists of a *polynomial encoding* (i.e. Reed-Muller encoding) of a supposed satisfying assignment to the CSP, and then the verification procedure consists of the *Low Degree Test* to check that the given proof is indeed a valid encoding of some assignment and the *Sum Check Protocol* to check that the assignment indeed satisfies the CSP.

Our reduction is a reworking of this first phase with two additional ingredients: firstly, we need to work out each step in its pre-processing version, and in particular start by proving the basic (i.e. without gap) NP-hardness for a pre-processing version of an algebraic CSP (this was also proved in [AKKV05]). Secondly, the quantitative setting of parameters is quite different from that in the proof of the PCP Theorem. The proof of the PCP Theorem encodes an n-bit assignment by a table of values of a function $f : \mathbb{F}_q^m \to \mathbb{F}_q$ where $|\mathbb{F}_q| = \text{poly}(\log n)$. On the other hand, we use a much larger field size $|\mathbb{F}_q| = n^{\text{poly}(\log n)}$. Also, in a typical CSP instance, the number of constraints is comparable (i.e. linear or polynomial) to the number of variables. On the other hand, we in the very first step, blow up the number of constraints to $n^{\text{poly}(\log n)}$ so that the soundness is roughly the inverse of the number of constraints and hence much lower than the inverse of the number of variables.

Just like the (first phase of the) proof of the PCP Theorem, we end up with a PCP with $O(\log n)$ queries (each query is a block of $\text{poly}(\log n)$ bits). It is easily observed that the proof can be partitioned into $O(\log n)$ layers and that each PCP test reads one query from each of the layers except the last and two queries from the last layer. This naturally leads to the hyper-graph variant of the label cover problem. We would like to point out that in principle the last two phases of the proof of the PCP Theorem can also be worked out similarly (along with the appropriate quantitative setting of parameters), yielding a full proof of the pre-processing version of the PCP Theorem. However we refrain from such an attempt since we do not need it and more importantly, [FJ12] already give a full (and much less tedious) proof via Dinur's proof of the PCP Theorem [Din07]. We now give a more detailed and technical outline of our reduction and its analysis below.

We start with an instance of $\mathbb{F}_q$-Quadratic Constraint Satisfaction Problem ($\mathbb{F}_q$-QCSP) for $q = 2^r$. The instance consists of k homogeneous degree 2 polynomial equations over $\mathbb{F}_q$ with n variables, where $k = \text{poly}(n)$. Each equation is of the form $p(z_1, \ldots, z_n) = v$, and further, depends on at most 3 variables. It can be shown that deciding if there is an assignment which satisfies all the equations is NP-hard (see Theorem 2.2.3), even when the left hand sides of these equations (i.e. the polynomials p) are available for pre-processing. We denote the pre-processing version by $\mathbb{F}_q$-QCSPP. Our first step is to boost soundness, i.e., to reduce the fraction of satisfied equations by any assignment, while keeping the number of variables small. This is done by combining an instance of $\mathbb{F}_q$-QCSPP with an appropriate Reed-Muller code over q. We will eventually set $q = n^{\log^{O(1/\epsilon)} n}$. This allows us to construct an $\mathbb{F}_q$-QCSPP instance where it is hard to distinguish between perfectly satisfiable instances and those where any assignment satisfies at most k/q fraction of the polynomial equations. An important feature of this reduction is that the variable set remains the same, so the number of variables is

n, number of equations is q and the soundness is k/q (which is essentially same as $1/q$). This (somewhat strange) quantitative setting of parameters is crucial for our result as the number of variables becomes negligible compared to the number of equations, and the reciprocal of the soundness. The details of this reduction appear in Section 2.2.2.

Each equation can now depend on almost all of the n variables and the next task is to deal with this. This is done by reducing checking an assignment for such a system of polynomial equations to the task of constructing a PCP which makes $O(\log n)$ queries and has soundness $1/q^e$ for some small constant $e > 0$. This is achieved by combining the low degree test and the sum check protocol and is the technical heart of the PCP construction.

First, the variables are identified with $\{0,1\}^{\log n}$ and embedded as a sub-cube of $\mathbb{F}_q^m$ where $m \stackrel{\text{def}}{=} \log n$. With this mapping, any assignment can be thought of as a function from $\{0,1\}^m$ to $\mathbb{F}_q$ and can be encoded as a polynomial over $\mathbb{F}_q^m$ of degree at most m. In this setting, if the equation was $\sum_{i,j\in[n]} c(i,j)z_iz_j = v = \sum_{\alpha,\beta\in\{0,1\}^m} c(\alpha,\beta)z(\alpha)z(\beta)$; z, c can be thought of as polynomials of degree at most m and $2m$ respectively. The Arora-Sudan points-vs-lines low degree test can be employed to ensure that z corresponds to a small list of degree m polynomials (assignments). This test is able to list-decode an assignment with success probability as low as $1/q^e$ for some small constant $e > 0$.

Once an assignment for the variables can be decoded, the task of verifying the polynomial equations $\sum_{\alpha,\beta\in\{0,1\}^m} c(\alpha,\beta)z(\alpha)z(\beta) = v$ is equivalent to performing a weighted sum check over the sub-cube $\{0,1\}^m$. We use the sum-check protocol of [LFKN92] to verify that the decoded assignment satisfies the equations. It can be shown that the soundness of the combined low degree test and the sum check protocol is at most $1/q^f$ for a small constant $f > 0$.

The result is a PCP with $2m + 2 = O(\log n)$ layers where the first $2m$ layers correspond to the sum check protocol while the last two layers correspond to the *lines* and the *points* table (in the low degree test) respectively. Only the values to be assigned to the first layer will depend on the right hand sides of the $\mathbb{F}_q$-QCSPP instance. Further, the use of low degree polynomials in encoding the assignments implicitly gives our PCP *smoothness* properties which are used in the final reduction. While the preliminaries of the low degree test and the sum check protocol appear in Sections 2.2.3 and 2.2.4 respectively, the PCP construction appears in Section 2.3.1.

This view of the PCP naturally leads to constructing an HLCPP instance which is the starting point of the reduction to MWSPP and appears in Section 2.3.2. The reduction from HLCPP to MWSPP is similar to the reduction of [AKKV05] from LCPP to MWSPP. This appears in Section 2.3.3. For the reduction to work, we define a notion of smoothness for HLCPP which is similar to the one for LCPP and we also need that the hyper-edges of the hyper-graph satisfy a uniformity condition which is inherited from the PCP construction.

The main differences in our reduction compared to the reduction of [AKKV05] are the following:

- As mentioned earlier, the constraints in the HLCPP graph are many-to-many constraints rather than many-to-one constraints. However, the earlier reduction to MWSPP still goes through in a relatively straightforward manner.

- We manage to construct an instance of HLCPP where the smoothness and soundness are both at most $1/q^f$ for some absolute constant $f > 0$ and the size of the instance is $q^{O(m)}$. Here $m = \log n$ where n is the number of variables in the original $\mathbb{F}_q$-QCSPP instance. It is not clear that such constructions are possible if we stick to the LCPP problem. The hardness factor can be made essentially as large as $q^{1/m}$ and we set q to be very large compared to m to get a good hardness factor relative to the size of the instance. Specifically, we set $q = n^{\log^{O(1/\epsilon)} n}$.

We believe the PCP constructed in the course of our proof is of independent interest and we state its properties in the theorem below. This theorem can be inferred from the construction in Section 2.3.1.

Theorem 2.1.1. *For any NP-complete language L there is a PCP with the following properties:*

1. *For an input of size n, the PCP verifier uses $O(\log n \cdot \log q)$ random bits and makes $O(\log^2 n)$ queries into the proof. The answer to each query is an element of $\mathbb{F}_q$. Here q can be chosen as any power of two such that $q \geq n^d$ for an absolute constant $d > 0$.*

2. *The acceptance predicate of the verifier involves $O(\log n)$ linear equations (of the form $\sum_i a_i x_i = b$ where $a_i, b \in \mathbb{F}_q$), and one quadratic equation in the symbols it reads. The verifier accepts if and only if all the equations are satisfied. Furthermore, the equations depend only on n, except for the r.h.s. in one linear equation which may depend on the input instance.*

3. *If the input instance is in L then there is a proof which the verifier accepts with probability 1.*

4. If the input instance is not in L then, for every proof, the verifier accepts with probability at most $1/q^c$ for some absolute constant $c > 0$.

2.1.1 Organization

The rest of the chapter is organized as follows.

In Section 5.3, we formally define the problems and state existing tools and results which will be useful in our reduction. In Section 2.2.1 we define the problems considered and summarize known reductions about them. Section 2.2.2 states a lemma about boosting the soundness of a particular CSP using codes. Sections 2.2.3 and 2.2.4 are devoted to introducing the tools of Low Degree Test and Sum Check Protocol respectively.

In Section 5.4.2 we present our reduction. The beginning of Section 5.4.2 describes how that section is organized.

Finally, Section 2.4 contains proofs of certain lemmas and theorems omitted from the main body of the chapter.

2.2 Preliminaries

In this section we state the problems we will consider and state basic results which will be useful in the construction of our PCP and the reduction.

2.2.1 Problem Definitions and Basic Results

We first formalize the notion of computational problems which allow pre-processing on a part of the input. A part of this formalism is taken from [AKKV12].

Pre-processing problems. Consider a problem Π where the input is split into two components (A, B) and the length of each component is polynomial in the size parameter n. In the pre-processing version, denoted by $\Pi\mathsf{P}$ (suffix of P to Π to emphasize the pre-processing), we consider subproblems where the first component A depends only on n and is called as the *fixed* input.

An algorithm which *solves* $\Pi\mathsf{P}(A, \cdot)$ *with pre-processing* is a polynomial time algorithm that solves the instance $\Pi\mathsf{P}(A, \cdot)$ given a polynomial sized advice. The advice captures arbitrary computation or pre-processing on the fixed input A and polynomial amount of stored information. The pre-processing version $\Pi\mathsf{P}(A, \cdot)$ is called NP-hard if there is a polynomial time reduction from SAT to $\Pi\mathsf{P}(A, \cdot)$ such that the fixed input A depends only on the size of the SAT instance.

In the definitions that follow, we will not explicitly split the input as (A, B) but we will indicate for each problem the part of the input on which pre-processing is allowed. We first define the quadratic CSP problem and its pre-processing version that will be a starting point of our reduction.

Definition 2.2.1. $\mathbb{F}_q$-Quadratic CSP ($\mathbb{F}_q$-QCSP): *A* $\mathbb{F}_q$-QCSP *instance* $Q \stackrel{\text{def}}{=} \left(\{p_j\}_{j=1}^m, \{c_j\}_{j=1}^m\right)$ *consists of a set of polynomial constraints over variables* $\{z_1, z_2, \ldots, z_n\}$. *Each equation is of the form*

$$p_j(z_1, z_2, \ldots, z_n) = c_j,$$

where p_j *is a homogeneous polynomial of degree* 2*, and* $c_j \in \mathbb{F}_q$*. The goal is to find an assignment to the variables* $\{z_1, z_2, \ldots, z_n\}$ *each taking a value in* $\mathbb{F}_q$ *which satisfies as many constraints as possible. Let* $OPT(Q)$ *denote the maximum, over assignments to the variables of* Q*, of the fraction of equations satisfied.*

Definition 2.2.2. $\mathbb{F}_q$-Quadratic CSP with Pre-processing ($\mathbb{F}_q$-QCSPP): *Given a* $\mathbb{F}_q$-QCSP *instance*

$$Q \stackrel{\text{def}}{=} \left(\{p_j\}_{j=1}^m, \{c_j\}_{j=1}^m\right)$$

over variables $\{z_1, z_2, \ldots, z_n\}$ *taking values in* $\mathbb{F}_q$*, the* $\mathbb{F}_q$-QCSPP *problem allows arbitrary pre-processing on the polynomials* $\{p_j\}_{j=1}^m$ *before the inputs* $\{c_j\}_{j=1}^m$ *are revealed.*

The following theorem can be proved in a similar manner as Theorem 4.2 in [AKKV05]. We include a proof in Section 2.4.1.

Theorem 2.2.3. *For all* $q = 2^r$*, there is a reduction from a* 3-SAT *instance of size* n *to an* $\mathbb{F}_q$-QCSPP *instance of size* $\text{poly}(n, q)$ *which runs in time* $\text{poly}(n, q)$.

Next we define the problem that we prove is hard to approximate and show that it is equivalent to the Nearest Codeword Problem with Pre-processing.

Definition 2.2.4. Minimum Weight Solution Problem with and without Pre-processing: *An instance of* MWSP *consists of a set of* fixed *linear forms described by* $B_f \in \mathbb{F}_2^{l \times N}$ *, a set of* variable *linear forms* $B_v \in \mathbb{F}_2^{l' \times N}$ *and a target vector* $t \in \mathbb{F}_2^l$*. The goal is to find a solution* $x \in \mathbb{F}_2^N$ *to the system* $B_f x = t$*, which minimizes the Hamming weight of the vector* $B_v x$*. In the pre-processing version,* MWSPP*, we allow arbitrary pre-processing on all parts of the input except the vector* t*.*

Definition 2.2.5. Nearest Codeword Problem with and without Pre-processing: *An instance of* NCP *is denoted by* (C, t) *where* $C \in \mathbb{F}_2^{n \times k}$*,* $t \in \mathbb{F}_2^n$*. The goal is to find a solution* $x \in \mathbb{F}_2^k$ *which minimizes the Hamming distance* *between Cx and t. In the pre-processing version,* NCPP*, we allow arbitrary pre-processing on all parts of the input except the vector t.*

We note that MWSP is actually same as the NCP problem in disguise, though we find it convenient to think of it as a separate problem. To see the equivalence with NCP, let x_0 be a fixed vector such that $B_f x_0 = t$, let $w = B_v x_0$ and consider the code $C \stackrel{\text{def}}{=} \{B_v x \mid \ x \text{ s.t. } B_f x = 0\}$. Then

$$\min_{x:B_f x=0} \delta(w, B_v x) = \min_{x:B_f(x+x_0)=t} \delta(B_v x_0, B_v x) = \min_{x:B_f x=t} wt(B_v x).$$

Here $\delta(\cdot,\cdot)$ measures the Hamming distance and $wt(\cdot)$ denotes the Hamming weight of a string.

Finally we note that proving the hardness for NCPP implies the hardness for CVPP.

Theorem 2.2.6. *[FM04] Let $1 \leq p < \infty$. If* NCPP *(*MWSPP*) is hard to approximate to factor f then* CVPP*, under the ℓ_p norm, is hard to approximate to factor $f^{1/p}$.*

2.2.2 Boosting Soundness through Codes

The following lemma shows how to boost soundness of the $\mathbb{F}_q$-QCSPP instance although it increases the number of variables per equation. The proof of this lemma employs Reed-Muller codes and appears in Section 2.4.2.

lemma 2.2.7. *Let Q be an instance of $\mathbb{F}_q$-*QCSPP *over n variables and $k = poly(n)$ equations, for any $q = 2^r$. There is an instance P of $\mathbb{F}_q$-*QCSPP *over the same set of variables and q equations such that:*

- *If* $OPT(Q) = 1$ *then* $OPT(P) = 1$ *and*
- *if* $OPT(Q) < 1$ *then* $OPT(P) \leq k/q$.

In our reduction q would be $n^{\log^{O(1/\epsilon)} n}$ and, hence, $q \gg k$.

2.2.3 Low Degree Test

Now we move on to the tools necessary for keeping the number of queries in our PCP small. The first step in this is the Low Degree Test. The Low Degree Test is a crucial ingredient in the (original) proof of the PCP theorem. We will use a quantitatively stronger version of the test proved in [RS97, AS03].

In this section we state the basics, the test and the theorem which will be used.

An affine line in $\mathbb{F}_q^m$ is parameterized by $(a, b) \in (\mathbb{F}_q^m \setminus \{0\}) \times \mathbb{F}_q^m$ such that $L_{a,b} \stackrel{\text{def}}{=} \{ax + b : x \in \mathbb{F}_q\}$. Sometimes, we will drop the subscript if it is clear from the context. In what follows, if it helps, one can think of $m \stackrel{\text{def}}{=} \log n$ and $d \stackrel{\text{def}}{=} m$ as will be the case in our reduction. For a polynomial $g : \mathbb{F}_q^m \to \mathbb{F}_q$ of degree d and a line $L \stackrel{\text{def}}{=} L_{a,b}$, let $g|_L$ be the restriction of g defined as $g|_L(x) \stackrel{\text{def}}{=} g(ax + b)$ for $x \in \mathbb{F}_q$. For two polynomials g, h we denote $g \equiv h$ if they are identical.

Definition 2.2.8 (Low Degree Test). *The Low Degree Test takes as input the value table of a function* $f : \mathbb{F}_q^m \to \mathbb{F}_q$ *and for every (affine) line* L *of* $\mathbb{F}_q^m$*, the coefficients of a degree* d *polynomial* g_L.

The goal is to check that f *is a degree* d *polynomial. The intention is that* g_L *is the restriction of* f *to the line* L.

The test proceeds as follows:

1. *Pick a random point* $x \in \mathbb{F}_q^m$ *and a random line* L *containing* x.

2. Test that $g_L(x) = f(x)$.

The following theorem can be inferred from Theorem 1 and Lemma 14 in [AS03].

Theorem 2.2.9 (Soundness of Low Degree Test). *There are absolute constants $0 < c_1, c_2 < 1$ such that for $\delta \stackrel{\text{def}}{=} 1/q^{c_1}$, $l \stackrel{\text{def}}{=} q^{c_2}$, if $f : \mathbb{F}_q^m \to \mathbb{F}_q$ passes the Low Degree Test (Definition 2.2.8) with probability p, then there are l degree d polynomials $f^1, f^2, \ldots, f^l$ such that :*

$$\Pr_{L,x}\left[g_L(x) = f(x) \;\; \& \;\; \exists\, j \in \{1, 2, \ldots, l\} \;:\; g_L \equiv f^j|_L\right] \geq p - \delta.$$

In words, whenever the Low Degree Test accepts, except with probability δ, the test picks a line L such that g_L corresponds to the restriction of one of the polynomials $f^1, f^2, \ldots, f^l$ to L.

We assume here that $d \ll q$ (in our application, $d \leq O(\log q)$).

2.2.4 Sum Check Protocol

We will also need the Sum Check Protocol for our PCP. Like the Low Degree Test, the Sum Check Protocol [LFKN92] is an essential ingredient of the original proof of the PCP theorem. We start with some definitions, state the test and the main theorem needed for our proof. Think of $M = 2m$ and, hence, $\mathbb{F}_q^M = \mathbb{F}_q^m \times \mathbb{F}_q^m$ in the discussion below. Also one can think of $d = 4m$. We first need a notion of partial sums of polynomials.

Definition 2.2.10 (Partial Sums). *Let $g : \mathbb{F}_q^M \to \mathbb{F}_q$ be a degree d polynomial. For every $0 \leq j \leq M - 1$ and every $a_1, a_2, \ldots, a_j \in \mathbb{F}_q$ we define the partial sum $g_{a_1,a_2,\ldots,a_j}$ as a polynomial from $\mathbb{F}_q \to \mathbb{F}_q$ as follows:*

$$g_{a_1,a_2,\ldots,a_j}(z) \stackrel{\text{def}}{=} \sum_{b_{j+2},\ldots,b_M \in \{0,1\}} g(a_1, a_2, \ldots, a_j, z, b_{j+2}, \ldots, b_M).$$

When $j = 0$ *we denote the polynomial as* $g_\emptyset$*. When* $j = M - 1$*, the summation is just* $g(a_1, \ldots, a_{M-1}, z)$*. Note that all the polynomials so defined are of degree at most* d*.*

Definition 2.2.11 (Sum Check Protocol). *The Sum Check Protocol takes as input a value table for a function* $g : \mathbb{F}_q^M \to \mathbb{F}_q$*, a target sum* $c \in \mathbb{F}_q$ *and for every* $0 \leq j \leq M - 1$ *and every* $a_1, a_2, \ldots, a_j \in \mathbb{F}_q$*, the coefficients of a degree* d *polynomial* $p_{a_1,a_2,\ldots,a_j}$*. The goal is to check whether* $\sum_{z \in \{0,1\}^M} g(z) = c$*. The intention is that* g *is a degree* d *polynomial and* $p_{a_1,a_2,\ldots,a_j}$ *correspond to partial sums of* g *as in Definition 2.2.10. The test proceeds by picking* $x \stackrel{\text{def}}{=} (a_1, a_2, \ldots, a_M) \in \mathbb{F}_q^M$ *uniformly at random and accepts if and only if all of the following tests pass.*

1. $p_\emptyset(0) + p_\emptyset(1) = c$.

2. *For all* $1 \leq j \leq M - 1$, $p_{a_1,a_2,\ldots,a_{j-1}}(a_j) = p_{a_1,a_2,\ldots,a_j}(0) + p_{a_1,a_2,\ldots,a_j}(1)$.

3. $p_{a_1,a_2,\ldots,a_{M-1}}(a_M) = g(x)$.

We state the soundness of the Sum Check Protocol in a somewhat different form that is convenient for us. The proof appears in Section 2.4.3.

Theorem 2.2.12 (Soundness of Sum Check Protocol). *[LFKN92]*
Let $g^1, g^2, \ldots, g^l : \mathbb{F}_q^M \to \mathbb{F}_q$ *be degree* d *polynomials and* $g : \mathbb{F}_q^M \to \mathbb{F}_q$ *an arbitrary function. Suppose for every* $1 \leq j \leq l$, $\sum_{z \in \{0,1\}^M} g^j(z) \neq c$*. For* $x \in \mathbb{F}_q^M$*, let* $\mathcal{P}(x)$ *be the event that the Sum Check Protocol (Definition 2.2.11) accepts on inputs* g*,* c

and polynomials $p_{a_1,a_2,\dots,a_i}$ for $0 \le i \le M-1$. Here x is the choice of randomness in the Sum Check Protocol.

Then

$$\Pr_{x \in \mathbb{F}_q^M}\left[\mathcal{P}(x) \;\; \& \;\; \exists\, j \in \{1,\dots,l\} \; : \; g(x) = g^j(x)\right] \le Mdl/q$$

In words, the probability that the Sum Check Protocol accepts when g is consistent with one of $g^1, g^2, \dots, g^l$ is at most Mdl/q where $g^1, g^2, \dots, g^l$ are degree d polynomials whose sum is not the required value.

2.3 The Reduction

The following is the main theorem about the reduction and implies Theorem 1.4.1 via Theorem 2.2.6.

Theorem 2.3.1. *Unless $NP \subseteq DTIME(2^{\log^{O(1/\epsilon)} n})$,* MWSPP *is hard to approximate to factor $2^{\log^{1-\epsilon} n}$ for an arbitrary small constant $\epsilon > 0$.*

Towards the proof of this theorem, we will give a reduction from $\mathbb{F}_q$-QCSPP to MWSPP. The reduction proceeds in three steps:

- Reduction from $\mathbb{F}_q$-QCSPP to a PCP with low query complexity (Section 2.3.1).
- Viewing the PCP as an HLCPP instance (Section 2.3.2).
- Reduction from HLCPP to MWSPP (Section 2.3.3).

Finally, we will complete the proof in Section 2.3.4 where the choice of parameters is made.

2.3.1 Smooth PCP with Low Query Complexity

Note that the $\mathbb{F}_q$-QCSPP instance given by Lemma 2.2.7 has almost all the variables appearing in every equation. For the reduction to MWSPP we require a PCP where every test depends on a few variables. We will also crucially need a *smoothness* property from the PCP similar to the one described for LCPP in Section 2.1. To this end, we use the Low Degree Test of [AS03] and the Sum Check Protocol of [LFKN92].

2.3.1.1 Describing the PCP

Let P be the instance of $\mathbb{F}_q$-QCSPP given by Lemma 2.2.7 over variables $\{z_1, \ldots, z_n\}$. Let $m \stackrel{\text{def}}{=} \log n$. Here we assume that n is a power of 2. We think of the variables of P as being embedded into $\{0,1\}^m$ within $\mathbb{F}_q^m$. Henceforth, we will refer to the variables by their corresponding points in $\{0,1\}^m$. Thus, an assignment $A : \{0,1\}^m \to \mathbb{F}_q$ to the variables can be extended to a degree m polynomial $f : \mathbb{F}_q^m \to \mathbb{F}_q$ such that f is consistent with A on $\{0,1\}^m$.

Let the equations be $E_1, \ldots, E_q$ where each equation is of the form

$$E_i \equiv P_i(z_1, \ldots, z_n) = C_i \equiv \sum_{s,t \in [n]} c_i(s,t) z_s z_t = C_i \equiv \sum_{\alpha,\beta \in \{0,1\}^m} c_i(\alpha,\beta) z_\alpha z_\beta = C_i.$$

For an assignment A to $\{z_\alpha\}_{\alpha \in \{0,1\}^m}$, let f_A denote the degree m polynomial encoding A. Now, checking whether an equation $E_i \in P$ is satisfied by A amounts to checking

$$\sum_{\alpha,\beta \in \{0,1\}^m} c_i(\alpha,\beta) f_A(\alpha) f_A(\beta) = C_i.$$

Note that $c_i(\alpha,\beta)$ can be thought of as a degree $2m$ polynomial over $\mathbb{F}_q^{2m}$ and is a part of the pre-processing.

The PCP we will construct expects the following tables:

1. **Points Table:** The value of a function $f : \mathbb{F}_q^m \to \mathbb{F}_q$ at every point in $\mathbb{F}_q^m$. The intention is that f is a degree m polynomial which encodes a satisfying assignment to P within $\{0,1\}^m$, i.e., for a satisfying assignment A, $f(\alpha) = f_A(\alpha)$ for all $\alpha \in \{0,1\}^m$. The size of this table is q^m.

2. **Lines Table:** The coefficients of a degree m polynomial g_L for every (affine) line L of $\mathbb{F}_q^m$. The intention is that g_L is the restriction of f on L. The size of this table is at most $q^{2m} \cdot (m+1)$.

3. **Partial Sums Table:** For every equation $E_i \in P$, every $0 \le j \le 2m-1$ and every $a_1, a_2, \ldots, a_j \in \mathbb{F}_q$, the coefficients of a degree $4m$ polynomial $p_{i,a_1,a_2,\ldots,a_j}$. The intention is that $p_{i,a_1,a_2,\ldots,a_j}$ correspond to partial sums of g_i (Definition 2.2.10) where $g_i(\alpha, \beta) \stackrel{\text{def}}{=} c_i(\alpha,\beta) f(\alpha) f(\beta)$ where $\alpha \stackrel{\text{def}}{=} (a_1, \ldots, a_m)$ and $\beta \stackrel{\text{def}}{=} (a_{m+1}, \ldots, a_{2m})$. Note that g_i has degree at most $4m$ and the size of the j-th partial sum table is $q \cdot q^j \cdot (4m+1)$.

PCP Test:

Pick equation $E_i \in P$ uniformly at random. Pick $\alpha \stackrel{\text{def}}{=} (a_1, a_2, \ldots, a_m) \in \mathbb{F}_q^m$, $\beta \stackrel{\text{def}}{=} (a_{m+1}, a_{m+2}, \ldots, a_{2m}) \in \mathbb{F}_q^m$ uniformly at random. Let L be the line passing through α and β. Read the following values from the corresponding tables:

- $f(\alpha), f(\beta) \in \mathbb{F}_q$ from the Points table.
- The polynomial g_L from the Lines table.
- The polynomials $p_{i,a_1,a_2,\ldots,a_j}$ from the Partial Sums table for every $0 \le j \le 2m-1$.

Acceptance Criteria for the Test:

Accept if and only if all of the following tests pass.

1. $g_L(\alpha) = f(\alpha)$ and $g_L(\beta) = f(\beta)$.

2. $p_{i,\emptyset}(0) + p_{i,\emptyset}(1) = C_i$.

3. For all $1 \leq j \leq 2m-1$, $p_{i,a_1,a_2,\ldots,a_{j-1}}(a_j) = p_{i,a_1,a_2,\ldots,a_j}(0) + p_{i,a_1,a_2,\ldots,a_j}(1)$.

4. $p_{i,a_1,a_2,\ldots,a_{2m-1}}(a_{2m}) = c_i(\alpha,\beta)f(\alpha)f(\beta)$.

Note that we allow arbitrary pre-processing on everything except $\{C_i\}_{i=1}^m$.

2.3.1.2 Completeness and Soundness of the PCP

We prove the following theorem here:

Theorem 2.3.2 (Low Degree and Sum Check)**.** *Let P be a $\mathbb{F}_q$-QCSPP instance. Then*

1. *If $OPT(P) = 1$, then there is a PCP proof such that the Test succeeds with probability* 1.

2. *If $OPT(P) \leq k/q$ and $k < q^c$ for a small enough c, then the test succeeds above with probability at most $1/q^e$ for some constant $e > 0$.*

The proof of the theorem follows from the following two lemmas.

lemma 2.3.3 (Completeness)**.** *If there exists an assignment A to $\{z_1, \ldots, z_n\}$ such that $OPT(P) = 1$, i.e., $E_1, \ldots, E_q$ are all satisfied, then there is an assignment to all the tables such that the test accepts with probability* 1.

Proof. We let $f \stackrel{\text{def}}{=} f_A$, $g_L \stackrel{\text{def}}{=} f_A|_L$ for all L, and for all $i \in [q]$, $0 \leq j \leq 2m-1$, and $a_1, \ldots, a_j \in \mathbb{F}_q$,

$$p_{i,a_1,\ldots,a_j} \stackrel{\text{def}}{=} \sum_{b_{j+2},\ldots,b_{2m} \in \{0,1\}} h_i(a_1, \ldots, a_j, z, b_{j+2}, \ldots, b_{2m}),$$

where $h_i(x, y)$ is the polynomial of degree at most $4m$ representing $c_i(x, y)f_A(x)f_A(y)$. It is clear that the test succeeds with probability 1. □

lemma 2.3.4 (Soundness)**.** *There is an absolute constant $e > 0$ such that if $OPT(P) \leq k/q$ and $k < q^c$ for a small enough c, then the PCP described above has soundness at most $1/q^e$.*

Proof. We first observe that Step 1 of the test is equivalent to running a low degree test (Definition 2.2.8) on L and α with input tables g_L and f respectively. This is because the choice of β is independent of α and uniform in $\mathbb{F}_q^m$. Let $0 < c_1, c_2 < 1$ be the constants given by Theorem 2.2.9. Let $f^1, f^2, \ldots, f^l$ be the list of $l \stackrel{\text{def}}{=} q^{c_2}$ polynomials promised by Theorem 2.2.9.

The following events can happen on a run of the PCP:

1. The Low Degree Test between L and α fails. That is, $g_L(\alpha) \neq f(\alpha)$. In this case, the PCP does not accept.

2. $g_L(\beta) \neq f(\beta)$. In this case, the PCP does not accept.

3. The low degree test accepts ($g_L(\alpha) = f(\alpha)$) but there is no $1 \leq j \leq l$ such that $g_L \equiv f^j|_L$. By theorem 2.2.9, this happens with probability at most $\delta \stackrel{\text{def}}{=} 1/q^{c_1}$.

If none of the events listed above occur, then we have that g_L is the restriction of f^j for some $1 \le j \le l$. Also, since Step 1 accepts, we must have $f(\alpha) = f^j(\alpha)$ and $f(\beta) = f^j(\beta)$.

Let E_i be an equation not satisfied by any f^j for $1 \le j \le l$. Note that Steps 5.8, 3 and 4 are equivalent to running the Sum Check Protocol (Definition 2.2.11) on $g_i : \mathbb{F}_q^{2m} \to \mathbb{F}_q$ defined as $g_i(\alpha, \beta) \stackrel{\text{def}}{=} c_i(\alpha, \beta) f(\alpha) f(\beta)$. g_i has degree at most $4m$. Let $g_i^j(\alpha, \beta) \stackrel{\text{def}}{=} c_i(\alpha, \beta) f^j(\alpha) f^j(\beta)$. Finally, for $x \in \mathbb{F}_q^{2m}$, let $\mathcal{P}_i(x)$ be the event that the Sum Check Protocol accepts.

Applying Theorem 2.2.12,

$$\Pr_{x \in \mathbb{F}_q^{2m}} \left[\mathcal{P}_i(x) \;\;\&\;\; \exists\, j \in \{1, \ldots, l\} \;:\; g_i(x) = g_i^j(x) \right] \le (2m)dl/q$$

Thus, when none of the events in the list occur, the PCP accepts with probability at most $(2m) \cdot (4m) \cdot l/q$ conditioned on choosing E_i. Note that every f^j may satisfy at most k of the q equations.

Thus, the total probability that the PCP accepts is at most $\delta + (1 - lk/q) \cdot O(m^2 l/q)$ and it is easy to check that by our choice of parameters this is smaller than $1/q^e$ for some absolute constant $e > 0$. $\square$

2.3.2 PCP as Hyper-Graph Label Cover

It will be useful to think of the PCP as a graph labeling problem. The labeling problem we consider is similar to the well-known Label Cover problem except for the following differences:

- The graph is not bipartite but consists of several layers, with edges between consecutive layers. In addition, there are hyper-edges which consist of several edges. The goal is to find a labeling which satisfies the maximum fraction of hyper-edges, where the constraint corresponding to a hyper-edge is the logical AND of the constraints corresponding to each of its edges.

- The constraints corresponding to edges are not projection constraints as in the case of LABEL COVER, but the more general *many-to-many* constraints. For an edge $e = (u, v)$, a many-to-many constraint is described by an ordered partition of the label set of u and the label set of v such that the constraint is satisfied if and only if both u and v receive labels from matching partitions. Formally, let $e = (u, v)$ be an edge and $[R_u]$, $[R_v]$ be the label sets of vertices u and v. Then the many-to-many constraint is described by a pair of maps $\pi_e : [R_u] \to [R_e]$, $\sigma_e : [R_v] \to [R_e]$ where $[R_e]$ is a label set associated to e. A label l to u and a label l' to v is said to satisfy edge e if $\pi_e(l) = \sigma_e(l')$.

We now formally describe the HYPER-GRAPH LABEL COVER problem. While the term HYPER-GRAPH LABEL COVER can be potentially used for a more general class of problems, in this thesis we restrict our attention to a very special class of graphs useful for our reduction.

Definition 2.3.5. HYPER-GRAPH LABEL COVER PROBLEM (HLCP)

An instance $G(V, E, \mathcal{E}, [R_0], [R_1], \ldots, [R_{2m+1}], \{\pi_e, \sigma_e\}_{e \in E}, \{\mathcal{S}_v, S_v\}_{v \in \mathcal{L}_0})$ *of* HLCP *consists of:*

- *A graph* $G(V, E)$*. The vertices are partitioned into* $2m + 2$ *disjoint layers,* $V \stackrel{\text{def}}{=} \mathcal{L}_0 \cup \mathcal{L}_1 \cup \cdots \cup \mathcal{L}_{2m+1}$*. The edges in* E *are always between a vertex in* $\mathcal{L}_i$ *and a vertex in* $\mathcal{L}_{i+1}$ *for some* i*.*

- *Label sets* $[R_i]$ *for vertices in layer* $\mathcal{L}_i$*. Furthermore, for every vertex* $v \in \mathcal{L}_0$*, there is a partition* $\mathcal{S}_v$ *of* $[R_0]$ *and an* **allowable set** *of labels* $S_v \in \mathcal{S}_v$*.*

- *A many-to-many constraint for every edge. Let* $e = (u, v)$ *be an edge where* $u \in \mathcal{L}_i$*,* $v \in \mathcal{L}_{i+1}$*. The instance contains projections* $\pi_e : [R_i] \to [R_e]$*,* $\sigma_e : [R_{i+1}] \to [R_e]$*. A labeling* (l, l') *to* (u, v) *is said to satisfy* e *if* $\pi_e(l) = \sigma_e(l')$*.*

- *A set of hyper-edges* $\mathcal{E}$*. Every hyper-edge consists of one vertex from the first* $2m + 1$ *layers and two vertices from the last layer, such that there is an edge between any pair of vertices in adjacent layers. A labeling to the graph satisfies a hyper-edge if all the edges contained in it are satisfied.*

The goal is to find a labeling to the vertices which satisfies the maximum fraction of hyper-edges. Vertices in $\mathcal{L}_i$ *are required to receive a label from* $[R_i]$*. Furthermore, vertices in* $\mathcal{L}_0$ *are required to receive labels from their* **allowable set***.*

We also define a pre-processing version of HLCP similar to the LCPP problem of [AKKV05].

Definition 2.3.6. Hyper-graph Label Cover Problem with Pre-processing (HLCPP)

Given an instance $G(V, E, \mathcal{E}, [R_0], [R_1], \ldots, [R_{2m+1}], \{\pi_e, \sigma_e\}_{e \in E}, \{\mathcal{S}_v, S_v\}_{v \in \mathcal{L}_0})$ *of* HLCP*, the* HLCPP *problem allows arbitrary pre-processing on all parts of the input except the allowable sets* $\{S_v\}_{v \in \mathcal{L}_0}$*.*

We will need a notion of smoothness similar to the definition of Smooth Label Cover.

Definition 2.3.7. *(Smoothness)*

We say that an HLCPP *instance* $G(V, E, \mathcal{E}, [R_0], [R_1], \ldots, [R_{2m+1}], \{\pi_e, \sigma_e\}_{e \in E}, \{\mathcal{S}_v, S_v\}_{v \in \mathcal{L}_0})$ *is* δ*-smooth if for every* $0 \leq i \leq 2m$*,* $u \in \mathcal{L}_i$*,* $l \neq l' \in [R_i]$ *we have*

$$\Pr_{e=(u,v)\in E}[\pi_e(l) = \pi_e(l')] \le \delta$$

Here $v \in \mathcal{L}_{i+1}$ *and* (π_e, σ_e) *is the many-to-many constraint associated to* e*.*

Lastly, we will need that the hyper-edges of the graph are regular in a certain sense.

Definition 2.3.8. *(Uniformity)*

Let $G(V, E, \mathcal{E}, [R_0], [R_1], \ldots, [R_{2m+1}], \{\pi_e, \sigma_e\}_{e\in E}, \{\mathcal{S}_v, S_v\}_{v\in\mathcal{L}_0})$ *be an* HLCPP *instance. We say that the instance is uniform if the following conditions are satisfied:*

1. *For every* $0 \le i \le 2m+1$*, every vertex in layer* $\mathcal{L}_i$ *has the same number of hyper-edges passing through it.*

2. *For every* $0 \le i \le 2m$*, the following two distributions are equivalent:*

   - *Select an edge between a vertex in layer* $\mathcal{L}_i$ *and a vertex in layer* $\mathcal{L}_{i+1}$ *uniformly at random.*

   - *Select a hyper-edge* $\mathcal{H} \in \mathcal{E}$ *uniformly at random and then select an edge from* $\mathcal{H}$ *between a vertex in layer* $\mathcal{L}_i$ *and a vertex in layer* $\mathcal{L}_{i+1}$ *uniformly at random. Recall that a hyper-edge contains exactly one edge between layers* $\mathcal{L}_i$ *and* $\mathcal{L}_{i+1}$ *for* $0 \le i \le 2m-1$ *and two edges between layers* $\mathcal{L}_{2m}$ *and* $\mathcal{L}_{2m+1}$*.*

We next briefly describe how the PCP described in Section 2.3.1 can be thought of as an HLCPP instance.

- **Layers $\mathcal{L}_{2m}$ and $\mathcal{L}_{2m+1}$:** These are the Lines table and the Points table respectively. There is a vertex L in $\mathcal{L}_{2m}$ corresponding to every line in $\mathbb{F}_q^m$. There is a label to L for every possible univariate degree m polynomial over $\mathbb{F}_q$. Hence, the number of vertices in $\mathcal{L}_{2m}$ is at most q^{2m} and the size of the label set for each vertex is $R_{2m} = q^{m+1}$. There is a vertex α in $\mathcal{L}_{2m+1}$ corresponding to every $\alpha \in \mathbb{F}_q^m$. There is a label a to α for every possible $a \in \mathbb{F}_q$. Hence, the size of the vertex set in $\mathcal{L}_{2m+1}$ is q^m and size of the label set is $R_{2m+1} = q$.

  There is an edge between L and α if the point α belongs to the line L. The constraint between the two vertices corresponds to Step 1 of the PCP.

- **Layers $\mathcal{L}_0$ through $\mathcal{L}_{2m}$:** These are the Partial Sums table and the Lines table respectively. For $1 \leq j \leq 2m-1$, there is a vertex corresponding to $(i, a_1, a_2, \ldots, a_j)$ in $\mathcal{L}_j$ for every equation $E_i \in P$ and every $a_1, a_2, \ldots, a_j \in \mathbb{F}_q$. There is a label to $(i, a_1, a_2, \ldots, a_j)$ for every possible univariate degree $4m$ polynomial over $\mathbb{F}_q$.

  The layer corresponding to $j = 0$ is a special case since that is the only layer with a partition on the label set for each vertex. For $j = 0$, there is a vertex $(i, \emptyset)$ corresponding to every equation $E_i \in P$. There is a label to $(i, \emptyset)$ for every univariate degree $4m$ polynomial over $\mathbb{F}_q$. Furthermore, there is a partition of the label set into q parts, indexed by $\mathbb{F}_q$ as follows:

$$P_a \stackrel{\text{def}}{=} \{\text{all polynomials } p \text{ of degree at most } 4m$$

$$\text{over } \mathbb{F}_q \text{ such that } p(0) + p(1) = a\}.$$

The **allowable set** of labels for every vertex corresponds to the part that satisfies Step 5.8 of the PCP.

Thus, for $0 \leq j \leq 2m-1$, the size of $\mathcal{L}_j$ is $q \cdot q^j$ while the size the label set is $R_0 = R_1 = \cdots = R_{2m-1} = q^{4m+1}$.

For $1 \leq j \leq 2m-1$, there is an edge between a vertex $(i, a_1, a_2, \ldots, a_{j-1})$ in $\mathcal{L}_{j-1}$ and a vertex $(i', a'_1, a'_2, \ldots, a'_j)$ in $\mathcal{L}_j$ if $i = i'$ and $a_k = a'_k$ for $1 \leq k \leq j-1$. The corresponding constraints are given by Step 3 of the PCP.

There is an edge between vertex $(i, a_1, a_2, \ldots, a_{2m-1})$ in $\mathcal{L}_{2m-1}$ and vertex L in $\mathcal{L}_{2m}$ if there is an $a_m \in \mathbb{F}_q$ such that for $\alpha \stackrel{\text{def}}{=} (a_1, \ldots, a_m)$, $\beta \stackrel{\text{def}}{=} (a_{m+1}, \ldots, a_{2m})$, the line L passes through α and β. The corresponding constraints are given by Step 4 of the PCP. Note that Step 4 requires the values of the function at points α and β both of which lie on line L. Thus, a label to L specifies the values of f at α and β.

It can be checked that the constraints so defined are many-to-many constraints.[1] Note that we allow pre-processing on everything except the **allowable set** of labels for vertices in layer $\mathcal{L}_0$ as required.

It can be seen that the HLCPP instance so constructed is $4m/q$-smooth, since no two distinct degree $4m$ polynomials over $\mathbb{F}_q$ can agree on more than $4m/q$ fraction of points in $\mathbb{F}_q$.

We record this identification of the PCP with an HLCPP instance as the following theorem.

[1] Actually, the constraint between vertices in layers $\mathcal{L}_{2m-1}$ and $\mathcal{L}_{2m}$ is not many-to-many when $c_i(\alpha, \beta) = 0$ but this happens for at most $2m/q$ fraction of vertices for every equation hence we can afford to ignore these vertices and any hyper-edges containing them.

Theorem 2.3.9. *There is a reduction from an* $\mathbb{F}_q$-QCSPP *instance* P *over* n *variables to an* HLCPP *instance*
$L = G(V, E, \mathcal{E}, [R_0], [R_1], \ldots, [R_{2m+1}], \{\pi_e, \sigma_e\}_{e \in E}, \{\mathcal{S}_v, S_v\}_{v \in \mathcal{L}_0})$ *where* $m = \log n$, *such that*

1. *If* $OPT(P) = 1$, *then* $OPT(L) = 1$.

2. *If* $OPT(P) \leq k/q$ *and* $k < q^c$ *for a small enough* c, *then* $OPT(L) \leq 1/q^e$ *for some constant* $e > 0$.

Furthermore, the HLCPP *instance* L *is* $(4m/q)$*-smooth (Definition 2.3.7) and uniform (Definition 2.3.8).*

2.3.3 Reduction to MWSPP

The reduction from HLCPP to MWSPP is very similar to the reduction from LCPP to MWSPP described in [AKKV05].

Let $G(V, E, \mathcal{E}, [R_0], [R_1], \ldots, [R_{2m+1}], \{\pi_e, \sigma_e\}_{e \in E}, \{\mathcal{S}_v, S_v\}_{v \in \mathcal{L}_0})$ be an instance of HLCPP. For each vertex $v \in V$ and each label l to v we have a variable $w_{v,l}$. We now describe the fixed linear forms B_f of the MWSPP instance. Below, $\bigoplus$ denotes addition over $\mathbb{F}_2$.

Vertex constraints:

1. $\forall 1 \leq j \leq 2m+1$, $\forall v \in \mathcal{L}_j$, $\bigoplus_{l \in [R_j]} w_{v,l} = 1$.

2. $\forall v \in \mathcal{L}_0$, $\forall S \in \mathcal{S}_v$,

$$\bigoplus_{l \in S} w_{v,l} = 1 \text{ if } S = S_v \text{ and } 0 \text{ otherwise.} \tag{2.1}$$

Notice that only the r.h.s. depends on the input (which is S_v).

Edge constraints:

Let $e = (u, v)$ be an edge where $u \in \mathcal{L}_i$, $v \in \mathcal{L}_{i+1}$. Let $\pi_e : [R_i] \to [R_e]$, $\sigma_e : [R_{i+1}] \to [R_e]$ be the projections describing the many-to-many constraint associated to e. For every element $a \in [R_e]$ we add the following fixed linear form:

$$\bigoplus_{l \in [R_i]:\pi_e(l)=a} w_{u,l} = \bigoplus_{l \in [R_{i+1}]:\sigma_e(l)=a} w_{v,l}. \tag{2.2}$$

We now describe the variable forms B_v for the MWSPP instance. Let q_j be the number of vertices in layer $\mathcal{L}_j$. Let $\tilde{q} \stackrel{\text{def}}{=} \prod_{j=0}^{2m+1} q_j$. For every layer $\mathcal{L}_j$, $0 \le j \le 2m+1$, every vertex $v \in \mathcal{L}_j$ and every label l to v, we have the variable form $w_{v,l}$ repeated $\tilde{q}/q_j$ times. This completes the description of the MWSPP instance. It remains to prove the completeness and the soundness of this reduction which we do next.

2.3.3.1 Soundness of the MWSPP instance

Here we show that the MWSPP instance constructed has a large gap.

Theorem 2.3.10 (Reduction from $\mathbb{F}_q$-QCSPP to MWSPP)**.** *Let h be such that $1/(m^3 h)^{3m} \ge 1/q^e$ for large enough m and for some fixed small constant e.*

- **Completeness:** *If P is satisfiable then the* MWSPP *instance constructed in Section 2.3.3 has a solution of weight at most $(2m+2) \cdot \tilde{q}$.*

- **Soundness:** *If P is such that $OPT(P) \le k/q$ then the* MWSPP *instance constructed in Section 2.3.3 has no solution of weight less than $h \cdot (2m+2) \cdot \tilde{q}$.*

Proof. **Completeness.** If the $\mathbb{F}_q$-QCSPP instance P is satisfiable then the HLCPP instance has a labeling which satisfies all constraints (Theorem 2.3.9).

For an MWSPP variable $w_{v,l}$ corresponding to vertex v and label l to v, we let $w_{v,l} = 1$ if v was assigned the label l and 0 otherwise. It is easy to see that this satisfies all fixed linear forms and gives a solution of weight

$$\sum_{j=0}^{2m+1} \sum_{v \in \mathcal{L}_j} 1 \cdot \tilde{q}/q_j = \sum_{j=0}^{2m+1} q_j \cdot \tilde{q}/q_j = (2m+2) \cdot \tilde{q}.$$

Soundness. In this case we are given that $OPT(P) \leq k/q$ and, hence by Theorem 2.3.9, any labeling to the HLCPP instance satisfies at most $1/q^e$ fraction of the hyper-edges for some small constant e. The number of hyper-edges in the instance is $|[q] \times \mathbb{F}_q^m \times \mathbb{F}_q^m| = q^{2m+1}$. Suppose there is a solution to the MWSPP instance of weight $h \cdot (2m+2) \cdot \tilde{q}$ which satisfies all fixed linear forms. We will give a (randomized) labeling to the HLCPP instance which in expectation satisfies more than $1/(m^3 h)^{3m} \geq 1/q^e$ fraction of the hyper-edges, contradicting Theorem 2.3.9.

Let $\{w_{v,l}\}$ be a solution of weight at most $h \cdot (2m+2) \cdot \tilde{q}$. Call a label l for v *nonzero* if $w_{v,l} = 1$. (Note that our variables are allowed only 0/1 values.) We know from our assumption that

$$\sum_{j=0}^{2m+1} \sum_{v \in \mathcal{L}_j, l} w_{v,l} \cdot \tilde{q}/q_j = h \cdot (2m+2) \cdot \tilde{q}.$$

Let n_v denote the number of nonzero variables for the vertex v. Then the above can be written as

$$\sum_{j=0}^{2m+1} \sum_{v \in \mathcal{L}_j} n_v/q_j = h \cdot (2m+2).$$

Hence, for all j, $\sum_{v \in \mathcal{L}_j} n_v/q_j \leq h \cdot (2m+2)$. Hence by Markov's inequality, for every j, the fraction of vertices v for which $n_v^j \geq m^3 h$ is at most $h \cdot (2m+2)/(m^3 \cdot h) \leq 3/m^2$ for large enough m. We remove all vertices from the graph which have more than $r \stackrel{\text{def}}{=} h \cdot m^3$ non-zero labels. This removes at most $3/m^2$ fraction of vertices from each layer. Next, we remove all hyper-edges containing any vertex removed in this step. To bound this number notice that our graph has the property that number of hyper-edges per vertex of layer j is at most q^{2m+1}/q_j (by Item 1 of the uniformity property: Definition 2.3.8). Since number of vertices removed per layer is at most $3q_j/m^2$, the number of hyper-edges removed in layer j is at most $3q^{2m+1}/m^2$. Hence, the number of hyper-edges removed overall is at most $3 \cdot (2m+2)q^{2m+1}/m^2 \leq 9/m \cdot q^{2m+1}$ for large enough m. Thus, the total fraction of hyper-edges removed is at most $9/m$ which is negligible. Thus, we have an HLCPP instance where every vertex has at most r non-zero labels and we wish to satisfy more than $1/q^e$ fraction of the queries.

Labeling. We define a randomized labeling for the HLCPP instance: randomly assign a non-zero label independently for each vertex. This is possible as the sum (over $\mathbb{F}_2$) of the variables corresponding to each v is 1 and hence not all variables for a vertex can be 0.

The next claim shows that the expected fraction of hyper-edges satisfied is at least $r^{-3m} = (h \cdot m^3)^{-3m}$ which is larger than $1/q^e$ by our assumption.

Claim 2.3.11. *Conditioned on the hyper-edge not being removed, the expected fraction of hyper-edges satisfied by the randomized labeling defined above is at least* r^{-3m} *where* $r = hm^3$.

Proof of Claim. We first remove all edges e in the graph for which some pair of non-zero labels map to the same label via the constraint associated to e. Formally, let $e = (u, v)$ be an edge, $l \neq l'$ be two non-zero labels for u and (π_e, σ_e) be the maps describing the many-to-many constraint associated to e. We remove the edge

e if $\pi_e(l) = \pi_e(l')$. Since the instance is $4m/q$-smooth (Definition 2.3.7), taking a union bound over all pairs of non-zero labels implies that the fraction of edges removed in the graph is at most $4mr^2/q$.

Next, we remove all hyper-edges containing any edge removed in the previous step. Using Item 2 of the uniformity property (Definition 2.3.8) and a union bound, it can be seen that the total fraction of hyper-edges removed is at most $3m \cdot 4mr^2/q \leq 12m^2r^2/q$, which is negligible by our choice of parameters. Thus, we have an HLCPP instance where every vertex has at most r non-zero labels and the many-to-many constraint maps all non-zero labels to distinct labels.

For a hyper-edge to be satisfied, its vertex in $\mathcal{L}_0$ should receive a label from the allowable set. By Equation (2.1), there is at least one non-zero label from this set. Thus, with probability at least $1/r$, we pick an allowed label for a hyper-edge.

For $0 \leq j \leq 2m$, we will show that if we have assigned label l to vertex $u \in \mathcal{L}_j$, then the probability of assigning a consistent label to any of its neighbors in $\mathcal{L}_{j+1}$ is at least $1/r$. By a consistent label we mean one which satisfies the constraint on the edge.

Suppose we have picked a label l for a vertex $u \in \mathcal{L}_j$. We claim that the left side of Equation 2.2 is 1, since there is no non-zero label l' for u such that $\pi_e(l) = \pi_e(l')$. This means that the r.h.s. is also 1 (since the fixed linear forms are satisfied). Hence there must be a non-zero label for v which satisfies the constraint associated with the edge $e = (u, v)$, and this label is assigned to v with probability at least $1/r$ (over the random choice of a labeling). Hence, the constraint between u and v is satisfied with probability at least $1/r$.

This shows that for a fixed hyper-edge, the probability (over the randomized labeling) it is satisfied is at least $r^{-(2m+3)}$ which is the number of vertices in the hyper-edge. Thus, the expected fraction of hyper-edges satisfied is at least $r^{-(2m+3)} \geq r^{-3m}$. This completes the proof of the claim. $\square$

Noticing that by our choice of parameters $1/q^e < r^{-3m}$, we obtain a contradiction. Hence, this completes the soundness proof and, hence, the theorem. $\square$

2.3.4 Choice of Parameters and the Proof of Main Theorem

Proof of Theorem 2.3.1. Let Q be the $\mathbb{F}_q$-QCSPP instance given by Theorem 2.2.3 over n variables and $k = \mathrm{poly}(n)$ equations. We apply Lemma 2.2.7 to get an $\mathbb{F}_q$-QCSPP instance over n variables and q equations where $q \stackrel{\mathrm{def}}{=} 2^{(\log n)^{(4/\epsilon)}}$. We then apply the series of reductions described in Sections 2.3.1, 2.3.2 and 2.3.3.

Let N be the size of the MWSPP instance constructed in Section 2.3.3. It can be checked that $N \leq q^{100m}$ for large enough m, where $m \stackrel{\mathrm{def}}{=} \log n$. Hence, $N \leq q^{\log^2 n}$ for large enough n. We need m and h to satisfy

$1/r^{3m} = 1/r^{3\log n} = 1/(m^3h)^{3m} \geq 1/q^e$. This is true if $\log h \leq \log q / \log^2 n$ and $\log q \gg \log n \log\log n$ and n is let to be large enough.

We set $h \stackrel{\mathrm{def}}{=} q^{\log^{-2} n}$. For a large enough positive integer $D = 4/\epsilon$, let q be such that $\log q \stackrel{\mathrm{def}}{=} \log^D n$. Hence, $\log q \gg \log n \log\log n$. Moreover $\log N \leq \log^{D+2} n$ and $\log h = \log^{D-2} n$. This implies that

$$\log^{1-\epsilon} N = \log^{(D+2)(1-\epsilon)} n \leq \log^{D-2} n = \log h$$

Finally, $N \leq q^{\log^2 n} = 2^{\log^{O(1/\epsilon)} n}$. Summarizing, our reduction is deterministic, the hardness factor is $2^{\log^{1-\epsilon} N}$ and takes time $2^{\log^{O(1/\epsilon)} n}$ and, hence, holds under the hypothesis $NP \not\subseteq DTIME(n^{\log^{O(1/\epsilon)} n})$. $\square$

2.4 Omitted Proofs

2.4.1 $\mathbb{F}_q$-QCSPP is NP-complete

Theorem 2.4.1. *For all $q = 2^r$, there is a reduction from a* 3-SAT *instance of size n to an $\mathbb{F}_q$-*QCSPP *instance of size* $\text{poly}(n, q)$ *which runs in time* $\text{poly}(n, q)$.

Proof. We reduce 3SAT to $\mathbb{F}_q$-QCSPP. For this proof, it is convenient to view the input for 3SAT in the following form: the input is (V, E), where $V, E \in \{0, 1\}^{m \times n}$ and corresponds to a 3SAT formula $\phi = C_1 \wedge \cdots \wedge C_m$ with variables $\{x_1, \ldots, x_n\}$. Each row of V corresponds to a clause C_i and V_{ij} is 1 if and only if x_j appears in C_i. Thus, each row of V has exactly three 1's. The entry E_{ij} is 1 if and only if the variable x_j appears as a negated literal in C_i.

Since 3SAT is in NP, for every n, there is a circuit $\mathcal{C}_n$ which takes as input (V, E) and an assignment $a \in \{0, 1\}^n$, such that, $\mathcal{C}_n(a, V, E) = 1$ if a is a satisfying assignment for ϕ, and 0 otherwise.

Now we present the reduction, which is exactly the same as in Theorem 4.2 of [AKKV05], except that we work over $\mathbb{F}_q$ rather than $\mathbb{F}_2$. Let (V, E) be the input corresponding to a 3SAT instance ϕ. We may assume that every gate in $\mathcal{C}_n$ has fan-in 2 and fan-out 1. For every bit in the input (a, V, E) to $\mathcal{C}_n$, there is a variable in $\mathbb{F}_q$: x_i is supposed to be assigned the i-th bit of a, x_{ij} is supposed to be assigned V_{ij}, while x'_{ij} is supposed to be assigned E_{ij}.

Associated to the output of the i-th internal gate[2] in $\mathcal{C}_n$ is a variable z_i. Further, let y_0 be the variable corresponding to the output gate which outputs whether an assignment a satisfies ϕ or not.

The computation of any gate can be written as a quadratic polynomial (over

[2]A gate is said to be internal if its output is not an output of the circuit.

$\mathbb{F}_2$) in its inputs (call these z, z') and output (call it z''): $z'' = zz'$ for an AND gate, $z'' = 1 + (1 + z)(1 + z')$ for an OR gate, and $z'' = 1 + z$ for a NOT gate.

Note that $\mathbb{F}_2$ is a sub-field of $\mathbb{F}_q$ since $q = 2^r$ is a power of 2. Thus, each of these equations can also be thought of as equations over $\mathbb{F}_q$ which gives the correct result when both the inputs are in $\mathbb{F}_2$.

We write such an equation for every gate in $\mathcal{C}_n$. Each equation is of degree at most 2 and has at most 3 variables. Note that every such equation depends only on the description of $\mathcal{C}_n$. Finally, we add the additional set of equations $y_0 = 1$, $x_{ij} = V_{ij}$, $x'_{ij} = E_{ij}$ and $x_i^2 = x_i$. Hence, we get a $\mathbb{F}_q$-QCSPP instance over the set of variables:

$$\{x_i : i \in [n]\} \cup \{x_{ij} : i \in [m], j \in [n]\} \cup \\ \{x'_{ij} : i \in [m], j \in [n]\} \cup \{z_i : 1 \leq i \leq \text{size}(\mathcal{C}_n)\} \cup \{y_0\}.$$

Notice that $\mathcal{C}_n$ can be generated by a polynomial time algorithm which is given as input 1^n. Hence, this reduction is a polynomial time reduction.

We claim that this quadratic system has a solution (over $\mathbb{F}_q$) if and only if ϕ has a satisfying solution. The corresponding claim when all variables take values in $\mathbb{F}_2$ follows by construction. Finally note that we have restricted all satisfying inputs to $\mathbb{F}_2$ because of the constraints $x_i^2 = x_i$.

The reduction described above gives constraints which are of degree at most 2, but not homogeneous. This is easy to fix by introducing an auxiliary variable z_0 and adding the constraint $z_0 z_0 = 1$. We then multiply all terms of degree less than 2 by z_0.

This completes the proof of the lemma.

$\square$

2.4.2 Boosting Soundness through Codes

We first need some basic definitions.

Definition 2.4.2. Codes: *A matrix $C \in \mathbb{F}_q^{m \times k}$ is said to be a generator of the linear code $\{Cx : x \in \mathbb{F}_q^k\}$ with distance $1-\delta$ if for any $x \neq y \in \mathbb{F}_q^k$, $C(x)$ and $C(y)$ agree on at most δm co-ordinates.*

fact 2.4.3 (Reed-Muller Codes). *For any q, let $\mathbb{F}_q$ be the field over q elements. There is a family of linear codes with generator matrix $C_k \in \mathbb{F}_q^{q \times k}$ with distance $1-k/q$. These are the so called Reed Muller codes over $\mathbb{F}_q$, where the message is thought of as the coefficients of a degree $k-1$ polynomial and the codeword as the evaluation of this polynomial on all the points in $\mathbb{F}_q$.*

lemma 2.4.4. *Let Q be an instance of $\mathbb{F}_q$-QCSPP over n variables and $k = poly(n)$ equations, for any $q = 2^r$. There is an instance P of $\mathbb{F}_q$-QCSPP over the same set of variables and q equations such that:*

- *If $OPT(Q) = 1$ then $OPT(P) = 1$ and*
- *if $OPT(Q) < 1$ then $OPT(P) \leq k/q$.*

2.4.3 Sum Check Protocol

Theorem 2.4.5 (Soundness of Sum Check Protocol). *[LFKN92]*
Let $g^1, g^2, \ldots, g^l : \mathbb{F}_q^M \to \mathbb{F}_q$ be degree d polynomials and $g : \mathbb{F}_q^M \to \mathbb{F}_q$ an arbitrary function. Suppose for every $1 \leq j \leq l$, $\sum_{z \in \{0,1\}^M} g^j(z) \neq c$. For $x \in \mathbb{F}_q^M$, let $\mathcal{P}(x)$ be the event that the Sum Check Protocol (Definition 2.2.11) accepts on inputs g, c

and polynomials $p_{a_1,a_2,\ldots,a_i}$ for $0 \le i \le M-1$. Here x is the choice of randomness in the Sum Check Protocol.

Then

$$\Pr_{x\in\mathbb{F}_q^M}\left[\mathcal{P}(x) \ \ \& \ \ \exists\, j \in \{1,\ldots,l\} \ : \ g(x) = g^j(x)\right] \le Mdl/q$$

In words, the probability that the Sum Check Protocol accepts when g is consistent with one of $g^1, g^2, \ldots, g^l$ is at most Mdl/q where $g^1, g^2, \ldots, g^l$ are degree d polynomials whose sum is not the required value.

Proof. We will prove the theorem by induction on M. The Steps in the discussion below refer to the Sum Check Protocol (Definition 2.2.11).

Base Case: M=1 We consider two cases:

1. $p_\emptyset = g^j_\emptyset$ for some $1 \le j \le l$. In this case Step 1 fails by our assumption on g^j.

2. $p_\emptyset \ne g^j_\emptyset$ for all $1 \le j \le l$. In this case,

$$\begin{aligned}
& \Pr_{x\in\mathbb{F}_q}[\mathcal{P}(x) \ \ \& \ \ \exists\, j \in \{1,\ldots,l\} \ : \ g(x) = g^j(x)] \\
\le \ & \Pr_{x\in\mathbb{F}_q}[g(x) = p_\emptyset(x) \ \ \& \ \ \exists\, j \in \{1,\ldots,l\} \ : \ g(x) = g^j(x)] \\
& \text{(Since Step 3 accepts)} \\
= \ & \Pr_{x\in\mathbb{F}_q}[\exists\, j \in \{1,\ldots,l\} \ : \ p_\emptyset(x) = g^j(x)] \\
\le \ & ld/q
\end{aligned}$$

The last inequality uses the fact that any two distinct degree d polynomials can agree on at most d/q fraction of the points followed by a union bound.

Inductive Case: M = N We again consider two cases as before:

1. $p_\emptyset = g^j_\emptyset$ for some $1 \le j \le l$. In this case Step 1 fails by our assumption on g^j.

2. $p_\emptyset \ne g^j_\emptyset$ for all $1 \le j \le l$. In this case, the fraction of points $a \in \mathbb{F}_q$ such that

$$\sum_{b_2,\ldots b_N \in \{0,1\}} g^j(a, b_2, \ldots, b_N) = p_\emptyset(a) \tag{2.3}$$

for some $1 \le j \le l$ is at most ld/q.

Note that for a fixed $a \in \mathbb{F}_q$, Steps 2 and 3 are equivalent to running the Sum Check Protocol for checking

$$\sum_{b_2,\ldots b_N \in \{0,1\}} g(a, b_2, \ldots, b_N) = c'$$

where $c' \stackrel{\text{def}}{=} p_\emptyset(a)$. For $x \in \mathbb{F}_q^{N-1}$, let $\mathcal{P}_a(x)$ be the event that this protocol accepts.

If Equation 2.3 does not hold for any $1 \le j \le l$ then we can use the inductive assumption to get

$$\Pr_{x \in \mathbb{F}_q^{N-1}} \left[\mathcal{P}_a(x) \;\;\&\;\; \exists\, j \in \{1, \ldots, l\} \;:\; g(a, x) = g^j(a, x)\right] \le (N-1)dl/q$$

Thus, the total probability of acceptance is at most $ld/q + (N-1)dl/q \le Ndl/q$.

$\square$

Chapter 3

Hardness of pricing Loss Leaders

In this chapter we will prove Theorem 1.4.2, Theorem 1.4.5 and Lemma 1.4.3. In the next section, we give an overview of our proof for Theorem 1.4.2 and Theorem 1.4.5. In Section 3.2 we formally define the problems and introduce some notation and tools that will be useful for the rest of the chapter. We restate and prove Lemma 1.4.3 in Section 3.2.5, Theorem 1.4.2 in Section 3.3 and Theorem 1.4.5 in Section 3.4.

3.1 Overview of our proof

In this section, we give a high level overview of our proof. We assume some familiarity with the proof of Max 2-LIN$_q$ in [KKMO07] and Max 3-LIN$_q$ in [Hås01]. Since the approximability of Highway Pricing is equivalent to that of Vertex Pricing$_2$ on bipartite graph, we would only describe our proof for Vertex Pricing$_2$ and Vertex Pricing$_3$.

The work of [KKMO07] establishes a connection between "Dictator Test" and "Hardness of approximation" of CSPs assuming the Unique Games Conjec-

TURE. The pricing problem can be viewed as a CSP with a generalized payoff function. Therefore, our hardness results for the VERTEX PRICING is based on building a proper Dictator Test. In addition, for the VERTEX PRICING$_3$ problem, since our result only assumes P$\neq$ NP, we need to combine the Dictator Test with a PCP construction that Håstad uses to obtain the NP-hardness result for MAX 3-LIN$_q$ [Hås01].

Roughly speaking, a Dictator Test for VERTEX PRICING is just a instance of the VERTEX PRICING problem defined over the vertex set $\mathbb{F}_q^n$ where n is thought of as a large number. A pricing to these items is as a function defined over $f : \mathbb{F}_q^n \to \mathbb{R}$. A Dictator function is functions that only depend on one of its coordinates. The Dictator Test is a VERTEX PRICING instance with the following properties:

- (completeness) There exists some one dimensional real function $h : \mathbb{R} \to \mathbb{R}$ such that $f(x) = h(x_i)$ has a high profit c for every $i \in [n]$.
- (soundness) Any function that depends on a lot of its coordinates will have at most profit s.

By the reduction in [KKMO07], a Dictator Test with above property would establish that it is UG-hard to distinguish whether a given instance of vertex pricing has profit above c or below s (which implies a s/c hardness of approximation result).

For example, to construct a Dictator Test for VERTEX PRICING$_2$ under the coupon model, it is enough to specify a distribution over $x, y \in \mathbb{F}_q^n, b \in \mathbb{R}^+$. Here we add a customer interested in x, y with budget b and the weight is the probability mass on (x, y, b). The profit of the price function $f : \mathbb{F}_q^n \to \mathbb{R}$ can be written as $\mathbf{E}_{x,y,b}[I(0 \leq f(x)+f(y) \leq b)\cdot(f(x)+f(y))]$, where $I(w)$ is the indicator function of the boolean condition w. The main task is to construct such a distribution which has a good profit for dictator functions and has a low profit for all functions that depend on a lot of coordinates.

3.1.1 The Dictator Test for Vertex Pricing$_2$

The Dictator Test for VERTEX PRICING$_2$ is, in a sense, similar to the Dictator Test that is used in [KKMO07] to obtain a hardness result of MAX 2-LIN$_q$. MAX 2-LIN$_q$ is a problem of solving a linear system over $\mathbb{Z}_q$ where each equation only depends on two variables. The main construction of the Dictator Test (as a instance of MAX 2-LIN$_q$ over $\mathbb{F}_q^n$) is described as follows: choose x to be uniformly random from $\mathbb{F}_q^n$ and y is generated by adding "ϵ noise" as follows: for every i, $\boldsymbol{y}_i = \boldsymbol{x}_i$ with probability $1-\epsilon$ and $\boldsymbol{y}_i$ is set to be a random element in $\mathbb{Z}_q$ with probability ϵ. Then the Dictator Test will add a equation $f(\boldsymbol{x}) - f(\boldsymbol{y}) = 0$.

Since our objective function is of the form $I(0 \le f(\boldsymbol{x})+f(\boldsymbol{y}) \le b)\cdot(f(\boldsymbol{x})+f(\boldsymbol{y}))$, we first construct $\boldsymbol{x}$ uniformly random from $[q]^n$ and $\boldsymbol{y}_i = [b - \boldsymbol{x}_i]_q$ where budget b is randomly chosen from $2, 4, 8, \ldots, 2^k$ where $k := \log\sqrt{q}$. We add $2^k/b$ edges between $\boldsymbol{x}$ and $\boldsymbol{y}$. Notice that if we use the Dictator price function $f(\boldsymbol{x}) = \boldsymbol{x}_i - q/2$ and $f(\boldsymbol{y}) = \boldsymbol{y}_i - q/2$, then with probability at least $1 - 1/\sqrt{q} \ge 1/2$, we have $f(\boldsymbol{x}) + f(\boldsymbol{y}) = b$. Therefore, the profit of the dictator function is at least $1/2 \cdot 1/k \cdot \sum_{i=1}^{k} 2^k/2^i = \Omega(2^k)$.

On the other hand, we manage to show that functions that depend on a lot of coordinates cannot have a profit significantly better than the constant price function that assign the same price to every item. It is easy to verify that for any constant price function, the profit on the above instance is at most $O(2^k/k)$. This gives us a $\Omega(\log q)$ gap between the profit of Dictator function and functions that depends on a lot of its coordinates. Note that q can be an arbitrarily large constant in this construction.

Technically, the main body of the proof is to show that functions that depends on a lot of coordinates behave like constant functions. We use the general approach of [KKMO07]. For simplicity, suppose the pricing function is integer valued $f : \mathbb{F}_q^n \to [q]$ and suppose we have a customer interested in x, y with budget q, then the profit can be written as $\sum_{0 \le i+j \le q} (i+j) f_i(x) f_j(x)$. One of the difficulty we face is that there can be $\Omega(q^2)$ terms in the sum while the analysis in [KKMO07] usually generate x, y with noise rate ϵ and this would bound the sum by $q^{2-\epsilon}$. To see why this is the case, let us recall how the analysis in [KKMO07] proceeds. The goal is to show that functions which depend on many coordinates satisfy a small fraction of equations in the MAX 2-LIN$_q$ instance. This is achieved by writing the fraction of equations satisfied by a function f in terms of its *Noise stability*, and using the invariance principle of [MOO05] to show that if f depends on many co-ordinates then the Noise stability of f is essentially the same as the Noise stability of a related function $\tilde{f}$ which takes as input gaussian random variables rather than $\mathbb{F}_q$-valued random variables. It is known by a result of [Bor85] that the function $\tilde{f}$ for which the noise stability is maximum in the gaussian domain is the half-space with the appropriate measure, and [KKMO07] use estimates about the noise stability of this function to prove their result.

We follow the same approach as [KKMO07] of writing the profit in terms of the Noise stability of the pricing function. It turns out that if the noise is small then the expression for profit is quite large when written for the half-space function in the gaussian domain. Our main technical contribution is to get around this issue by introducing a large noise ($1 - o(1)$ noise) and carefully analyzing the noise stability of pairs of half-spaces in the gaussian domain (Lemma 3.1.4).

Another technical challenge in our proof is that we need our hardness result hold even for Vertex Pricing$_2$ on bipartite graph, therefore we use a bipartite version of the invariance principle due to [DMR09].

3.1.1.1 Main Technical Lemma

Here we state the main technical lemma used in the reduction for Vertex Pricing$_2$.

Definition 3.1.1. *Let ϕ be the probability density function of the standard gaussian i.e. $\phi(t) := \frac{1}{\sqrt{2\pi}} e^{-t^2/2}$.*

Definition 3.1.2. *Let N be the cumulative distribution function of the standard gaussian i.e. $N(t) := \int_t^\infty \phi(x)dx$. Equivalently, $N(t) := \Pr[X \geq t]$ where X is a standard gaussian random variable.*

Definition 3.1.3. *(Gaussian noise stability of half-spaces) $\Lambda_\rho(\mu, \nu) := \Pr[X \geq t \text{ and } Y \geq s]$ where $t := N^{-1}(\mu)$, $s := N^{-1}(\nu)$ and X, Y are standard Gaussians with $\mathbf{E}[XY] = \rho$.*

lemma 3.1.4. *Let $1/(q \log q) \leq \mu \leq 1$, $\rho \leq (\log q)^{-(1/2+\epsilon)}$, $k \leq \log q$ and $\{\nu_1, \nu_2, \ldots, \nu_k\}$ be such that $\sum_{i=1}^k \nu_i \leq 1$. Then for q large enough,*

$$\sum_{i=1}^{k} \Lambda_\rho(\mu, \nu_i) = O(\mu)$$

3.1.2 The Dictator Test for Vertex Pricing$_3$

Our construction for the Vertex Pricing$_3$ is based on Håstad's seminal result of Max 3-LIN$_q$ [Hås01] and a Dictator Test for Vertex Pricing$_3$ that is previously introduced in [Wu11]. Håstad essentially construct "Matching-Dictator Test" on two functions for $f\colon \mathbb{Z}_q^K \to \mathbb{Z}_q$, $g\colon \mathbb{Z}_q^L \to \mathbb{Z}_q$ and $\pi\colon L \to K$. The test is defined by a distribution over $\boldsymbol{x} \in \mathbb{Z}_q^K, \boldsymbol{y} \in \mathbb{Z}_q^L, \boldsymbol{z} \in \mathbb{Z}_q^L$ with the check $f(x) + g(y) + g(z) = 0 \mod q$. Håstad's Test has the following completeness and soundness promises:

- If $f(x) = x_i$ and $g(y) = y_j$ such that $\pi(i) = j$, then f and g passes with probability $1 - \epsilon$.

- If f and g are far from being a pair of matching dictator functions, then they behave like constant functions.

Our proof essentially use the same distribution of x, y, z and add $\lfloor \sqrt{q} \rfloor$ buyers such that they are interested in $x, y, z \oplus_q \lfloor q/t \rfloor \cdot (1, 1, ...1)$ with budget q/t for every $t \in \lfloor \sqrt{q} \rfloor$. It is easy to verify that for every $i \in [q]$ and $f(x) = x_i - q/3, g(y) = y_i - q/3$ would have $\log q$ times more profit than setting f, g to be a constant. The main body of the work is to show if f, g are far from being "matching" dictator functions" , then they just behave like being constant functions.

Notice that in [Wu11], the author manages to construct such a test for $K = L$ and $f = g$, which suffices to give a hardness result assuming the Unique Games Conjecture. Technically speaking, in [Wu11], the author used the *invariance principle* [MOO05] to analyze the profit of functions that depends on a lot of coordinates. However we can not use the invariance here directly partly because it requires pairwise independent distributions. Also the projection instead of bijection in our test make it hard for us to use the same analysis. We also found it hard to directly use the Fourier Analysis with complex function basis by which Håstad proved the hardness result for Max 3-LIN$_q$. This is because our objective function is less symmetric compared with the objective function of Max 3-LIN$_q$.

Instead we use the Efron-Stein Decomposition combining with a Håstad style decoding. Such a proof also avoid the use of the "invariance principle" which we view as a simplification of the proof in [Wu11]. Our proof is inspired by a recent work [OWZ11] which also uses the same method to generalize Håstad's MAX 3-LIN$_q$ result to the integer domain without using the complex Fourier analysis.

3.1.3 Open problems

We show that GRAPH VERTEX PRICING and HIGHWAY PRICING are UG-hard to approximate to any constant factor under the Coupon model. It would be interesting to prove a similar result for the discount model. Our techniques fall short of achieving this because of the necessity to introduce very large noise as explained in Section 3.1.1.

3.2 Preliminaries

In this section, we formally define all the pricing problems and some notation used frequently throughout the chapter.

3.2.1 Notations

For q being a positive integer, we define:

- $\mathbb{Z}_q$: the set $\{0, 1 \ldots, q-1\}$.
- $[x]_q$: the remainder of x divided by q
- $\oplus_q$: addition of integers (or integer vectors) modulo q.

- For any statement ω, $I(\omega) \to \{0,1\}$ is the indicator function of whether w is correct (when $I(w) = 1$) or not (when $I(w) = 0$).
- $\lfloor x \rfloor$: for any $x \in \mathbb{R}$, $\lfloor x \rfloor$ is the largest integer less than or equal to x.
- $\lceil x \rceil$: for any $x \in \mathbb{R}$, $\lceil x \rceil$ is the smallest integer greater than or equal to x.

3.2.2 Problem Definitions

Definition 3.2.1 (Vertex Pricing)**.** *An instance*

$$\mathcal{I}(G(V,E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$$

of Vertex Pricing *is characterized by a multi-hypergraph $G(V,E)$. Here each vertex corresponds to an item and each hyper-edge corresponds to the bundle of items a customer is interested in. For each edge $e \in E$, there is an associated budget $b_e > 0$ and a weight w_e. When G is a k-hypergraph, we call the corresponding problem* Vertex Pricing$_k$*.*

The goal is to find a pricing function $f : V \to \mathbb{R}$ so as to maximize the profit. As we have discussed, there are mainly two kinds of profit models considered previously. The first profit model is the discount model. Given a vertex pricing instance $\mathcal{I}(G(V,E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$ and a price function $f : V \to \mathbb{R}$:

Definition 3.2.2 (profit under the discount model)**.**

$$\mathbf{profit}_{\mathcal{I}}(f) = \sum_e I\left(f(e) \le b_e\right) \cdot w_e \cdot f(e)$$

where $f(e) := \sum_{v \in e} f(v)$.

Under the above model, the seller may lose money to the buyer when $\sum_{v \in e} f(v) < 0$. The coupon model assumes the seller would have at least profit 0 from each buyer.

Definition 3.2.3 (profit under the coupon model)**.**

$$\mathbf{profit}^{+}_{\mathcal{I}}(f) = \sum_{e} I(0 \leq f(e) \leq b_e) \cdot w_e \cdot f(e)$$

where $f(e) := \sum_{v \in e} f(v)$.

Now given a vertex pricing instance $\mathcal{I}(G(V,E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$, we can study the problem of maximizing the profit under the following three settings. The first one is the widely studied one when the seller want to price each item with a *positive* profit margin:

Definition 3.2.4. *(Positive price model)* $\mathrm{Opt}(\mathcal{I}) = \max_{f: V \to \mathbb{R}^+} \mathbf{profit}_{\mathcal{I}}(f)$.

When we allow a real-valued price function, we can maximize the profit under either the coupon or discount model.

Definition 3.2.5. *(Discount Model)* $\mathrm{Opt}^{D}(\mathcal{I}) = \max_{f: V \to \mathbb{R}} \mathbf{profit}_{\mathcal{I}}(f)$.

Definition 3.2.6. *(Coupon Model)* $\mathrm{Opt}^{C}(\mathcal{I}) = \max_{f: V \to \mathbb{R}} \mathbf{profit}^{+}_{\mathcal{I}}(f)$.

We also consider the following Highway Pricing problem.

Definition 3.2.7 (Highway Pricing)**.** *Let* $V = \{0, 1, 2, \ldots, n\}$*.* G *is an* n*-edge line with* $e_i = (i-1, i)$ *for* $i = 1, 2, \ldots, n$*. We are given a set of intervals* $I_1, I_2, \ldots I_m$ *where each interval is specified by* $I_j = [s_j, t_j]$ *for* $s_j, t_j \in V$ *with an associated budget* b_j *and weight* w_j*. The goal is to properly price each* e_i *with a price function* $p : [n] \to \mathbb{R}$ *so as to maximize the total revenue.*

The item of the HIGHWAY PRICING problem is the segments of a line graph. Alternatively, we can think of the problem as finding a function that assign a price on $f : V \to \mathbb{R}$ and the toll on $\{i, i+1\}$ is defined as $p(i) = f(i+1) - f(i)$.

3.2.3 Gaussians

lemma 3.2.8.

$$N^{-1}(\mu) = \Theta(\sqrt{\log(1/\mu)})$$

Proof. Let $t = N^{-1}(\mu)$. Use the well known fact that $N(t) \sim \phi(t)/t$ along with the definition of ϕ and $N(t) = \mu$. □

3.2.4 Tools from Discrete Fourier Analysis

We recall some standard definitions from the discrete Fourier analysis (see, e.g., [Rag09]). We will be considering functions of the form $f : \mathbb{Z}_q^n \to \mathbb{R}$. The set of all functions $f : \mathbb{Z}_q^n \to \mathbb{R}$ forms an inner product space with inner product

$$\langle f, g \rangle = \mathop{\mathbf{E}}_{\boldsymbol{x} \sim \mathbb{Z}_q^n} [f(\boldsymbol{x}) \cdot g(\boldsymbol{x})]$$

where $\boldsymbol{x} \sim \mathbb{Z}_q^n$ means that $\boldsymbol{x}$ is chosen uniformly at random from $\mathbb{Z}_q^n$. We also write $\|f\|_2 = \sqrt{\langle f, f \rangle}$ as usual.

The following Efron–Stein decomposition theorem is well-known;
see e.g. [KKMO07].

Theorem 3.2.9. *Any $f : \mathbb{Z}_q^n \to \mathbb{R}$ can be uniquely decomposed into sum of func-*

tions

$$f(x) = \sum_{S \subseteq [n]} f^S(x),$$

where

- $f^S(x)$ *depends only on* $x_S = (x_i, i \in S)$,
- *for every* $S \subseteq [n]$, *for every* S' *such that* $S \setminus S' \neq \emptyset$, *and for every* $y \in \mathbb{Z}_q^n$, *it holds that*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}}[f^S(\boldsymbol{x}) | \boldsymbol{x}_{S'} = y_{S'}] = 0.$$

We also need define the noise operator as follows:

Definition 3.2.10. *For* $x \in \mathbb{Z}_q^n$, *we define random variable* $\boldsymbol{y} \sim_\rho x$ *if* $\boldsymbol{y}$ *is generated as follows: for each coordinate* $i \in [n]$, *independently we set* $\boldsymbol{y}_i = x_i$ *with probability* ρ *and uniformly random in* $[q]$ *with probability* $1 - \rho$. *For functions* $f : \mathbb{Z}_q^n \to \mathbb{R}$, *define the noise operator* T_ρ *to be*

$$T_\rho f(x) = \mathop{\mathbf{E}}_{\boldsymbol{y} \sim_\rho x}[f(\boldsymbol{y})].$$

Definition 3.2.11 (influence)**.** *For function* $f : \{-1, 1\}^n \to \mathbb{R}$, *we define the influence of the* i*-th coordinate* $\mathbf{Inf}_i f$ *as*

$$\mathbf{Inf}_i f = \sum_{S \ni i} \|f^S\|_2^2$$

Definition 3.2.12 (low-degree influence)**.** *For function* $f : \{-1, 1\}^n \to \mathbb{R}$, *we define the* k*-degree influence of the* i*-th coordinate* $\mathbf{Inf}_i^k f$ *as*

$$\mathbf{Inf}_i^k f = \sum_{S \ni i,\ |S| \le k} \|f^S\|_2^2$$

Following facts are well known.

fact 3.2.13. $\|T_\rho f^S\|_2^2 = \rho^{|S|}\|f^S\|_2^2$.

fact 3.2.14. *For* $f : \mathbb{F}_q^n \mapsto \mathbb{F}_q$

$$\left[\sum_{i=1}^{n} \mathbf{Inf}_i(T_{1-\eta} f) \le 1/\eta.\right]$$

fact 3.2.15. *For* $f : \mathbb{F}_q^n \mapsto \mathbb{F}_q$

$$\left[\sum_{i=1}^{n} \mathbf{Inf}_i^k(f) \le k\right]$$

Definition 3.2.16. *For a function* $f : \mathbb{F}_q^n \mapsto \mathbb{F}_q$ *and* $a \in \mathbb{F}_q$*, let* $f^a : \mathbb{F}_q^n \mapsto \mathbb{R}$ *be defined as* $f^a(x) := 1$ *if* $f(x) = a$ *and* $f^a(x) := 0$ *otherwise.*

Definition 3.2.17. *The noise stability of* f *and* g *at* ρ *is defined to be* $\mathbb{S}_\rho(f, g) := \langle f, T_\rho g\rangle$.

Theorem 3.2.18. *[DMR09]*

Fix $q \ge 2$ *and* $0 < \rho < 1$*. Then for any* $\delta > 0$ *there is a* $\tau = \tau(\rho, \delta, q) > 0$ *small enough and* $k = k(\rho, \delta, q)$ *large enough such that if* $f, g : [q]^n \mapsto [0, 1]$ *are any functions satisfying* $E[f] = \mu$*,* $E[g] = \nu$ *and* $\min(\mathbf{Inf}_i^k(f), \mathbf{Inf}_i^k(g)) \le \tau$ *for all* $i = 1 \ldots n$*, then*

$$\mathbb{S}_\rho(f, g) \le \Lambda_\rho(\mu, \nu) + \delta$$

3.2.5 Proof of Lemma 1.4.3

In this subsection, we restate and prove Lemma 1.4.3 which states that the approximability of Highway Pricing is equivalent to that of Vertex Pricing$_2$ on a bipartite graph.

lemma. *(Restatement of Lemma 1.4.3) Consider the profit maximization problem under the coupon model. If* Vertex Pricing$_2$ *on bipartite graph is hard to approximate to factor α , then* Highway Pricing *problem is also hard to approximate to factor α under the coupon model.*

Proof. Let $\mathcal{I}(G(V,E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$, $V := V_L \cup V_R$ be an instance of Vertex Pricing$_2$ and assuming that G is a bipartite graph. We construct an instance of Highway Pricing $\mathcal{J}(G(V,E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$, $V := V_L \cup V_R$ as follows.

The vertex set remains the same. We align all the vertices in V_L to the left of the vertices in V_R in a line. Then for every edges $e = (v_1, v_2)$ in $\mathcal{I}$, we also add a driver interested in the interval between v_1 and v_2.

We claim that $\mathrm{Opt}^+(\mathcal{I}) = \mathrm{Opt}^+(\mathcal{J})$. This is because if $f_L : V_L \mapsto \mathbb{R}$, $f_R : V_R \mapsto \mathbb{R}$ is a pair of pricing functions for $\mathcal{I}$ then $(-f_L, f_R)$ is a pair of pricing functions for $\mathcal{J}$ such that $\mathbf{profit}^+_{\mathcal{I}}(f_L, f_R) = \mathbf{profit}^+_{\mathcal{J}}(-f_L, f_R)$, and the argument is reversible. □

3.3 UG-hardness of Graph Vertex Pricing

In this Section we prove Theorem 1.4.2, restated below.

Theorem. *(Theorem 1.4.2 restated)* Graph Vertex Pricing *under the coupon* *model is UG-hard to approximate to any constant factor, even when the graph is bipartite.*

In the next subsection, we state and prove the main technical lemma used in our proof. In Section 3.3.2 we describe our dictatorship test and prove its completeness and soundness. Finally in Section 3.3.3, we compose the dictatorship test with UNIQUE GAMES to obtain our final reduction. Theorem 1.4.2 follows from Theorem 3.3.6.

3.3.1 Main technical lemma

lemma 3.3.1. *Let $1/(q \log q) \leq \mu \leq 1$, $\rho \leq (\log q)^{-(1/2+\epsilon)}$, $k \leq \log q$ and $\{\nu_1, \nu_2, \ldots, \nu_k\}$ be such that $\sum_{i=1}^{k} \nu_i \leq 1$. Then for q large enough,*

$$\sum_{i=1}^{k} \Lambda_\rho(\mu, \nu_i) = O(\mu)$$

Proof. Fix $1 \leq i \leq k$. Let $t := N^{-1}(\mu)$, $s_i := N^{-1}(\nu_i)$ and (X, Y) be standard Gaussians with $\mathbf{E}[XY] = \rho$.

$$\begin{aligned}
\Lambda_\rho(\mu, \nu_i) &= \Pr[X \geq t \ , \ Y \geq s_i] \\
&= \Pr[Y \geq s_i \mid X \geq t] \cdot \Pr[X \geq t] \\
&= \mu \cdot \Pr[Y \geq s_i \mid X \geq t] \\
&\leq \mu \cdot (\Pr[Y \geq s_i \mid X \in [t, 2t]] + \Pr[X \geq 2t \mid X \geq t])
\end{aligned}$$

Using $N(x) \sim \phi(x)/x$ and $N(t) = \mu$ we get

$$\Pr[X \geq 2t \mid X \geq t] = O(\mu^2 \log(1/\mu))$$

Thus,

$$\begin{aligned}
\sum_{i=1}^{k} \Lambda_\rho(\mu, \nu_i) &\leq \sum_{i=1}^{k} \mu \cdot (\Pr[Y \geq s_i \mid X \in [t, 2t]] + \Pr[X \geq 2t \mid X \geq t]) \\
&= \mu \cdot O(\mu^2 \log^2(1/\mu)) + \sum_{i=1}^{k} \mu \cdot \Pr[Y \geq s_i \mid X \in [t, 2t]] \\
&= O(\mu) + \mu \cdot \sum_{i=1}^{k} \Pr[Y \geq s_i \mid X \in [t, 2t]]
\end{aligned}$$

Thus, it suffices to show that $\sum_{i=1}^{k} \Pr[Y \geq s_i \mid X \in [t, 2t]] = O(1)$. Let Z be a standard gaussian independent of X. For a fixed i we have,

$$\begin{aligned}
\Pr[Y \geq s_i \mid X \in [t, 2t]] &= \Pr[\rho X + \sqrt{1-\rho^2} Z \geq s_i \mid X \in [t, 2t]] \\
&\leq \Pr[Z \geq (s_i - 2\rho t)/\sqrt{1-\rho^2}] \\
&\leq \Pr[Z \geq s_i - (2\rho t/\sqrt{1-\rho^2})] \\
&\leq \Pr[Z \geq s_i - 4\rho t]
\end{aligned}$$

where we used $\rho^2 < 3/4$.

Using Fact 3.2.8 we have $t = O(\sqrt{\log(1/\mu)}) = O(\sqrt{\log q})$. Since $\rho \leq (\log q)^{-(1/2+\epsilon)}$, we have that $4\rho t = O((\log q)^{-\epsilon})$.

Now $\Pr[Z \geq a-b] \leq \Pr[Z \geq a] + b\phi(a-b)$. Also, $\phi(a-b) = 1/\sqrt{2\pi} e^{-(a-b)^2/2} = O(e^{-a^2/2} \cdot e^{-b^2/2} \cdot e^{ab}) = O(\phi(a) \cdot e^{ab})$. Thus, $\Pr[Z \geq s_i - 4\rho t] \leq \Pr[Z \geq s_i] + \rho t \cdot \phi(s_i) \cdot e^{O(\rho t s_i)} \leq \nu_i + \rho t \cdot \phi(s_i) \cdot e^{O((\log q)^{-\epsilon} s_i)}$. We consider two cases.

- If $s_i \leq (\log q)^{\epsilon/2}$, then $\rho t \cdot \phi(s_i) \cdot e^{O((\log q)^{-\epsilon} s_i)} \leq \rho t \cdot O(\phi(s_i)) = \rho t \cdot O(s_i N(s_i)) = O((\log q)^{-\epsilon} \nu_i s_i) = O(\nu_i)$.

- If $s_i \geq (\log q)^{\epsilon/2}$, then $\nu_i + \rho t \cdot \phi(s_i) \cdot e^{O((\log q)^{-\epsilon} s_i)} \leq O(\phi(s_i)) + O(\rho t \cdot \phi(s_i) \cdot e^{s_i}) =$

$e^{-\Omega(s_i^2)} = O(e^{-\Omega((\log q)^\epsilon)})$.

Putting everything together we have,

$$\begin{aligned}
& \sum_{i=1}^{k} \Pr[Y \geq s_i \mid X \in [t, 2t]] \\
\leq & \sum_{i=1}^{k} \Pr[Z \geq s_i - 4\rho t] \\
= & \sum_{i: s_i \leq (\log q)^{\epsilon/2}} \Pr[Z \geq s_i - 4\rho t] + \sum_{i: s_i \geq (\log q)^{\epsilon/2}} \Pr[Z \geq s_i - 4\rho t] \qquad (3.1) \\
= & \sum_{i: s_i \leq (\log q)^{\epsilon/2}} O(\nu_i) + \sum_{i: s_i \geq (\log q)^{\epsilon/2}} O(e^{-\Omega((\log q)^\epsilon)}) \\
\leq & O(k e^{-\Omega((\log q)^\epsilon)}) + \sum_{i: s_i \leq (\log q)^{\epsilon/2}} O(\nu_i) = O(1)
\end{aligned}$$

where the last line uses $k \leq \log q$ and $\sum_{i=1}^{k} \nu_i \leq 1$.

□

3.3.2 Dictatorship Test

We will create an instance of Graph Vertex Pricing where the vertex set consists of two disjoint hypercubes L, R where $L = R = \mathbb{F}_q^n$. The instance will have the property that dictator pricing functions have good profit in the coupon model. On the other hand, if there is a pair of pricing functions $f_L : L \mapsto \mathbb{R}$, $f_R : R \mapsto \mathbb{R}$ which has sufficiently high profit then we will show that f_L and f_R have a common influential co-ordinate.

Formally, we will describe the Graph Vertex Pricing instance $\mathcal{I}(G(V, E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$ where $V = L \cup R$ as above and E is given by the randomized procedure in Figure 3.1. The weight w_e of an edge e corresponds

Figure 3.1: Dictatorship test for Graph Vertex Pricing

.

1. Generate $\boldsymbol{x} \in \mathbb{F}_q^n$ uniformly at random and $\boldsymbol{x}' \sim_\rho \boldsymbol{x}$ where $\rho := (\log q)^{-2/3}$.
2. For $j \in [\log q]$, let $\boldsymbol{y}^j \in \mathbb{F}_q^n$ be defined as $\boldsymbol{y}^j := (\vec{2^j} - \boldsymbol{x})_q$, where $\vec{2^j} := \{2^j, \ldots, 2^j\} \in \mathbb{F}_q^n$.
3. Add $t/2^j$ hyper edges between $(\boldsymbol{x}'_L, \boldsymbol{y}^j_R)$ each of budget 2^j for every $j \in \{1, 2, \ldots, k\}$ where $k := \log t$, $t := \sqrt{q}$.

to the probability with which it was generated. Note that the total weight of all edges is at most $2t$.

For $\boldsymbol{x} \in \mathbb{F}_q^n$, we will denote by $\boldsymbol{x}_L$ as its copy in L and $\boldsymbol{x}_R$ as its copy in R.

Theorem 3.3.2. *The following holds for the dictatorship test described above:*

- **Completeness:** *Let $f : L \cup R \mapsto \mathbb{R}$ be of the form*

$$f(\boldsymbol{x}_L) = f(\boldsymbol{x}_R) = \boldsymbol{x}_i$$

for some $i \in [n]$ then $\mathbf{profit}^+_{\mathcal{I}}(f) = \Omega(\rho t \log t) = \Omega(t(\log q)^{1/3})$

- **Soundness:** *Let $f_L : L \mapsto \mathbb{R}, f_R : R \mapsto \mathbb{R}$. Then there is a $\tau = \tau(q)$ small enough and $k = k(q)$ large enough such that if*

$$\min(\mathbf{Inf}^k_i(f_L), \mathbf{Inf}^k_i(f_R)) \leq \tau$$

for all $i \in [n]$ then $\mathbf{profit}^+_{\mathcal{I}}(f_L, f_R) = O(t)$

Proof. **Completeness:** Let $f(\boldsymbol{x}_L) = f(\boldsymbol{x}_R) = \boldsymbol{x}_i$. For a hyper edge e, let $f(e) := f(\boldsymbol{y}_R) + f(\boldsymbol{x}_L)$.

$$\begin{aligned}\mathbf{profit}^+_{\mathcal{I}}(f) &= \sum_e I(0 \le f(e) \le b_e) \cdot w_e \cdot f(e) \\ &= \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{x}' \sim_\rho \boldsymbol{x}} \left[\sum_{j=1}^{k} I(0 \le \boldsymbol{y}_i^j + \boldsymbol{x}'_i \le 2^j) \cdot \frac{(\boldsymbol{y}_i^j + \boldsymbol{x}'_i) \cdot t}{2^j} \right]\end{aligned}$$

Now note that $\boldsymbol{y}_i^j + \boldsymbol{x}_i \in \{2^j, 2^j + q\}$. Furthermore, whenever $\boldsymbol{x}_i \ge 2^j$ it holds that $\boldsymbol{y}_i^j + \boldsymbol{x}_i = q + 2^j$. This happens for all j with probability at least $1 - t/q \ge 1/2$ over the choice of $\boldsymbol{x}$. With probability at least ρ over the choice of $\boldsymbol{x}'$, we have $\boldsymbol{x}'_i = \boldsymbol{x}_i$. Thus, with probability at least $\rho/2$ we have $\boldsymbol{y}_i^j + \boldsymbol{x}'_i = 2^j$.

Thus, $\mathbf{profit}^+_{\mathcal{I}}(f) \ge \Omega(\rho t k) = \Omega(\rho t \log t) = \Omega(t (\log q)^{1/3})$.

Soundness:

For a hyper-edge e, let $f(e) := f(\boldsymbol{y}_R) + f(\boldsymbol{x}_L)$.

We first show that it suffices to work with $\mathbb{F}_q$-valued pricing functions.

Definition 3.3.3. *($\mathbb{F}_q$-valued pricing)*

Let $\mathcal{I}(G(V,E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$ be an instance of VERTEX PRICING *or* HIGHWAY PRICING. *For a function $f : V \mapsto \mathbb{F}_q$, the price for an edge $e \in E$ is defined as $f(e) := (\sum_{v \in e} f(v))_q$. Given the price of each edge, $\mathbf{profit}_{\mathcal{I}}(f)$ and $\mathbf{profit}^+_{\mathcal{I}}(f)$ are defined in the usual manner.*

lemma 3.3.4. *Let $f'_L := \lfloor f_L \rfloor$, $f'_R := \lfloor f_R \rfloor$ then*

$$\mathbf{profit}^+_{\mathcal{I}}(f_L, f_R) \le \mathbf{profit}^+_{\mathcal{I}}(f'_L, f'_R) + 2 \cdot \sum_{e \in E} w_e = \mathbf{profit}^+_{\mathcal{I}}(f'_L, f'_R) + O(t)$$

Proof. Let $f'(e) := f'_R(\boldsymbol{y}) + f'_L(\boldsymbol{x})$

For every edge $e \in E$, $f'(e) \leq f(e) \leq f'(e) + 2$. Thus,

$$\begin{aligned}
\textbf{profit}^+_{\mathcal{I}}(f_L, f_R) &= \textstyle\sum_e I(0 \leq f(e) \leq b_e) \cdot w_e \cdot f(e) \\
&\leq \textstyle\sum_e I(0 \leq f'(e) \leq b_e) \cdot w_e \cdot (f'(e) + 2) \\
&\leq \textbf{profit}^+_{\mathcal{I}}(f'_L, f'_R) + 2 \cdot \textstyle\sum_{e \in E} w_e
\end{aligned}$$

□

lemma 3.3.5. *Let f_L, f_R be integral pricing functions for the instance given by the dictatorship test, and let $f'_L := (f_L)_q$, $f'_R := (f_R)_q$. Then*

$$\textbf{profit}^+_{\mathcal{I}}(f_L, f_R) \leq \textbf{profit}^+_{\mathcal{I}}(f'_L, f'_R) + \textbf{profit}^+_{\mathcal{I}}(f'_L - q, f'_R)$$

Proof. Fix an edge $e = (\boldsymbol{x}_L, \boldsymbol{y}_R)$ with budget b_e. f gets a non-zero profit on e if and only if $0 \leq f(e) \leq b_e$. Since $f'_R(\boldsymbol{y}) + f'_L(\boldsymbol{x}) = (f_R(\boldsymbol{y}) + f_L(\boldsymbol{x}))_q$, we must have $f'_R(\boldsymbol{y}) + f'_L(\boldsymbol{x}) \in \{f(e), f(e) + q\}$. In either case, one of (f'_L, f'_R) and $(f'_L - q, f'_R)$ has the same profit as f on e, which immediately implies the lemma.

□

Thus, it suffices to prove the soundness for $f_L : L \mapsto \mathbb{F}_q, f_R : R \mapsto \mathbb{F}_q$.

We now arithmetize the profit.

$$
\begin{aligned}
& \mathbf{profit}^{+}_{\mathcal{I}}(f_L, f_R) \\
= \;& \sum_{e \in E : 0 \le f(e) \le b_e} w_e \cdot f(e) \\
= \;& \mathop{\mathbf{E}}_{\boldsymbol{x}, \boldsymbol{x}' \sim_\rho \boldsymbol{x}} \left[\sum_{j=1}^{k} I(0 \le f_R(\boldsymbol{y}^j) + f_L(\boldsymbol{x}') \le 2^j) \cdot \frac{(f_R(\boldsymbol{y}^j) + f_L(\boldsymbol{x}')) \cdot t}{2^j} \right] \\
= \;& \mathop{\mathbf{E}}_{\boldsymbol{x}, \boldsymbol{x}' \sim_\rho \boldsymbol{x}} \left[\sum_{j=1}^{k} \sum_{0 \le b+a \le 2^j} f_R^b(\boldsymbol{y}^j) \cdot f_L^a(\boldsymbol{x}') \cdot \frac{(b+a) \cdot t}{2^j} \right] \\
= \;& \sum_{j=1}^{k} \sum_{0 \le b+a \le 2^j} \mathop{\mathbf{E}}_{\boldsymbol{x}} \left[f_R^b(\boldsymbol{y}^j) \cdot (T_\rho f_L)^a(\boldsymbol{x}) \cdot \frac{(b+a) \cdot t}{2^j} \right]
\end{aligned}
$$

Let $g_j : \mathbb{F}_q^n \mapsto \mathbb{F}_q$ be defined as $g_j(\boldsymbol{x}) := f_R(\boldsymbol{y}^j)$ where $\boldsymbol{y}^j$ is as in Step 2 of the Test. It is easy to see that $\mathbf{E}[g_j^a] = E[f_R^a]$ for all $a \in \mathbb{F}_q$. This gives,

$$
\mathbf{profit}^{+}_{\mathcal{I}}(f_L, f_R) = t \cdot \sum_{j=1}^{k} \sum_{0 \le b+a \le 2^j} \mathbb{S}_\rho(f_L^a, g_j^b) \frac{(b+a)}{2^j}
$$

For $l \in \{1, 2, \ldots, k\}$ Let

$$
F_{j,l}^a(\boldsymbol{x}) := \sum_{b : 2^{l-1} < b+a \le 2^l} g_j^b(\boldsymbol{x})
$$

Equivalently, $F_{j,l}^a(\boldsymbol{x}) = 1$ if $2^{l-1} < g_j(\boldsymbol{x}) + a \le 2^l$ and 0 otherwise.

It is clear that $\sum_{l=1}^{k} \mathbf{E}[F_{j,l}^a] \le 1$. Also, for every l, $\mathbf{E}[F_{j,l}^a]$ is independent of j since $\mathbf{E}[g_j^b] = \mathbf{E}[f_R^b]$ is independent of j for each $b \in \mathbb{F}_q$. Let $\mu^a := E[f_L^a]$ and $\nu_l^a := \mathbf{E}[F_{j,l}^a]$.

Note that $\mathbf{Inf}_i^k(F_{j,l}^a) \le \mathbf{Inf}_i^k(f_R) \le \tau$ for all $i \in [n]$. Similarly, $\mathbf{Inf}_i^k(f_L^a) \le \mathbf{Inf}_i^k(f_L) \le \tau$.

We thus have,

$$
\begin{aligned}
& \mathbf{profit}^{+}_{\mathcal{I}}(f_L, f_R) \\
= \;& t \cdot \sum_{j=1}^{k} \sum_{0 \le b+a \le 2^j} \mathbb{S}_\rho(f_L^a, g_j^b) \frac{(b+a)}{2^j} \\
\le \;& t \cdot \sum_{j=1}^{k} \sum_{a \in \mathbb{F}_q} \sum_{l=1}^{j} \mathbb{S}_\rho(f_L^a, F_{j,l}^a) \frac{2^l}{2^j} \\
\le \;& t \cdot \textstyle\sum_{a \in \mathbb{F}_q} \displaystyle\sum_{j=1}^{k} \sum_{l=1}^{j} \Lambda_\rho(\mu^a, \nu_l^a) \frac{2^l}{2^j} + o(1) \\
& \text{(Choosing } \delta = (tqk^2)^{-1} \text{ in Theorem 3.2.18)} \\
= \;& t \cdot \textstyle\sum_{a \in \mathbb{F}_q} \displaystyle\sum_{l=1}^{k} \Lambda_\rho(\mu^a, \nu_l^a) \sum_{j=l}^{k} \frac{2^l}{2^j} + o(1) \\
\le \;& 2t \cdot \textstyle\sum_{a \in \mathbb{F}_q} \displaystyle\sum_{l=1}^{k} \Lambda_\rho(\mu^a, \nu_l^a) + o(1) \\
= \;& 2t \cdot \textstyle\sum_{a \in \mathbb{F}_q : \mu_a \ge (q \log q)^{-1}} \displaystyle\sum_{l=1}^{k} \Lambda_\rho(\mu^a, \nu_l^a) \\
& + 2t \cdot \textstyle\sum_{a \in \mathbb{F}_q : \mu_a \le (q \log q)^{-1}} \displaystyle\sum_{l=1}^{k} \Lambda_\rho(\mu^a, \nu_l^a) \\
\le \;& 2t \cdot \textstyle\sum_{a \in \mathbb{F}_q : \mu_a \ge (q \log q)^{-1}} \displaystyle\sum_{l=1}^{k} \Lambda_\rho(\mu^a, \nu_l^a) \\
& + 2t \cdot \textstyle\sum_{a \in \mathbb{F}_q : \mu_a \le (q \log q)^{-1}} \displaystyle\sum_{l=1}^{k} \mu^a \\
\le \;& 2t \cdot \textstyle\sum_{a \in \mathbb{F}_q : \mu_a \ge (q \log q)^{-1}} \displaystyle\sum_{l=1}^{k} \Lambda_\rho(\mu^a, \nu_l^a) + O(t) \qquad \text{(Using } k \le \log q\text{)} \\
\le \;& 2t \cdot \textstyle\sum_{a \in \mathbb{F}_q} O(\mu^a) + O(t) \qquad \text{(Using Lemma 3.3.1)} \\
= \;& O(t)
\end{aligned}
$$

□

Figure 3.2: Reduction from UNIQUE GAMES to GRAPH VERTEX PRICING

1. Pick a random edge $e = (u, w) \in E'$.
2. Generate $\boldsymbol{x} \in \mathbb{F}_q^n$ uniformly at random and $\boldsymbol{x}' \sim_\rho \boldsymbol{x}$ where $\rho := (\log q)^{-(2/3)}$.
3. For $j \in [\log q]$, let $\boldsymbol{y}^j \in \mathbb{F}_q^n$ be defined as $\boldsymbol{y}^j := (\vec{2^j} - \boldsymbol{x})_q$, where $\vec{2^j} := \{2^j, \ldots, 2^j\} \in \mathbb{F}_q^n$.
4. Add $t/2^j$ edges between $((u, \boldsymbol{x}'), (w, \pi_{uw}(\boldsymbol{y}^j)))$ each of budget 2^j for each $j \in \{1, 2, \ldots, k\}$ where $k := \log t$, $t := \sqrt{q}$ and $\pi(x) := (x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n})$.

3.3.3 Reduction from Unique Games

Given a bipartite UNIQUE-GAMES instance $\mathcal{U}(G'(U, W, E'), [n], \{\pi_e\}_{e \in E})$ we create an instance of GRAPH VERTEX PRICING $\mathcal{I}(G(V, E), \{b_e \mid e \in E\}, \{w_e \mid e \in E\})$ where $V = U \times \mathbb{F}_q^n \cup W \times \mathbb{F}_q^n$ and E is defined by the randomized procedure in Figure 5.4.2. The weight w_e of an edge $e \in E$ corresponds to the probability with which it was generated and the budgets are as specified in the procedure.

Theorem 3.3.6. *The following holds for the* GRAPH VERTEX PRICING *instance constructed above:*

- **Completeness:** *If $OPT(\mathcal{U}) \geq 1 - \epsilon$ then there is an assignment of prices $f : V \mapsto \mathbb{R}$ such that $\mathbf{profit}^+_{\mathcal{I}}(f) = \Omega(\rho t \log t) = \Omega(t(\log q)^{1/3})$.*
- **Soundness:** *There is an $\eta = \eta(q)$ small enough such that if $OPT(\mathcal{U}) \leq \eta$ then for every assignment of prices $f : V \mapsto \mathbb{R}$,*

$$\mathbf{profit}^+_{\mathcal{I}}(f) = O(t)$$

Since q can be arbitrarily large, this implies Theorem 1.4.2.

Proof. For a fixed edge $e = (u, w) \in E'$, the instance $\mathcal{I}$ of Figure 5.4.2 restricted to $u \times \mathbb{F}_q^n \cup w \times \mathbb{F}_q^n$ is same as the one constructed by Figure 3.1 up to renumbering of labels according to π_{uw}. Formally, let $L = R = \mathbb{F}_q^n$, let $f_L : L \mapsto \mathbb{R}$ be defined as $f_L^e(\boldsymbol{x}) := f(u, \boldsymbol{x})$ and $f_R^e : R \mapsto \mathbb{R}$ be defined as $f_R(\boldsymbol{x}) := f(w, \pi_{uw}(\boldsymbol{x}))$. Then it is clear that

$$\mathbf{profit}_{\mathcal{I}}^{+}(f) = \mathop{\mathbf{E}}_{e \in E'}[\mathbf{profit}_{\mathcal{I}}^{+}(f_L^e, f_R^e)]$$

where $\mathbf{profit}_{\mathcal{I}}^{+}(f_L^e, f_R^e)$ refers to the profit of (f_L^e, f_R^e) on $\mathcal{I}$ restricted to e.

- **Completeness:**

  Let $L : U \cup W \mapsto [n]$ be a labeling which satisfies $1 - \epsilon$ fraction of the constraints. We define the pricing function $f : V \mapsto \mathbb{R}$ as $f(u, \boldsymbol{x}) := \boldsymbol{x}_{L(u)}$.

  If e is satisfied by L, then $f_L^e(\boldsymbol{x}) = f_R^e(\boldsymbol{x}) = \boldsymbol{x}_i$ for some $i \in [n]$. By Theorem 3.3.2 we get that $\mathbf{profit}_{\mathcal{I}}^{+}(f_L^e, f_R^e)$ is at least $\Omega(\rho t \log t)$.

  Thus, the overall profit is at least $(1 - \epsilon)\Omega(\rho t \log t) = \Omega(\rho t \log t)$.

- **Soundness:**

  We will show that if $\mathbf{profit}_{\mathcal{I}}^{+}(f) = \omega(t)$ then there is a randomized labeling strategy to the Unique Games instance which in expectation satisfies more than η fraction of the edges.

  Note that the profit for any $e = (u, w) \in E'$ is bounded by $O(t \log t)$. So if $\mathbf{profit}_{\mathcal{I}}^{+}(f) = \omega(t)$ then for at least $1/(\log t)$ fraction of the edges $e \in E'$ we have $\mathbf{profit}_{\mathcal{I}}^{+}(f_L^e, f_R^e) = \omega(t)$. For such e by Theorem 3.3.2 we have that $\mathbf{Inf}_i^k(f_L^e)$ and $\mathbf{Inf}_i^k(f_R^e)$ are both larger than τ for some $i \in [n]$. By definition of f_L^e and f_R^e this implies $\mathbf{Inf}_i^k(f_u)$ and $\mathbf{Inf}_{\pi_{uw}(i)}^k(f_v)$ are both larger than τ where f_u is f restricted to $u \times \mathbb{F}_q^n$.

For each $u \in U \cup W$, let

$$\mathbf{Inf}(u) := \{i \in [n] \mid \mathbf{Inf}_i(f_u) \geq \tau\}$$

The labeling strategy is to assign for each $u \in U \cup W$ a label independently and uniformly at random from $\mathbf{Inf}(u)$. If $\mathbf{Inf}(u)$ is empty, assign an arbitrary label to u. By Fact 3.2.15 (and since we can work with $\mathbb{F}_q$-valued functions) , the size of $\mathbf{Inf}(u)$ is at most k/τ for each u.

The above analysis shows that the expected fraction of edges satisfied by this labeling is at least

$$\frac{1}{\log t} \cdot \left(\frac{\tau}{k}\right)^2$$

Since this quantity depends only on q, we can choose $\eta = \eta(q)$ small enough so that more than η fraction of the edges are satisfied. This completes the proof.

□

3.4 NP-Hardness of Vertex Pricing$_3$

In this section we restate and prove Theorem 1.4.5.

Theorem. *(Theorem 1.4.5 restated)* VERTEX PRICING$_3$ *under the coupon or the discount model is NP-hard to approximate to factor* $\Omega(\log \log \log n)$.

Figure 3.3: Dictator test $\mathcal{T}_\pi$

.

Test $\mathcal{T}_\pi$.

1. Generate $\boldsymbol{x}$ to be uniformly random from $\mathbb{Z}_q^K$.
2. Generate $\boldsymbol{y}$ to be a uniformly random from $\mathbb{Z}_q^L$.
3. For each $j \in [L]$ and $i = \pi(j)$, set

$$\boldsymbol{z}_j = \begin{cases} q - (\boldsymbol{x}_i + \boldsymbol{y}_j) & \text{if } \boldsymbol{x}_i + \boldsymbol{y}_j \leq q \\ 2q - (\boldsymbol{x}_i + \boldsymbol{y}_j) & \text{if } \boldsymbol{x}_i + \boldsymbol{y}_j > q \end{cases}$$

4. Let $\boldsymbol{x'} \sim_{1-\epsilon} x$, $\boldsymbol{y'} \sim_{1-\epsilon} y$ and $\boldsymbol{z'} \sim_{1-\epsilon} z$ for $\epsilon = 1/q$
5. Randomly generate a integer $k \in [\sqrt{q}]$,
6. Let $\boldsymbol{z''} = \boldsymbol{z'} + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor$ and add a customer interested in three items $\boldsymbol{x'}, \boldsymbol{y'}, \boldsymbol{z''}$ with budget $\lfloor \sqrt{q}/k \rfloor$

In Section 3.4.1 we present the dictatorship test for Vertex Pricing$_3$ while in Section 3.4.2 we compose the dictatorship test with Label Cover to obtain our final reduction. Theorem 1.4.5 follows by combining Theorem 3.4.4, Theorem 3.4.5 and Theorem 3.4.6 and observing that $\text{Opt}^D(\mathcal{I}) \leq \text{Opt}^C(\mathcal{I})$ for any Vertex Pricing instance $\mathcal{I}$.

3.4.1 Dictatorship test for Vertex Pricing$_3$

Let $K, L \in \mathbb{Z}^+$ and $L > K$. For $\pi : [L] \to [K]$ being a projection, we define the Vertex Pricing$_3$ instance corresponding to the Dictator Test $\mathcal{T}_\pi$ on $V = [q]^L \cup [q]^K$ in Figure 3.3.

Below are two key properties of $\mathcal{T}_\pi$.

Theorem 3.4.1 (completeness)**.** *For every $j \in [L]$ and $i = \pi(j)$, if we set $f(t) = t_i - q/3$ for $t \in \mathbb{Z}_q^K$ and $g(r) = r_j - q/3$ for $r \in \mathbb{Z}_q^L$, then* $\mathbf{profit}_{\mathcal{T}_\pi}(f, g) \geq \Omega(\log q)$*.*

Proof. It is easy to check with probability at least 1/3, we have that $\boldsymbol{x}_i + \boldsymbol{y}_j \leq q$ for randomly and independently generated $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$ after the second step in Figure 3.3. Therefore for these $\boldsymbol{x}_i, \boldsymbol{y}_j$, we have $\boldsymbol{z}_j = q - \boldsymbol{x}_i - \boldsymbol{y}_j$ at the third step.

Since $\boldsymbol{x}'_i, \boldsymbol{y}'_j$ and $\boldsymbol{z}'_j$ is generated by perturbing $\boldsymbol{x}_i, \boldsymbol{y}_j, \boldsymbol{z}_j$ with probability at most $1/q$, by union bound we know that with probability at least $1/3 - 3/q$, we still have $\boldsymbol{x}'_i + \boldsymbol{y}'_j + \boldsymbol{z}'_j = q$. Also since $\boldsymbol{z}'_j$ follows the uniform distribution on $\mathbb{Z}_q$, we know with probability at least $1/3 - 3/q - 1/\sqrt{q} \geq 1/4$ it holds that $\boldsymbol{z}'_j \leq q - \sqrt{q}$. Let us call these $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'$ "good".

For "good" $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'$, we know that $f(\boldsymbol{x}') = \boldsymbol{x}_i - q/3$ and $g(\boldsymbol{y}') = \boldsymbol{y}_j - q/3$ and $g(\boldsymbol{z}' \oplus_q + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor) = \boldsymbol{z}_j + \lfloor \sqrt{q}/k \rfloor - q/3$ (since $\boldsymbol{z}'_j \leq q - \sqrt{q}$). Thus, $f(\boldsymbol{x}') + g(\boldsymbol{y}') + g(\boldsymbol{z}' + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor) = \lfloor \sqrt{q}/k \rfloor$. Therefore, for good $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'$, we made $\lfloor \sqrt{q}/k \rfloor$ on the buyer interested in $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}' + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor$. Since we have at least 1/4 "good" $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'$, we made profit $1/4 \cdot 1/\sqrt{q} \cdot \sum_k \lfloor \sqrt{q}/k \rfloor = \Omega(\log q)$ on them.

We also need to bound the negative profit as f and g can also take negative value. This is when the case that $f(\boldsymbol{x}) + f(\boldsymbol{y}) + f(\boldsymbol{z} + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor) < 0$. We claim that we may only lose money in one of the following two cases:

1. $\boldsymbol{x}_i \neq \boldsymbol{x}'_i$ or $\boldsymbol{y}_j \neq \boldsymbol{y}'_j$ or $\boldsymbol{z}_j \neq \boldsymbol{z}'_j$.

2. $\boldsymbol{x}_i + \boldsymbol{y}_j <= \sqrt{q}$.

To verify this, if case 1 and case 2 do not happen, then $\boldsymbol{x} = \boldsymbol{x}'$, $\boldsymbol{y} = \boldsymbol{y}'$ and $\boldsymbol{z} = \boldsymbol{z}'$ and $\boldsymbol{x}_i + \boldsymbol{y}_j > \sqrt{q}$. When $\sqrt{q} < \boldsymbol{x}_i + \boldsymbol{y}_j \leq q$,we know that $f(\boldsymbol{x}) + f(\boldsymbol{y}) +$

$f(\boldsymbol{z}+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)=\lfloor\sqrt{q}/k\rfloor$; when $\boldsymbol{x}_i+\boldsymbol{y}_j>q$, we know that $f(\boldsymbol{x})+f(\boldsymbol{y})+f(\boldsymbol{z}+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)\geq\boldsymbol{x}_i+\boldsymbol{y}_j-q+[\boldsymbol{z}_j+\lfloor\sqrt{q}/k\rfloor]_q>0$.

By union bound we know that case 1 happens with probability at most $3/q$; and case 2 could only happen when $x_i+y_j\leq\sqrt{q}$ and this also happens with probability at most $\frac{1}{q}$ (because we must have both x,y less than $\sqrt{q}$ which occur with probability $1/q$). Overall, we know that only for $4/q$ fraction of the x,y,z generated, we can possibly lose money. Also since $f,g>=-q/3$ by definition, we can lose money for at most q on each customer. Therefore, we can at most lose profit $4/q\cdot q<=4$.

Overall the profit we have on f and g is still $\Omega(\log q)$. □

Theorem 3.4.2 (soundness)**.** *If for some function $f:\mathbb{Z}_q^K\to\mathbb{R}$ and $g:\mathbb{Z}_q^L\to\mathbb{R}$, we have $\mathbf{profit}^+_{\mathcal{T}}(f,g)\geq 12$, then we can have a (randomized) way of decoding f in to a coordinate $i_f\in[K]$ and g into a coordinate $j_g\in[L]$ such that the*

$$Pr(\pi(j_g)=i_f)\geq 1/q^6.$$

In addition, the decoding of f is independent of g or π; i.e., there is one decoding procedure that works for all possible π,g. Similarly the decoding procedure of g is independent of f and π

Proof. First, we can assume that the pricing function is integer with profit loss 3 simply by taking the integer part of f and g.. To see this, for any fixed $\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}''$, we have some *real* pricing function $f(\boldsymbol{x}')+g(\boldsymbol{y}')+g(\boldsymbol{z}'')$, then $\lfloor f(\boldsymbol{x}')\rfloor+\lfloor g(\boldsymbol{y}')\rfloor+\lfloor g(\boldsymbol{z}'')\rfloor\geq f(\boldsymbol{x}')+g(\boldsymbol{y}')+g(\boldsymbol{z}'')-3$. Therefore, we have that

$$\mathbf{profit}^+(\lfloor f\rfloor,\lfloor g\rfloor)\geq\mathbf{profit}^+(f,g)-3$$

Further, we restrict the range of f, g by modulo q as in Definition 3.3.3. For any function f, we define $\tilde{f} = [\lfloor f \rfloor]_q$ $\tilde{g} = [\lfloor g \rfloor]_q$. Following lemma illustrates the relationship between the profit of using $\lfloor f \rfloor, \lfloor g \rfloor$ and $\tilde{f}, \tilde{g}$.

lemma 3.4.3. $\mathbf{profit}^+_{\mathcal{T}_\pi}(\lfloor f \rfloor, \lfloor g \rfloor) \leq \mathbf{profit}^+_{\mathcal{T}_\pi}(\tilde{f}, \tilde{g}) + \mathbf{profit}^+_{\mathcal{T}_\pi}(\tilde{f} - q/3, \tilde{g} - q/3) + \mathbf{profit}^+_{\mathcal{T}_\pi}(\tilde{f} - 2q/3, \tilde{g} - 2q/3)$

Proof. We know that if for some buyer who is interested in $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}''$ with budget $\lfloor \sqrt{q}/k \rfloor$, then if $f(\boldsymbol{x}') + g(\boldsymbol{y}') + g(\boldsymbol{z}'') \leq \lfloor \sqrt{q}/k \rfloor$. Then it must be the case that $0 < \tilde{f}(\boldsymbol{x}') + \tilde{g}(\boldsymbol{y}') + \tilde{g}(\boldsymbol{z}'') < \lfloor \sqrt{q}/k \rfloor$ or $q < \tilde{f}(\boldsymbol{x}') + \tilde{g}(\boldsymbol{y}') + \tilde{g}(\boldsymbol{z}'') < q + \lfloor \sqrt{q}/k \rfloor$ or $2q < \tilde{f}(\boldsymbol{x}') + \tilde{g}(\boldsymbol{y}') + \tilde{g}(\boldsymbol{z}'') < 2q + \lfloor \sqrt{q}/k \rfloor$. Therefore, at least one of the pricing strategy among $(\tilde{f}, \tilde{g})$, $(\tilde{f} - q/3, \tilde{g} - q/3)$ or $(\tilde{f} - 2q/3, \tilde{g} - 2q/3)$ will have the same profit as (f, g) on $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}''$. $\square$

It remains to bound $\mathbf{profit}^+_{\mathcal{T}}(\tilde{f}, \tilde{g}) + \mathbf{profit}^+_{\mathcal{T}}(\tilde{f} - q/3, \tilde{g} - q/3) + \mathbf{profit}^+(\tilde{f} - 2q/3, \tilde{g} - 2q/3)$. We will only show how to bound $\mathbf{profit}^+(\tilde{f}, \tilde{g}) \leq 3$ and the other proof is similar.

Let us also introduce the notion $\tilde{f}_i : \mathbb{Z}_q^L \to \{0, 1\}$ as indicator function of whether $\tilde{f} = i$. We similarly define $\tilde{g}_i = i$. We also write $\tilde{f}_i = \sum \tilde{f}_i^S$ and $\tilde{g}_i = \sum g_i^S$ as the Efron-Stein Decomposition of $\tilde{f}_i, \tilde{g}_i$. .

We can represent the $\mathbf{profit}^+(\tilde{f}, \tilde{g})$ as follows:

$$\mathbf{profit}^+_{\mathcal{T}_\pi}(\tilde{f}, \tilde{g}) \leq \mathop{\mathbf{E}}_{\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'', k} \left[\sum_{0 < i+j+l \leq \lfloor \sqrt{q}/k \rfloor} (i + j + l) \cdot \tilde{f}_i(\boldsymbol{x}') \tilde{g}_j(\boldsymbol{y}') \tilde{g}_l(\boldsymbol{z}'') \right] \quad (3.2)$$

Now we plug in the Efron-Stein Decomposition of $\tilde{f}_i, \tilde{g}_j, \tilde{g}_l$:

$$(3.2) = \mathop{\mathbf{E}}_{k}\left[\sum_{0<i+j+l\leq\lfloor\sqrt{q}/k\rfloor}\ \sum_{T_1,T_2\subseteq[L],S\subseteq[K],}(i+j+l)\mathop{\mathbf{E}}_{\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}''}[\tilde{f}_i^S(\boldsymbol{x}')\tilde{g}_j^{T_1}(\boldsymbol{y}')\tilde{g}_l^{T_2}(\boldsymbol{z}'')\right] \tag{3.3}$$

We know that $\boldsymbol{x}'$ is independent of $\boldsymbol{y}'$ and $\boldsymbol{x}'$ is independent of $\boldsymbol{z}''$. By the second property of Efron-Stein Decomposition, we must have that $T_1 = T_2 = T$ as otherwise ,$\mathbf{E}_{\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}''}[\tilde{f}_i^S(x)\tilde{g}_j^{T_1}(y)\tilde{g}_l^{T_2}(\boldsymbol{z}'')] = 0$. For the similar reason if we write the set $\pi(T) = \{\pi(j) \mid j \in T\}$, then we must also have $S \subseteq \pi(T)$. We know then

$$(3.3) = \mathop{\mathbf{E}}_{k}\left[\sum_{0<i+j+l\leq\lfloor\sqrt{q}/k\rfloor}\ \sum_{\substack{T\subseteq[L]\\ S\subseteq\pi(T)}}(i+j+l)\mathop{\mathbf{E}}_{\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}'}[\tilde{f}_i^S(\boldsymbol{x}')\tilde{g}_j^{T}(\boldsymbol{y}')\tilde{g}_l^{T}(\boldsymbol{z}'+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)]\right] \tag{3.4}$$

$$= \mathop{\mathbf{E}}_{k}\left[\sum_{0<i+j+l\leq\lfloor\sqrt{q}/k\rfloor}\ \sum_{\substack{T\subseteq[L]\\ S=\emptyset}}(i+j+l)\mathop{\mathbf{E}}_{\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}'}[\tilde{f}_i^S(\boldsymbol{x}')\tilde{g}_j^{T}(\boldsymbol{y}')\tilde{g}_l^{T}(\boldsymbol{z}'+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)]\right] \tag{3.5}$$

$$+ \mathop{\mathbf{E}}_{k}\left[\sum_{0<i+j+l\leq\lfloor\sqrt{q}/k\rfloor}\ \sum_{\substack{T\subseteq[L]\\ \emptyset\subsetneq S\subseteq\pi(T)}}(i+j+l)\mathop{\mathbf{E}}_{\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}'}[\tilde{f}_i^S(\boldsymbol{x}')\tilde{g}_j^{T}(\boldsymbol{y}')\tilde{g}_l^{T}(\boldsymbol{z}'+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)]\right] \tag{3.6}$$

In the second equality above, we divide (**??**) into two parts. (3.5) is when $S = \emptyset$ and the (3.6) is when $S \neq \emptyset$ and we will bound the two parts individually.

Case i) First we prove that $(3.5) \leq 2$. Notice that $\tilde{f}_i^{\emptyset}(\boldsymbol{x}')$ is a constant, say $\tilde{f}_i^{\emptyset}$. Thus (3.5) is equal to

$$\mathop{\mathbf{E}}_{k}\left[\sum_{0<i+j+l\leq\lfloor\sqrt{q}/k\rfloor}\ \sum_{T\subseteq[L]}(i+j+l)\ \tilde{f}_i^{\emptyset}\mathop{\mathbf{E}}_{\boldsymbol{y}',\boldsymbol{z}'}[g_j^T(\boldsymbol{y}')\tilde{g}_l^T(\boldsymbol{z}'+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)]\right] \tag{3.7}$$

Figure 3.4: Test $\mathcal{T}'_\pi$

Test $\mathcal{T}'$.

1. generate $\boldsymbol{y}', \boldsymbol{z}', k$ according to their margin distribution on $\mathcal{T}$.
2. **set $\boldsymbol{x}'$ to be uniformly over $\mathbb{Z}_q^K$ independent with $\boldsymbol{y}', \boldsymbol{z}', k$.**
3. let us write $\boldsymbol{z}'' = \boldsymbol{z}' + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor$ and add a customer interested in three items $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}''$ with budget $\lfloor \sqrt{q}/k \rfloor$

A crucial observation is that above expression can be viewed as the the profit of $\lfloor \tilde{f} \rfloor, \lfloor \tilde{g} \rfloor$on Dictator test $\mathcal{T}'$ defined in Figure 3.4.The test $\mathcal{T}'_\pi$ is different from $\mathcal{T}_\pi$ only in the generation of $\boldsymbol{x}'$ which is set to be independent with $\boldsymbol{y}'$ and $\boldsymbol{z}''$.Then when we calculate the profit as in equation (**??**), we would get that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'' \sim \mathcal{T}'_\pi} [\tilde{f}_i^S(\boldsymbol{x}') \tilde{g}_j^{T_1}(\boldsymbol{y}') \tilde{g}_l^T(\boldsymbol{z}'')] \neq 0$$

only when $S = \emptyset$ and $T_1 = T_2$. This is exactly the same as (3.7). It remains to bound the profit on $\mathcal{T}'_\pi$.

The next important observation on $\mathcal{T}'_\pi$ here is that actually $\boldsymbol{y}', \boldsymbol{z}'$ and $\boldsymbol{y}', \boldsymbol{z}''$ has the same marginal distribution. Therefore, we can further simplify the test as $\mathcal{T}''_\pi$ defined in Figure 3.5. As for the test $\mathcal{T}''_\pi$, for every fixed $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'$, and suppose $\sqrt{q}/(k_0+1) < \tilde{f}(x) + \tilde{g}(y) + \tilde{g}(z) \leq \sqrt{q}/k_0$. Then such a pricing function will only have profit when $k \leq (k_0+1)$ and for that fixed $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'$, the expected profit conditioned on k (being randomly generated from $[\sqrt{q}]$) is at most $\frac{1}{\sqrt{q}} \cdot \sqrt{q}/k_0 \cdot (k_0+1) \leq 2$..

Figure 3.5: Test $\mathcal{T}''_\pi$

Test $\mathcal{T}''$.

1. generate $\boldsymbol{y}', \boldsymbol{z}', k$ according to their marginal distribution on $\mathcal{T}$.
2. **set $\boldsymbol{x}'$ to be uniformly over $\mathbb{Z}_q^K$ independent with $\boldsymbol{y}', \boldsymbol{z}', k$.**
3. add a customer interested in three items $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'$ with budget $\lfloor\sqrt{q}/k\rfloor$.

Overall, we proved that

$$(3.5) = \mathbf{profit}^+_{\mathcal{T}'_\pi}(\lfloor\tilde{f}\rfloor, \lfloor\tilde{g}\rfloor) = \mathbf{profit}^+_{\mathcal{T}''_\pi}(\lfloor\tilde{f}\rfloor, \lfloor\tilde{g}\rfloor) \leq 2.$$

Case ii) It remains to bound (3.6). Let us prove this by contradiction. We will show that if (3.6) ≥ 1, then there exists a way of decoding f and g as described in Theorem 3.4.2.

We know that (3.6) is equal to

$$\mathop{\mathbf{E}}_k\left[\sum_{0<i+j+l\leq\lfloor\sqrt{q}/k\rfloor}\ \sum_{\substack{T\subseteq[L]\\ \emptyset\subsetneq S\subseteq\pi(T)}}(i+j+l)\mathop{\mathbf{E}}_{\boldsymbol{x}',\boldsymbol{y}',\boldsymbol{z}'}[\tilde{f}_i^S(\boldsymbol{x}')\tilde{g}_j^T(\boldsymbol{y}')\tilde{g}_l^T(\boldsymbol{z}'+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)]\right] \tag{3.8}$$

Let us now focus on the second sum within the expectation for fixed i, j and l. Notice that $\boldsymbol{x}' \sim_{1-\epsilon} \boldsymbol{x}$, $\boldsymbol{y}' \sim_{1-\epsilon} \boldsymbol{y}'$ and $\boldsymbol{z}' \sim_{1-\epsilon} \boldsymbol{z}$, by the definition of the noise operator, we have that the second sum in (3.8) is equal to

$$\sum_{\substack{T\subseteq[L]\\ \emptyset\subsetneq S\subseteq\pi(T)}}(i+j+l)\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}}[T_{1-\epsilon}\tilde{f}_i^S(\boldsymbol{x})T_{1-\epsilon}\tilde{g}_j^T(\boldsymbol{y})T_{1-\epsilon}\tilde{g}_l^T(\boldsymbol{z}+\vec{1}\cdot\lfloor\sqrt{q}/k\rfloor)] \tag{3.9}$$

Let us write $\tilde{f}^{\pi(T)} = \sum_{\emptyset \subsetneq S \subseteq \pi(T)} \hat{\hat{\tilde{f}}}(S)$. Then we know that (3.9) is equal to

$$\sum_{T \subseteq [L]} (i+j+l) \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{z}} [T_{1-\epsilon} \tilde{f}_i^{\pi(T)}(\boldsymbol{x}) T_{1-\epsilon} \tilde{g}_j^T(\boldsymbol{y}) T_{1-\epsilon} \tilde{g}_l^T(\boldsymbol{z} + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor)]$$

Using Cauchy Inequality in the expectation of above formula and noticing that $\boldsymbol{x}, \boldsymbol{y}$ and $\boldsymbol{y}, \boldsymbol{z}$ are independent, we have that (3.9) is at most

$$\begin{aligned} &(i+j+l) \sum_{T \subseteq [L]} \sqrt{\mathop{\mathbf{E}}_{\boldsymbol{x}} [T_{1-\epsilon} \tilde{f}_i^{\pi(T)}(\boldsymbol{x})^2] \mathop{\mathbf{E}}_{\boldsymbol{y}} [T_{1-\epsilon} \tilde{g}_j^T(\boldsymbol{y})^2] \mathop{\mathbf{E}}_{\boldsymbol{z}} [T_{1-\epsilon} \tilde{g}_l^T(\boldsymbol{z} + \vec{1} \cdot \lfloor \sqrt{q}/k \rfloor)^2]} \\ &\qquad = (i+j+l) \sum_{T \subseteq [L]} \|T_{1-\epsilon} \tilde{f}_i^{\pi(T)}\|_2 \cdot \|T_{1-\epsilon} \tilde{g}_j^T\|_2 \cdot \|T_{1-\epsilon} \tilde{g}_l^T\|_2 \end{aligned} \tag{3.10}$$

We can further use Cauchy inequality to bound the inside sum for every i, j, l :

$$\begin{aligned} &\sum_{T \subseteq [L]} \|T_{1-\epsilon} \tilde{f}_i^{\pi(T)}\|_2 \cdot \|T_{1-\epsilon} \tilde{g}_j^T\|_2 \cdot \|T_{1-\epsilon} \tilde{g}_l^T\|_2 \\ \leq &\sqrt{\sum_{T \subseteq [L]} \|T_{1-\epsilon} \tilde{f}_i^{\pi(T)}\|_2^2 \cdot \|T_{1-\epsilon} \tilde{g}_j^T\|_2^2 \sum_{T \subseteq [L]} \|T_{1-\epsilon} \tilde{g}_l^T\|_2^2} \end{aligned} \tag{3.11}$$

Notice that $\sum_{T \subseteq [L]} \|T_{1-\epsilon} \tilde{g}_l^T\|_2^2 = \|T_{1-\epsilon} \tilde{g}_l\|_2^2 \leq 1$. Therefore, we have that

$$1 \leq (3.6) \leq (3.10) \leq \sum_{0 < i+j+l \leq \sqrt{q}} (i+j+l) \sqrt{\sum_{T \subseteq [L]} \|T_{1-\epsilon} f_i^{\pi(T)}\|_2^2 \cdot \|T_{1-\epsilon} g_j^T\|^2}$$

Thus there must exist some i_0, j_0 such that

$$\sum_{T \subseteq [L]} \|T_{1-\epsilon} \tilde{f}_{i_0}^{\pi(T)}\|_2^2 \cdot \|T_{1-\epsilon} \tilde{g}_{j_0}^T\|^2 = \sum_{\substack{T \subseteq [L] \\ \emptyset \subsetneq S \subseteq \pi(T)}} (1-\epsilon)^{|S|+|T|} \|\tilde{f}_{i_0}^S\|_2^2 \|\tilde{g}_{j_0}^T\|_2^2 \geq 1/q^5.$$

It is easy to verify that $\sum_{i\in\mathbb{F}_q, S\subseteq[K]} \|\tilde{f}_i^S\|_2^2 = 1$ and $\sum_{j\in\mathbb{F}_q, T\subseteq[L]} \|\tilde{g}_j^T\|_2^2 = 1$. Below is the randomized decoding procedure for f and g. For f, we sample (i, S) with probability $\|\tilde{f}_i^S\|_2^2$ and randomly output a $m_f \in S$. Similarly for g, we randomly sample (j, T) with probability $\|g_j^T\|_2^2$ and randomly output a coordinate n_g in T.

Then the probability that $\pi(n_g) = m_f$ is at least

$$\Pr(\pi(n_g) = m_f) \geq \sum_{\emptyset\subsetneq S\subseteq\pi(T), T\subseteq[L]} \frac{\|\tilde{f}_{i_0}^S\|_2^2 \cdot \|\tilde{g}_{j_0}^T\|^2}{T} \tag{3.12}$$

Above we only count the probability when (i_0, S) and (j_0, T) are selected such that $\emptyset \subsetneq S \subseteq \pi(T)$. Then we know that for randomly picked elements $m_f \in S$ and $n_g \in T$, with probability at least $1/T$, we have $\pi(n_g) = m_f$.

Also notice that $1/|T| \geq \epsilon \cdot (1-\epsilon)^{|T|}$. Since $\epsilon = 1/q$, we have that

$$\Pr(\pi(n_g) = m_f) \geq (3.12) \geq \sum_{\emptyset\subsetneq S\subseteq\pi(T), T\subseteq[L]} \epsilon(1-\epsilon)^T \|\tilde{f}_{i_0}^S\|_2^2 \cdot \|\tilde{g}_{j_0}^T\|^2 \geq 1/q^6. \tag{3.13}$$

□

3.4.2 Reduction from Label Cover

The starting point of our hardness reduction is the following Label Cover problem (Definition 1.3.2).

The following is known about the hardness of approximating Label Cover.

Theorem 3.4.4 ([MR10]). *For some positive constant $c > 0$, it is NP-hard to distinguish a label cover problem vertices of n vertices and alphabet size $K, L \leq \sqrt{\log n}$.*

- *YES Case:* $\mathrm{Opt}(\mathcal{L}) = 1$.

Figure 3.6: Hardness reduction from LABEL COVER

1. randomly sample an edge $e = (u, v) \in E$.
2. sample $\boldsymbol{x}', \boldsymbol{y}', \boldsymbol{z}'', k$ according to the $\mathcal{T}_{\pi_e}$
3. add a customer interested in $(v, \boldsymbol{x}')$,$(u, \boldsymbol{y}')$, $(u, \boldsymbol{z}'')$ with budget k.

- *NO Case:* $\mathrm{Opt}(\mathcal{L}) \le \delta$ *for* $\delta = 1/(\log\log n)^c$.

Given a LABEL COVER instance $\mathcal{L}(G(U, V, E), [L], [K], \{\pi_e | e \in E\})$, we construct a VERTEX PRICING$_3$ instance $\mathcal{I}$ with its vertices defined over $(U \times [q]^L \cup V \times [q]^R)$ for $q = (\log\log n)^{c/10}$. The construction of edges and budget is described in in Figure 3.6. It is easy to verify the reduction is in polynomial time.We identify each item by (w, r) for $w \in U, r \in [q]^L$ or $w \in V, r \in [q]^K$. Let us denote the corresponding pricing function to be $\{f_u : [q]^L \to \mathbb{R} | u \in U\} \cup \{f_v : [q]^K \to \mathbb{R} \mid v \in V\}$: we price items (w, r) by $f_w(r)$. We will prove that the reduction has the following properties(Theorem 3.4.5 and Theorem 3.4.6).

Theorem 3.4.5 (Completeness)**.** *If there is a labelling that satisfies every edge for* $\mathcal{L}$*, then* $\mathrm{Opt}^D(\mathcal{I}) \ge \Omega(\log q)$.

Proof. If there is a labelling $\sigma : U \to [L], V \to [K]$, then we can simply use the following pricing function: for $w \in U \cup V$, we use the price function $f_w(t) = t_{\sigma(w)} - q/3$.

By the completeness property of $\mathcal{T}_{\pi_e}$, we know that such a pricing strategy will have profit $\Omega(\log q)$. □

Theorem 3.4.6 (Soundness)**.** *If* $\mathrm{Opt}^C(I) \ge 13$*, then there is a labelling that satisfies more than* $1/q^7 \ge \delta$ *fraction of the edges.*

Proof. (soundness) Suppose $\mathrm{Opt}^C(\mathcal{I}) \geq 13$, notice that the maximum profit is at most $\sqrt{q}$ with each customer, then by an average argument, we know that for $1/q$ fraction of the edges (u, v) picked, we have that f_u, f_v has expected profit at least $13 - 1/\sqrt{q} > 12$. Let us call these (u, v) to be good.

Then by Theorem 3.4.2, there is way of decoding the f_u, f_v into coordinate i_u, i_v with the promise that $\Pr(\pi(i_u) = i_v) \geq 1/q^6$. Then if we just label each "good" edge (u, v) with i_u, i_v, such a labelling will satisfy at least $1/q \cdot 1/q^6 = 1/q^7$ fraction of the edges. $\square$

Chapter 4

Integrality gap for 2-to-1 Label Cover

In this chapter we restate and prove Theorem 1.4.6 as well as state and prove our results about 2-TO-2 LABEL COVER and α LABEL COVER.

Theorem. *(Theorem 1.4.6 restated) There are instances of 2-TO-1 LABEL COVER with alphabet size K and optimum value $O(1/\sqrt{\log K})$ on which the SDP has value 1. The instances have size $2^{\Omega(K)}$.*

In Section 4.1 we describe some notation and tools that will be used throughout the chapter. In Section 4.2 we state and prove our result for 2-TO-2 LABEL COVER. Section 4.3 describes the integrality gap for 2-TO-1 LABEL COVER, Theorem 1.4.6 follows from the completeness and soundness analysis therein. In Section 4.4 we show that every integrality gap instance for 2-TO-1 LABEL COVER with sufficiently many edges can be converted to an integrality gap instance for α LABEL COVER. We close the chapter with some discussion in Section 4.5 about our results and future work.

4.1 Preliminaries and Notation

4.1.1 Label Cover Problems

Here we will use a slightly broader definition of LABEL COVER since Definition 1.3.2 doesn't include one of the problems we are interested in, namely, α LABEL COVER.

Definition 4.1.1. *A* LABEL COVER *instance* $\mathcal{L}$ *is defined by a tuple* $((V, E), R, \Psi)$. *Here* (V, E) *is a graph,* R *is a positive integer and* Ψ *is a set of constraints (relations), one for each edge:* $\Psi = \{\psi_e \subseteq \{1, \ldots, R\}^2 \mid e \in E\}$. *A labeling* A *is a mapping* $A : V \to [R]$. *We say that an edge* $e = (u, v)$ *is* satisfied *by* A *if* $(A(u), A(v)) \in \psi_e$. *We define:*

$$\mathrm{OPT}(\mathcal{L}) = \max_{A:V\to[R]} \Pr_{e=(u,v)\in E}[(A(u), A(v)) \in \psi_e]$$

Here the probability is over the uniform distribution of edges, i.e. each edge is equally likely to be picked.

In Figure 4.1, we write down a natural SDP relaxation for the LABEL COVER problem. The relaxation is over the vector variables $\mathbf{z}_{(v,i)}$ for every vertex $v \in V$ and label $i \in [R]$.

Our goal in this work is to study integrality gaps for the above SDP for various special cases of the LABEL COVER problem. We already discussed the UNIQUE GAMES and 2-TO-1 GAMES conjectures on the hardness of certain very special cases of LABEL COVER. We now discuss two other variants of LABEL COVER and their conjectured inapproximability.

Definition 4.1.2. *A constraint* $\psi \subseteq \{1, \ldots, 2R\}^2$ *is said to be a* 2-to-2 constraint *if*

$$
\begin{array}{lll}
\text{maximize} & \mathop{\mathbf{E}}_{e=(u,v)\in E}\Big[\sum_{i,j\in\psi_e}\left\langle \mathbf{z}_{(u,i)},\mathbf{z}_{(v,j)}\right\rangle\Big] & \\
\text{subject to} & \sum_{i\in[R]}\left\|\mathbf{z}_{(v,i)}\right\|^2 = 1 & \forall\, v\in V \\
 & \left\langle \mathbf{z}_{(v,i)},\mathbf{z}_{(v,j)}\right\rangle = 0 & \forall\, i\neq j\in[R], v\in V
\end{array}
$$

Figure 4.1: SDP for Label Cover

there are two permutations $\sigma_1,\ \sigma_2 : \{1,\ldots,2R\} \to \{1,\ldots,2R\}$ *such that* $(i,j)\in\psi$ *if and only if* $(\sigma_1(i),\sigma_2(j))\in T$ *where*

$$T := \{(2l-1,2l-1),(2l-1,2l),(2l,2l-1),(2l,2l)\}_{l=1}^{R}.$$

A Label Cover *instance is said to be* 2*-to-*2 *if all its constraints are* 2*-to-*2 *constraints.*

A constraint $\psi\subseteq\{1,\ldots,2R\}^2$ *is said to be an* α-constraint *if there are two permutations* $\sigma_1,\ \sigma_2 : \{1,\ldots,2R\}\to\{1,\ldots,2R\}$ *such that* $(i,j)\in\psi$ *if and only if* $(\sigma_1(i),\sigma_2(i))\in T'$ *where*

$$T' := \{(2l-1,2l-1),(2l-1,2l),(2l,2l-1)\}_{l=1}^{R}.$$

A Label Cover *instance is said to be* α *if all its constraints are* α *constraints.*

Conjecture 4.1.3. *[DMR09] (***2-to-2 Conjecture***) For any* $\delta>0$*, it is NP-hard to decide whether a* 2-to-2 Label Cover *instance* $\mathcal{L}$ *has* $\mathrm{OPT}(\mathcal{L})=1$ *or has* $\mathrm{OPT}(\mathcal{L})\leq\delta$*.*

It was shown in [DMR09] that the 2-to-2 Conjecture is no stronger than the

2-to-1 Conjecture.

Conjecture 4.1.4. *[DMR09] (α **Conjecture**) For any $\delta > 0$, it is NP-hard to decide whether an α* Label Cover *instance $\mathcal{L}$ has* $\mathrm{OPT}(\mathcal{L}) = 1$ *or has* $\mathrm{OPT}(\mathcal{L}) \leq \delta$.

4.1.2 Fourier Analysis

Let $\mathcal{V} := \{f : \mathbb{F}_2^k \to \mathcal{R}\}$ denote the vector space of all real functions on $\mathbb{F}_2^k$, where addition is defined as point-wise addition. We always think of $\mathbb{F}_2^k$ as a probability space under the uniform distribution, and therefore use notation such as $\|f\|_p := \mathbf{E}_{x \in \mathbb{F}_2^k}[|f(x)|^p]$. For $f, g \in \mathcal{F}$, we also define the inner product $\langle f, g \rangle := \mathbf{E}[f(x)g(x)]$.

For any $\alpha \in \mathbb{F}_2^k$ the Fourier character $\chi_\alpha \in \mathcal{F}$ is defined by $\chi_\alpha(x) := (-1)^{\alpha \cdot x}$. The Fourier characters form an orthonormal basis for $\mathcal{V}$ with respect to the above inner product, hence every function $f \in \mathcal{V}$ has a unique representation as $f = \sum_{\alpha \in \mathbb{F}_2^k} \widehat{f}(\alpha)\chi_\alpha$, where the Fourier coefficient $\widehat{f}(\alpha) := \langle f, \chi_\alpha \rangle$.

We also sometimes identify each α with the set $S_\alpha = \{i \mid \alpha_i = 1\}$ and denote the Fourier coefficients as $\widehat{f}(S)$. We use the notation $|\alpha|$ for $|S_\alpha|$, the number of coordinates where α is 1.

The following well-known fact states that the norm of a function on $\mathbb{F}_2^k$ is unchanged when expressing it in the basis of the characters.

Proposition 4.1.5. *(**Parseval's identity**) For any $f : \mathbb{F}_2^k \to \mathbb{R}$, $\sum_{\alpha \in \mathbb{F}_2^k} \widehat{f}(\alpha)^2 = \|f\|_2^2 = \mathbf{E}[f(x)^2]$.*

We shall also need the following result due to Talagrand (Proposition 2.3 in [Tal94]), proven using hypercontractivity methods:

Theorem 4.1.6. *Suppose $F : \mathbb{F}_2^k \to \mathbb{R}$ has $\mathbf{E}[F] = 0$. Then*

$$\sum_{\alpha \in \mathbb{F}_2^k \setminus \{0\}} \widehat{F}(\alpha)^2/|\alpha| = O\left(\frac{\|F\|_2^2}{\ln(\|F\|_2/(e\|F\|_1))}\right)$$

More precisely, we will need the following easy corollary:

Corollary 4.1.7. *If $F : \mathbb{F}_2^k \to \{0,1\}$ has mean $1/K$, then*

$$\widehat{F}(0)^2 + \sum_{\alpha \in \mathbb{F}_2^k \setminus \{0\}} \widehat{F}(\alpha)^2/|\alpha| = O\left(1/(K \log K)\right)$$

Proof. We have $\widehat{F}(0)^2 = \mathbf{E}[F]^2 = 1/K^2 \leq O(1/(K \log K))$, so we can disregard this term. As for the sum, we apply Theorem 4.1.6 to the function $F' = F - 1/K$, which has mean 0 as required for the theorem. It is easy to calculate that $\|F'\|_2 = \Theta(1/\sqrt{K})$ and $\|F'\|_1 = \Theta(1/K)$, and so the result follows. $\square$

4.2 Integrality Gap for 2-to-2 Games

In this section we prove the following integrality gap for 2-to-2 Label Cover.

Theorem 4.2.1. *There are instances of* 2-to-2 Label Cover *with alphabet size K and optimum value $O(1/\log K)$ on which the SDP has value* 1*. The instances have size $2^{\Omega(K)}$.*

The theorem follows by the completeness and soundness analysis of the instance described below. The instance for 2-to-1 Label Cover will be an extension of the one below. In fact, our analysis of OPT in the 2-to-1 case will follow simply by reducing it to the analysis of OPT for the 2-to-2 instance below.

The vertex set V in our instance is same as the vertex set of the Unique Games integrality gap instance constructed in [KV05]. Let $\mathcal{F} := \{f : \mathbb{F}_2^k \mapsto \{-1,1\}\}$ denote the family of all boolean functions on $\mathbb{F}_2^k$. For f, $g \in \mathcal{F}$, define the product fg as $(fg)(x) := f(x)g(x)$. Consider the equivalence relation $\sim$ on $\mathcal{F}$ defined as $f \sim g \Leftrightarrow \exists \alpha \in \mathbb{F}_2^k$ *s.t.* $f \equiv g\chi_\alpha$. This relation partitions $\mathcal{F}$ into equivalence classes $\mathcal{P}_1, \ldots, \mathcal{P}_n$, with $n := 2^K/K$. The vertex set V consists of the equivalence classes $\{\mathcal{P}_i\}_{i\in[n]}$. We denote by $[\mathcal{P}_i]$ the lexicographically smallest function in the class $\mathcal{P}_i$ and by $\mathcal{P}_f$, the class containing f.

We take the label set to be of size K and identify $[K]$ with $\mathbb{F}_2^k$ in the obvious way. For each tuple of the form (γ, f, g) where $\gamma \in \mathbb{F}_2^k \setminus \{0\}$ and $f, g \in \mathcal{F}$ are such that $(1+\chi_\gamma)f \equiv (1+\chi_\gamma)g$, we add a constraint $\psi_{(\gamma,f,g)}$ between the vertices $\mathcal{P}_f$ and $\mathcal{P}_g$. Note that the condition on f and g is equivalent to saying that $\chi_\gamma(x) = 1 \implies f(x) = g(x)$. If $f = [\mathcal{P}_f]\chi_\alpha$ and $g = [\mathcal{P}_g]\chi_\beta$ and if $A : [n] \to \mathbb{F}_2^k$ denotes the labeling, the relation $\psi_{(\gamma,f,g)}$ is defined as

$$(A(\mathcal{P}_f), A(\mathcal{P}_g)) \in \psi_{(\gamma,f,g)} \quad \Leftrightarrow \quad (A(\mathcal{P}_f)+\alpha) - (A(\mathcal{P}_g)+\beta) \ \in \ \{0, \gamma\}.$$

Note that for any $\omega \in \mathbb{F}_2^k$, the constraint maps the labels $\{\omega, \omega+\gamma\}$ for $\mathcal{P}_f$ to the labels $\{\omega+\alpha-\beta, \omega+\alpha-\beta+\gamma\}$ for $\mathcal{P}_g$ in a 2-to-2 fashion. We denote the set of all constraints by Ψ. We remark that, as in [KV05], our integrality gap instances contain multiple constraints on each pair of vertices.

4.2.1 SDP Solution

We give below a set of feasible vectors $\mathbf{z}_{(\mathcal{P}_i,\alpha)} \in \mathbb{R}^K$ for every equivalence class $\mathcal{P}_i$ and every label α, achieving SDP value 1. Identifying each coordinate with an

$x \in \mathbb{F}_2^k$, we define the vectors as

$$\mathbf{z}_{(\mathcal{P}_i,\alpha)}(x) \;:=\; \frac{1}{K}([\mathcal{P}_i]\chi_\alpha)(x).$$

It is easy to check that $\left\|\mathbf{z}_{(\mathcal{P}_i,\alpha)}\right\|^2 = 1/K$ for each of the vectors, which satisfies the first constraint. Also, $\mathbf{z}_{(\mathcal{P}_i,\alpha)}$ and $\mathbf{z}_{(\mathcal{P}_i,\beta)}$ are orthogonal for $\alpha \neq \beta$ since

$$\left\langle \mathbf{z}_{(\mathcal{P}_i,\alpha)}, \mathbf{z}_{(\mathcal{P}_i,\beta)} \right\rangle \;=\; \frac{1}{K^2}\left\langle [\mathcal{P}_i]\chi_\alpha, [\mathcal{P}_i]\chi_\beta \right\rangle \;=\; \frac{1}{K^2}\left\langle \chi_\alpha, \chi_\beta \right\rangle \;=\; 0$$

using the fact that $[\mathcal{P}_i]^2 = 1$. The following claim proves that the solution achieves SDP value 1.

Claim 4.2.2. *For any edge e indexed by a tuple (γ, f, g) with $f(1+\chi_\gamma) \equiv g(1+\chi_\gamma)$, we have*

$$\sum_{\omega_1,\omega_2 \in \psi_{(\gamma,f,g)}} \left\langle \mathbf{z}_{(\mathcal{P}_f,\omega_1)}, \mathbf{z}_{(\mathcal{P}_g,\omega_2)} \right\rangle \quad = \quad 1.$$

Proof. Let $f \equiv [\mathcal{P}_f]\chi_\alpha$ and $g \equiv [\mathcal{P}_g]\chi_\beta$. Then, $(\omega_1, \omega_2) \in \psi_e$ iff $(\omega_1+\alpha)-(\omega_2+\beta) \in \{0, \gamma\}$. Therefore, the above quantity equals (divided by 2 to account for double counting of ω)

$$\begin{aligned}
& \frac{1}{2}\cdot\sum_\omega \Big(\left\langle \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha)}, \mathbf{z}_{(\mathcal{P}_g,\omega+\beta)} \right\rangle + \left\langle \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha+\gamma)}, \mathbf{z}_{(\mathcal{P}_g,\omega+\beta)} \right\rangle \\
& \qquad\qquad + \left\langle \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha)}, \mathbf{z}_{(\mathcal{P}_g,\omega+\beta+\gamma)} \right\rangle + \left\langle \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha+\gamma)}, \mathbf{z}_{(\mathcal{P}_g,\omega+\beta+\gamma)} \right\rangle\Big) \\
= \;\; & \frac{1}{2}\sum_\omega \left\langle \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha)} + \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha+\gamma)}, \mathbf{z}_{(\mathcal{P}_f,\omega+\beta)} + \mathbf{z}_{(\mathcal{P}_f,\omega+\beta+\gamma)} \right\rangle
\end{aligned} \tag{4.1}$$

However, for each ω, we have $\mathbf{z}_{(\mathcal{P}_f,\omega+\alpha)} + \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha+\gamma)} \;=\; \mathbf{z}_{(\mathcal{P}_f,\omega+\beta)} + \mathbf{z}_{(\mathcal{P}_f,\omega+\beta+\gamma)}$, since for all coordinates x,

$$\begin{aligned}\mathbf{z}_{(\mathcal{P}_f,\omega+\alpha)}(x) \; + \; \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha+\gamma)}(x) \quad &= \quad \frac{1}{K}([\mathcal{P}_f]\chi_{\omega+\alpha}(x) \; + \; [\mathcal{P}_f]\chi_{\omega+\alpha+\gamma}(x)) \\ = \frac{1}{K}(f(x) + f\chi_\gamma)\chi_\omega(x) &= \frac{1}{K}(g(x) + g\chi_\gamma)\chi_\omega(x) \\ = \quad \frac{1}{K}([\mathcal{P}_g]\chi_{\omega+\beta}(x) \; + \; [\mathcal{P}_g]\chi_{\omega+\beta+\gamma}(x)) \quad &= \quad \mathbf{z}_{(\mathcal{P}_f,\omega+\beta)}(x) \; + \; \mathbf{z}_{(\mathcal{P}_f,\omega+\beta+\gamma)}(x).\end{aligned}$$

This completes the proof as the value of (4.1) then becomes

$$\frac{1}{2}\sum_{\omega} \left\|\mathbf{z}_{(\mathcal{P}_f,\omega+\alpha)} + \mathbf{z}_{(\mathcal{P}_f,\omega+\alpha+\gamma)}\right\|^2 = \frac{1}{2}\sum_{\omega}\left(\left\|\mathbf{z}_{(\mathcal{P}_f,\omega+\alpha)}\right\|^2 + \left\|\mathbf{z}_{(\mathcal{P}_f,\omega+\alpha+\gamma)}\right\|^2\right) = 1.$$

□

4.2.2 Soundness

We now prove that any labeling of the instance described above, satisfies at most $O(1/\log K)$ fraction of the constraints. Let $A : V \to \mathbb{F}_2^k$ be a labeling of the vertices. We extend it to a labeling of all the functions in $\mathcal{F}$ by defining $A([\mathcal{P}_i]\chi_\alpha) := A(\mathcal{P}_i) + \alpha$.

For each $\alpha \in \mathbb{F}_2^k$, define $A_\alpha : \mathcal{F} \to \{0, 1\}$ to be the indicator that A's value is

α. By definition, the fraction of constraints satisfied by the labeling A is

$$\begin{aligned} \mathrm{val}(A) &= \mathop{\mathbf{E}}_{(\gamma,f,g)\in\Psi}\left[\sum_{\alpha\in\mathbb{F}_2^k} A_\alpha(f)(A_\alpha(g)+A_{\alpha+\gamma}(g))\right] \\ &= \mathop{\mathbf{E}}_{(\gamma,f,g)\in\Psi}\left[\sum_{\alpha\in\mathbb{F}_2^k} A_\alpha(f)(A_\alpha(g)+A_\alpha(g\chi_\gamma))\right] \\ &= 2\cdot \mathop{\mathbf{E}}_{(\gamma,f,g)\in\Psi}\left[\sum_{\alpha\in\mathbb{F}_2^k} A_\alpha(f)(A_\alpha(g)\right] \qquad (4.2) \end{aligned}$$

where the last equality used the fact that for every tuple $(\gamma, f, g) \in \Psi$, we also have $(\gamma, f, g\chi_\gamma) \in \Psi$.

Note that the extended labeling $A : \mathcal{F} \to \mathbb{F}_2^k$ takes on each value in $\mathbb{F}_2^k$ an equal number of times. Hence

$$\mathop{\mathbf{E}}_f[A_\alpha(f)] = \Pr_f[A(f) = \alpha] = 1/K \quad \text{for each } \alpha \in \mathbb{F}_2^k. \qquad (4.3)$$

For our preliminary analysis, we will use only this fact to show that for any $\alpha \in \mathbb{F}_2^k$ it holds that

$$\mathop{\mathbf{E}}_{(\gamma,f,g)\in\Psi}[A_\alpha(f)A_\alpha(g)] \le O(1/(K\log K)). \qquad (4.4)$$

It will then follow that the soundness (4.2) is at most $O(1/\log K)$. Although this tends to 0, it does so only at a rate proportional to the logarithm of the alphabet size, which is $K = 2^k$.

Beginning with the left-hand side of (4.4), let's write $F = A_\alpha$ for simplicity. We think of the functions f and g being chosen as follows. We first choose a function $h : \gamma^\perp \to \{-1, 1\}$. Note that $\gamma^\perp \subseteq \mathbb{F}_2^k$ is the set of inputs where $\chi_\gamma = 1$ and hence

$f = g$, and we let $f(x) = g(x) = h(x)$ for $x \in \gamma^\perp$. The values of f and g on the remaining inputs are chosen independently at random. Then

$$\begin{aligned} \mathop{\mathbf{E}}_{(\gamma,f,g)\in\Psi}[F(f)F(g)] &= \mathop{\mathbf{E}}_{\gamma} \mathop{\mathbf{E}}_{h:\gamma^\perp\to\{-1,1\}} \left[\mathop{\mathbf{E}}_{f,g|h}[F(f)F(g)] \right] \\ &= \mathop{\mathbf{E}}_{\gamma} \mathop{\mathbf{E}}_{h:\gamma^\perp\to\{-1,1\}} \left[\mathop{\mathbf{E}}_{f|h}[F(f)] \mathop{\mathbf{E}}_{g|h}[F(g)] \right]. \end{aligned} \tag{4.5}$$

Let us write $P_\gamma F(h)$ for $\mathbf{E}_{f|h} F(f)$, which is also equal to $\mathbf{E}_{g|h} F(g)$. We now use the Fourier expansion of F. Note that the domain here is $\{-1,1\}^K$ instead of $\mathbb{F}_2^k$. To avoid confusion with characters and Fourier coefficients for functions on $\mathbb{F}_2^k$, we will index the Fourier coefficients below by sets $S \subseteq \mathbb{F}_2^k$. Given an $f \in V$, we'll write f^S for $\prod_{x\in S} f(x)$ (which is a Fourier character for the domain $\{-1,1\}^K$). Now for fixed γ and h,

$$P_\gamma F(h) = \mathop{\mathbf{E}}_{f|h}[F(f)] = \mathop{\mathbf{E}}_{f|h}\left[\sum_{S\subseteq\mathbb{F}_2^k} \widehat{F}(S) f^S\right] = \sum_{S\subseteq\mathbb{F}_2^k} \widehat{F}(S)\cdot \mathop{\mathbf{E}}_{f|h}[f^S].$$

The quantity $\mathbf{E}_{f|h}[f^S]$ is equal to h^S if $S \subseteq \gamma^\perp$ and is 0 otherwise. Thus, using the Parseval identity, we deduce that (4.5) equals

$$\mathop{\mathbf{E}}_{\gamma} \mathop{\mathbf{E}}_{h:\gamma^\perp\to\{-1,1\}} \left[(P_\gamma F(h))^2\right] = \mathop{\mathbf{E}}_{\gamma}\left[\sum_{S\subseteq\gamma^\perp} \left(\widehat{F}(S)\right)^2\right] = \sum_{S\subseteq\mathbb{F}_2^k} \Pr_{\gamma}[S\subseteq\gamma^\perp]\cdot\left(\widehat{F}(S)\right)^2.$$

Recalling that $\gamma \in \mathbb{F}_2^k \setminus \{0\}$ is chosen uniformly, we have that

$$\sum_{S\subseteq\mathbb{F}_2^k} \Pr_{\gamma}[S\subseteq\gamma^\perp]\cdot\left(\widehat{F}(S)\right)^2 = \sum_{S\subseteq\mathbb{F}_2^k} 2^{-\dim(S)}\cdot\left(\widehat{F}(S)\right)^2,$$

where we are writing $\dim(S) = \dim(\text{span } S)$ for shortness (and defining $\dim(\emptyset) =$

0). For $|S| \geq 1$ we have $\dim(S) \geq \log_2 |S|$ and hence $2^{-\dim(S)} \geq 1/|S|$. Thus

$$\sum_{S \subseteq \mathbb{F}_2^k} 2^{-\dim(S)} \cdot \widehat{F}(S)^2 \leq \widehat{F}(\emptyset)^2 + \sum_{\emptyset \neq S \subseteq \mathbb{F}_2^k} \widehat{F}(S)^2/|S|.$$

Corollary 4.1.7 shows that this is at most $O(1/(K \log K))$. This completes the proof:

$$\mathrm{val}(A) \;=\; 2 \cdot \sum_{\alpha \in \mathbb{F}_2^k} \mathop{\mathbf{E}}_{(\gamma,f,g) \in \Psi} [A_\alpha(f) A_\alpha(g)] \;\leq\; 2 \cdot \sum_{\alpha \in \mathbb{F}_2^k} 2^{-\dim(S)} \widehat{A_\alpha}(S)^2 \;=\; O(1/\log K)$$

4.3 Integrality gap for 2-to-1 label cover

The instances for 2-to-1 label cover are bipartite. We denote such instances as (U, V, E, R_1, R_2, Π) where $R_2 = 2R_1$ denote the alphabet sizes on the two sides. For a bipartite instance, the label cover SDP can be written as in Figure 4.2 involving vectors $\mathbf{y}_{(u,i)}$ for each $u \in U, i \in [R_1]$ and vectors $\mathbf{z}_{(v,j)}$ for each $v \in v, j \in [R_2]$.

$$\begin{array}{lrll}
\text{maximize} & \mathop{\mathbf{E}}_{e=(u,v)\in E}\Big[\sum_{i \in [R_2]} \big\langle \mathbf{y}_{(u,\pi_e(i))}, \mathbf{z}_{(v,j)} \big\rangle\Big] & & \\
\text{subject to} & \sum_{i \in [R_1]} \left\| \mathbf{y}_{(u,i)} \right\|^2 = 1 & & \forall\, u \in U \\
& \sum_{i \in [R_2]} \left\| \mathbf{z}_{(v,i)} \right\|^2 = 1 & & \forall\, v \in V \\
& \big\langle \mathbf{y}_{(u,i)}, \mathbf{y}_{(u,j)} \big\rangle = 0 & \forall\, i \neq j \in [R_1], u \in U & \\
& \big\langle \mathbf{z}_{(v,i)}, \mathbf{z}_{(v,j)} \big\rangle = 0 & \forall\, i \neq j \in [R_2], v \in V & \blacksquare
\end{array}$$

Figure 4.2: SDP for 2-to-1 Games

4.3.1 Gap Instance

As in the case of 2-to-2 games, the set V consists of equivalence classes $\mathcal{P}_1, \ldots, \mathcal{P}_n$, which partition the set of functions $\mathcal{F} = \{f : \mathbb{F}_2^k \to \{-1, 1\}\}$, according to the equivalence relation $\sim$ defined as $f \sim g \;\Leftrightarrow\; \exists \alpha \in \mathbb{F}_2^k \; s.t. \; f \equiv g\chi_\alpha$. The label set $[R_2]$ is again identified with $\mathbb{F}_2^k$ and is of size $K = 2^k$.

To describe the set U, we further partition the vertices in V according to other equivalence relations. For each $\gamma \in \mathbb{F}_2^k, \gamma \neq 0$, we define an equivalence relation $\cong_\gamma$ on the set $\mathcal{P}_1, \ldots, \mathcal{P}_n$ as

$$\mathcal{P}_i \cong_\gamma \mathcal{P}_j \qquad \Leftrightarrow \qquad \exists f \in \mathcal{P}_i, g \in \mathcal{P}_j \text{ s.t. } f(1 + \chi_\gamma) \equiv g(1 + \chi_\gamma)$$

This is equivalent to saying:

$$\mathcal{P}_i \cong_\gamma \mathcal{P}_j \qquad \Leftrightarrow \qquad \exists f \in \mathcal{P}_i, g \in \mathcal{P}_j \text{ s.t. } fg(x) = -1 \Rightarrow \chi_\gamma(x) = -1 \;\forall x \in \mathbb{F}_2^k$$

This partitions $\mathcal{P}_1, \ldots, \mathcal{P}_n$ (and hence also the set $\mathcal{F}$) into equivalence classes $\mathcal{Q}_1^\gamma, \ldots, \mathcal{Q}_m^\gamma$. Here $m = 2^{K/2+1}/K$ (this is immediate from the second definition and the fact that $n = 2^K/K$) and the partition is different for each γ. The set U has one vertex for each class of the form $\mathcal{Q}_i^\gamma$ for all $i \in [m]$ and $\gamma \in \mathbb{F}_2^k \setminus \{0\}$. As before, we denote by $[\mathcal{Q}_i^\gamma]$ the lexicographically smallest function in the class $\mathcal{Q}_i^\gamma$, and by $\mathcal{Q}_f^\gamma$ the class under $\cong_\gamma$ containing f. Note that if $f \in \mathcal{Q}_i^\gamma$, then there exists a $\beta \in \mathbb{F}_2^k$ such that $f(1 + \chi_\gamma) \equiv [\mathcal{Q}_i^\gamma]\chi_\beta(1 + \chi_\gamma)$.

The label set R_1 has size $K/2$. For each vertex $\mathcal{Q}_i^\gamma \in U$, we think of the labels as pairs of the form $\{\alpha, \alpha + \gamma\}$ for $\alpha \in \mathbb{F}_2^k$. More formally, we identify it with the space $\mathbb{F}_2^k/\langle\gamma\rangle$. We impose one constraint for every pair of the form (γ, f) between the vertices $\mathcal{P}_f$ and $\mathcal{Q}_f^\gamma$. If $f \equiv [\mathcal{P}_f]\chi_\alpha$ and $f(1+\chi_\gamma) \equiv [\mathcal{Q}_i^\gamma]\chi_\beta(1+\chi_\gamma)$, then the corresponding relation $\psi_{(\gamma,f)}$ is defined by requiring that for any labelings $A : V \to [R_2]$ and $B : U \to [R_1]$,

$$(B(\mathcal{Q}_f^\gamma), A(\mathcal{P}_f)) \in \psi_{(\gamma,f)} \quad \Leftrightarrow \quad A(\mathcal{P}_f) + \alpha \ \in \ B(\mathcal{Q}_f^\gamma) + \beta$$

Here, if $B(\mathcal{Q}_f^\gamma)$ is a pair of the form $\{\omega, \omega+\gamma\}$, then $B(\mathcal{Q}_f^\gamma) + \beta$ denotes the pair $\{\omega+\beta, \omega+\gamma+\beta\}$.

4.3.2 SDP Value

As before, we give a set of vectors $\mathbf{y}_{(\mathcal{Q}_i^\gamma,\{\alpha,\alpha+\gamma\})}$ and $\mathbf{z}_{(\mathcal{P}_i,\alpha)}$ in $\mathbb{R}^K$, identifying each coordinate with an $x \in \mathbb{F}_2^k$. We define the vectors as

$$\begin{aligned} \mathbf{y}_{(\mathcal{Q}_i^\gamma,\{\alpha,\alpha+\gamma\})}(x) &:= \frac{1}{K}\left([\mathcal{Q}_i^\gamma]\chi_\alpha(1+\chi_\gamma)\right)(x), \\ \mathbf{z}_{(\mathcal{P}_i,\alpha)}(x) &:= \frac{1}{K}\left([\mathcal{P}_f]\chi_\alpha\right)(x). \end{aligned}$$

We have already shown that $\left\langle \mathbf{z}_{(\mathcal{P}_i,\alpha)}, \mathbf{z}_{(\mathcal{P}_i,\beta)} \right\rangle = 0$ for $\alpha \neq \beta$ and $\left\|\mathbf{z}_{(\mathcal{P}_i,\alpha)}\right\|^2 = 1/K$. It again follows by the orthogonality of characters that for disjoint pairs $\{\alpha, \alpha+\gamma\}$ and $\{\beta, \beta+\gamma\}$, the vectors $\mathbf{y}_{(\mathcal{Q}_i^\gamma,\{\alpha,\alpha+\gamma\})}$ and $\mathbf{y}_{(\mathcal{Q}_i^\gamma,\{\beta,\beta+\gamma\})}$ are orthogonal. It is also easy to verify that $\left\|\mathbf{y}_{(\mathcal{Q}_i^\gamma,\{\alpha,\alpha+\gamma\})}\right\|^2 = 2/K$. Hence, the vectors form a feasible solution.

To show that the SDP value is equal to 1, we consider an arbitrary constraint indexed by the pair (γ, f). Let $f \equiv [\mathcal{P}_f]\chi_\alpha$ and $f(1+\chi_\gamma) \equiv [\mathcal{Q}_i^\gamma]\chi_\beta(1+\chi_\gamma)$. Then for any $\omega \in \mathbb{F}_2^k$, this constraint maps the label $\omega+\alpha$ for $\mathcal{P}_f$ to the pair

$\{\omega+\beta, \omega+\gamma+\beta\}$ for $\mathcal{Q}_f^\gamma$. Hence, the value of the SDP solution on this constraint is given by

$$\sum_{\omega \in \mathbb{F}_2^k} \left\langle \mathbf{y}_{(\mathcal{Q}_i^\gamma, \{\omega+\beta, \omega+\beta+\gamma\})}, \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega)} \right\rangle$$

We will show that for every ω, $\mathbf{y}_{(\mathcal{Q}_i^\gamma, \{\omega+\beta, \omega+\beta+\gamma\})} = \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega)} + \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega+\gamma)}$. This will complete the proof as the above expression then becomes

$$\sum_{\omega \in \mathbb{F}_2^k} \left\langle \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega)} + \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega+\gamma)}, \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega)} \right\rangle \;=\; \sum_{\omega \in \mathbb{F}_2^k} \left\| \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega)} \right\|^2 \;=\; 1$$

To show the vector identity, we simply note that for each coordinate x, we have

$$\begin{aligned}
\mathbf{y}_{(\mathcal{Q}_i^\gamma, \{\omega+\beta, \omega+\beta+\gamma\})}(x) &= \frac{1}{K}\left([\mathcal{Q}_i^\gamma]\chi_\beta(1+\chi_\gamma)\right)(x) \\
&= \frac{1}{K}\left(f(1+\chi_\gamma)\right)(x) \\
&= \frac{1}{K}\left([\mathcal{P}_f]\chi_\alpha + [\mathcal{P}_f]\chi_{\alpha+\gamma}\right)(x) \\
&= \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega)}(x) + \mathbf{z}_{(\mathcal{P}_i, \alpha+\omega+\gamma)}(x).
\end{aligned}$$

4.3.3 Soundness

We now bound the fraction of constraints satisfied by any pair of labelings $A : V \to [K]$ and $B : U \to [K/2]$. Let $\mathbf{1}_{\{\mathcal{E}\}}$ denote the indicator of the event $\mathcal{E}$, and $N(u)$ denote the neighborhood of a vertex $u \in U$. Then, the fraction of constraints satisfied by any assignments A, B, can be bound by an application of

Cauchy-Schwarz as

$$\begin{aligned}
\mathrm{val}(A,B) &= \mathop{\mathbf{E}}_{u\in U}\mathop{\mathbf{E}}_{v\in N(u)}\left[\mathbf{1}_{\{\pi_{uv}(A(v))=B(u)\}}\right] \\
&\leq \left(\mathop{\mathbf{E}}_{u\in U}\left(\mathop{\mathbf{E}}_{v\in N(u)}\left[\mathbf{1}_{\{\pi_{uv}(A(v))=B(u)\}}\right]\right)^2\right)^{1/2} \\
&= \left(\mathop{\mathbf{E}}_{u\in U}\mathop{\mathbf{E}}_{v_1,v_2\in N(u)}\left[\mathbf{1}_{\{\pi_{uv_1}(A(v_1))=B(u)=\pi_{uv_2}(A(v_2))\}}\right]\right)^{1/2} \\
&\leq \left(\mathop{\mathbf{E}}_{u\in U}\mathop{\mathbf{E}}_{v_1,v_2\in N(u)}\left[\mathbf{1}_{\{\pi_{uv_1}(A(v_1))=\pi_{uv_2}(A(v_2))\}}\right]\right)^{1/2}
\end{aligned}$$

Note that if π_{uv_1} and π_{uv_2} are 2-to-1 projections, then the inner quantity in the last expression denotes the value of a 2-to-2 label cover instance, each of whose constraints is defined by two 2-to-1 constraints in the original instance. For the 2-to-1 instance described above, we will show that the inner quantity in fact denotes the fraction of constraints satisfied by A for the 2-to-2 instance described in Section 4.2. This will show that the fraction of constraints satisfied by any assignment in the above 2-to-1 instance can be at most $O(1/\sqrt{\log K})$.

To see this, note that a vertex $u \in U$ and a vertex $v_1 \in V$ can be sampled jointly by picking a pair (γ, f) and taking $u = \mathcal{Q}_f^\gamma$ and $v_1 = \mathcal{P}_f$. Sampling $v_2 \in N(u)$ corresponds to choosing a class $\mathcal{P}_i$ such that for some $\beta \in \mathbb{F}_2^k$ $[\mathcal{P}_i]\chi_\beta(1+\chi_\gamma) \equiv f(1+\chi_\gamma)$. Thus, v_2 can be sampled by choosing a random g such that $f(1+\chi_\gamma) \equiv g(1+\chi_\gamma)$ and taking $v_2 = \mathcal{P}_g$.

Also, if $f \equiv [\mathcal{P}_f]\chi_{\alpha_1}$ and $g \equiv [\mathcal{P}_g]\chi_{\alpha_2}$, then the constraint $\pi_{uv_1}(A(v_1)) = \pi_{uv_2}(A(v_2))$ simply requires that for some $\omega \in \mathbb{F}_2^k$, $A(\mathcal{P}_f) + \alpha_1$ and $A(\mathcal{P}_g) + \alpha_2$

both lie in the set $\{\omega, \omega+\gamma\}$ and hence

$$(A(\mathcal{P}_f)+\alpha_1)-(A(\mathcal{P}_g)+\alpha_2) \in \{0, \gamma\}$$

4.4 From 2-to-1 constraints to α-constraints

In this section we show that any integrality gap instance for 2-to-1 games, with sufficiently many edges, can be converted to an integrality gap instance for games with α-constraints. The following theorem follows by combining Theorem 4.4.2 below and Theorem 1.4.6.

Theorem 4.4.1. *There are instances of α Label Cover with alphabet size K and optimum value $O(1/\sqrt{\log K})$ on which the SDP has value 1. The instances have size $2^{\Omega(K)}$.*

Theorem 4.4.2. *Let $\mathcal{L} = (U, V, E, R, 2R, \Psi)$ be a bipartite instance of 2-to-1 label cover problem with $\mathrm{OPT}(\mathcal{L}) \leq \delta$ and SDP value 1. Also, let $|E| \geq 4(|U| + |V|)\log(R)/\epsilon^2$. Then there exists another instance $\mathcal{L}' = (U, V, E, 2R, \Psi')$ of Label Cover with α-constraints having SDP value 1 and $\mathrm{OPT}(\mathcal{L}') \leq \delta + \epsilon + 1/R$.*

Proof. The proof simply follows by adding R "fake" labels for each vertex $u \in U$, and then randomly augmenting the constraints to make them of the required form. In particular, let the new labels we add for each $u \in U$ be $R+1, \ldots, 2R$. Let $e = (u, v)$ be an edge. Since the constraints in Ψ are 2-to-1 type, there exist permutations $\sigma_{1,e} : [R] \to [R]$ and $\sigma_{2,e} : [2R] \to [2R]$ such that after permuting the labels on each side, the projection π_e maps labels $(2i-1, 2i)$ to i i.e. $\pi_e(\sigma_{2,e}^{-1}(2i-1)) = \pi_e(\sigma_{2,e}^{-1}(2i)) = \sigma_{1,e}^{-1}(i)$.

To incorporate the new labels into the constraint, choose a random bijection $\sigma'_{1,e} : \{R+1, \ldots, 2R\} \to [R]$. We now construct a new permutation $\tilde{\sigma}_{1,e} : [2R] \to [2R]$ as $\tilde{\sigma}_{1,e}(i) = 2\sigma_{1,e}(i) - 1$ if $i \leq R$ and $\tilde{\sigma}_{1,e}(i) = 2\sigma'_{1,e}(i)$ if $i > R$ i.e. the new labels are mapped to the even positions $2, 4, \ldots, 2R$ while the others are mapped to the odd positions.

The original 2-to-1 constraints are satisfied by a labeling A iff the pair $(\tilde{\sigma}_{1,e}(A(u), \sigma_{2,e}(A(v)))$ is of the form $(2i-1, 2i-1)$ or $(2i-1, 2i)$ for some $i \leq R$. We augment the constraint by also allowing $(\tilde{\sigma}_{1,e}(A(u), \sigma_{2,e}(A(v)))$ to be $(2i, 2i-1)$ for some i. Note that if the constraint is satisfied in this way, then u must get one of the new labels. Also, note that the augmentation is random as we choose the map $\sigma'_{1,e}$ independently at random for each edge e.

Given a vector solution $\{\mathbf{y}_{(u,i)}\}_{u \in U, i \in [R]}$ and $\{\mathbf{z}_{(v,j)}\}_{v \in V, j \in [2R]}$ for Ψ, we leave the vectors $\mathbf{z}_{(v,j)}$ unchanged and for each $u \in U$, take $\mathbf{z}_{(u,i)} = \mathbf{y}_{(v,i)}$ if $i \leq R$ and 0 otherwise. It is immediate that the solution is feasible. Also, the value of the objective is the same as the value of the 2-to-1 SDP, as all the additional terms in the objective involve some vector $\mathbf{z}_{(u,i)}$ for some $i > R$ and are hence 0. Thus, the SDP value for the new instance is 1.

To bound the optimal value of any labeling $A : U \cup V \to [2R]$, we split it as

$$\begin{aligned}
\mathop{\mathbf{E}}_{e=(u,v)\in E}\left[\mathbf{1}_{\{(A(u),A(v)) \text{ satisfy } e\}}\right] &= \mathop{\mathbf{E}}_{e=(u,v)\in E}\left[\mathbf{1}_{\{A(u)\leq R\}} \cdot \mathbf{1}_{\{(A(u),A(v)) \text{ satisfy } e\}}\right] \\
&\quad + \mathop{\mathbf{E}}_{e=(u,v)\in E}\left[\mathbf{1}_{\{A(u)> R\}} \cdot \mathbf{1}_{\{(A(u),A(v)) \text{ satisfy } e\}}\right]
\end{aligned}$$

Note that the first term is simply the number of 2-to-1 constraints satisfied by A and it at most δ by assumption.

Also, for any fixed labeling A, the probability over the choice of the random maps $\{\sigma'_{1,e}\}_{e\in E}$, that $(A(u), A(v))$ satisfy e given that $A(u) > R$, is at most $1/R$. By a Chernoff bound, the fraction of edges (u, v) satisfied with $A(u) > R$ is at most $1/R+\epsilon$ with probability $\exp(-\epsilon^2|E|/3)$ over the choice of the random maps. By a union bound and the condition on ϵ, the second term is at most $1/R + \epsilon$ for all labelings A, with high probability over the choice of $\{\sigma'_{1,e}\}_{e\in E}$. Picking an instance with appropriate choice of maps $\sigma'_{1,e}$ gives the required instance $\mathcal{L}'$. $\square$

4.5 Discussion

The instances we construct have SDP value 1 only for the most basic semidefinite programming relaxation. It would be desirable to get gaps for stronger SDPs, beginning with the most modest extensions of this basic SDP. For example, in the SDP for 2-to-1 Label Cover from Figure 4.2, we can add valid nonnegativity constraints for the dot product between every pair of vectors in the set

$$\{\mathbf{y}_{(u,i)} \mid u \in U, i \in [R_1]\} \bigcup \{\mathbf{z}_{(v,j)} \mid v \in V, j \in [R_2]\}$$

since in the integral solution all these vectors are $\{0, 1\}$-valued. The vectors we construct do *not* obey such a nonnegativity requirement. For the case of Unique Games, Khot and Vishnoi [KV05] were able to ensure nonnegativity of all dot products by simply taking tensor products of the vectors with themselves and defining new vectors $\mathbf{y}'_{(u,i)} = \mathbf{y}^{\otimes 2}_{(u,i)} = \mathbf{y}_{(u,i)} \otimes \mathbf{y}_{(u,i)}$ and $\mathbf{z}'_{(v,j)} = \mathbf{z}^{\otimes 2}_{(v,j)} = \mathbf{z}_{(v,j)} \otimes \mathbf{z}_{(v,j)}$. Since $\langle \mathbf{a}^{\otimes 2}, \mathbf{b}^{\otimes 2}\rangle = \langle \mathbf{a}, \mathbf{b}\rangle^2$, the desired nonnegativity of dot products is ensured.

We cannot apply this tensoring idea in our construction as it does not preserve the SDP value at 1. For example, for 2-to-1 Label Cover, if we have $\mathbf{y}_{(u,i)} = \mathbf{z}_{(v,j_1)} + \mathbf{z}_{(v,j_2)}$ (so that these vectors contribute 1 to the objective value to the

SDP of Figure 4.2), then upon tensoring we no longer necessarily have $\mathbf{y}_{(u,i)}^{\otimes 2} = \mathbf{z}_{(v,j_1)}^{\otimes 2} + \mathbf{z}_{(v,j_2)}^{\otimes 2}$. Extending our gap instances to obey the nonnegative dot product constraints is therefore a natural question that we leave open. While this seems already quite challenging, one can of course be more ambitious and ask for gap instances for stronger SDPs that correspond to certain number of rounds of some hierarchy, such as the Sherali-Adams hierarchy together with consistency of vector dot products with pairwise marginals. As mentioned in Section 1.3.1, for UNIQUE GAMES such gap instances were constructed for several rounds of such a hierarchy in [RS09, KS09].

Chapter 5

Approximate Lasserre integrality gap for Unique Label Cover

In this chapter we state and prove our result about an approximate Lasserre integrality gap for Unique Label Cover. In Section 5.1 we describe the Lasserre hierarchy and our results in more detail. In Section 5.2 we provide a high level overview of our construction. The rest of the chapter is organized as described in Section 5.2.3.

5.1 Lasserre hierarchy of SDP Relaxations

For a CSP such as Unique Games on n vertices with a label set $[k]$, a t-round Lasserre SDP relaxation introduces vectors $\mathbf{x}_{S,\sigma}$ for every subset S of vertices of size at most t and every assignment $\sigma : S \mapsto [k]$ of labels to the vertices in S. The intention is that in an integral solution, $\mathbf{x}_{S,\sigma} = 1$ if σ is restriction of the global assignment and $\mathbf{x}_{S,\sigma} = 0$ otherwise. Therefore, for a fixed set S, one adds the SDP constraint that the vectors $\{\mathbf{x}_{S,\sigma}\}_\sigma$ are orthogonal and the sum of their squared Euclidean norms is 1. One may interpret the squared Euclidean norms of these vectors as a probability distribution over assignments to S (in an integral solution the distribution is concentrated on a single assignment). Natural consistency constraints satisfied by an integral solution are added as well. Specifically, for two sets $T \subseteq S$, each of size at most t, and every assignment τ to T, the following natural constaint is added:

$$\sum_{\sigma:S\mapsto[k],\sigma|_T=\tau} \mathbf{x}_{S,\sigma} = \mathbf{x}_{T,\tau}, \tag{5.1}$$

where $\sigma|_T$ denotes the restriction of σ to subset T. Note that in an integral solution, both sides of the above equation are 1 if τ is restriction of the global assignment to T and zero otherwise. The objective value of the relaxation can be written in terms of pairwise inner products of vectors on singleton sets. The t-round Lasserre SDP relaxation entails adding $O(n^t)$ constraints in the SDP relaxation.

We will be interested in *approximate* solutions to the Lasserre hierarchy. Towards this end, we call a vector solution δ-approximate if Equation (5.1) is satisfied with error δ, i.e.

$$\left\| \sum_{\sigma:\sigma|_T=\tau} \mathbf{x}_{S,\sigma} - \mathbf{x}_{T,\tau} \right\| \le \delta. \tag{5.2}$$

We now state informally the main result of this chapter.

Theorem 5.1.1. *(Informal) Let $\epsilon > 0$ and $k, t \in \mathbb{Z}^+$ be arbitrary constants. Then for every constant $\delta > 0$, there is an instance $\mathcal{U}$ of* Unique Games *with label set $[k]$ that satisfies:*

1. *There exist vectors $\mathbf{x}_{S,\sigma}$ for every set S of vertices of $\mathcal{U}$ of size at most t, and every assignment of labels σ to the vertices in S such that it is a δ-*

approximate solution to the SDP relaxation with t-round Lasserre hierarchy.

2. *The SDP objective value of the above approximate vector solution is at least* $1 - \epsilon$.

3. *Any labeling to the vertices of $\mathcal{U}$ satisfies at most $k^{-\epsilon/2}$ fraction of edges.*

5.2 Overview of Our Construction

Our construction relies in large part on the work of Khot and Vishnoi [KV05] who gave SDP integrality gap examples for Unique Games and cut-problems including Maximum Cut. We also borrow ideas from [KS09] and [RS09] who build upon the work of [KV05] to obtain stronger integrality gap results as mentioned earlier.

Our strategy is to first construct approximate Lasserre vectors for the Unique Games instance $\mathcal{U}$ presented in [KV05]. This construction is not good enough by itself as the number of labels $[N]$ is too large relative to the accuracy parameter. We therefore apply the reduction of [KKMO07] to the instance $\mathcal{U}$ to obtain a new instance $\tilde{\mathcal{U}}$ of Unique Games with a much smaller label set $[k]$. This reduction preserves the low integral optimum, transforms the vectors corresponding to the instance $\mathcal{U}$ into corresponding vectors for the instance $\tilde{\mathcal{U}}$, and preserves the high SDP objective. These new vectors constitute the final δ-approximate Lasserre solution to $\tilde{\mathcal{U}}$. Below we describe the construction of Lasserre vectors for the instance $\mathcal{U}$. In the actual construction we present, we do no explicitly construct these vectors, but rather directly construct the instance $\tilde{\mathcal{U}}$ along with its approximate Lasserre solution. However, the description of the implicit intermediate step does illustrate the main ideas involved.

5.2.1 Lasserre Vectors for [KV05] Unique Games instance

We start with the Unique Games instance $\mathcal{U}$ along with a basic SDP solution constructed in [KV05]. Let $G(V, E)$ be its constraint graph and $[N]$ be the label set. The SDP solution consists of (up to a normalization) an orthonormal tuple $\{\mathbf{T}_{u,j}\}_{j\in[N]}$ for every vertex $u \in V$. A useful property of this solution is that the sum of vectors in every tuple is the same, i.e. for some fixed unit vector $\mathbf{T}$,

$$\mathbf{T} = \frac{1}{\sqrt{N}} \sum_{j\in[N]} \mathbf{T}_{u,j} \quad \forall u \in V. \tag{5.3}$$

As observed in [KS09], one can define a single vector $\mathbf{T}_u := \frac{1}{\sqrt{N}} \sum_{j\in[N]} \mathbf{T}_{u,j}^{\otimes 4}$ for each tuple $\{\mathbf{T}_{u,j}\}$ such that the distance $\|\mathbf{T}_u - \mathbf{T}_v\|$ captures the closeness between the pair of tuples $\{\mathbf{T}_{u,j}\}$ and $\{\mathbf{T}_{v,j}\}$. Roughly speaking, the edge (i.e. constraint) set E corresponds to all pairs (u, v) such that $\|\mathbf{T}_u - \mathbf{T}_v\| \le \gamma$ for a sufficiently small $\gamma > 0$. For any such edge, it necessarily holds that $\forall j \in [N]$, $\|\mathbf{T}_{u,j} - \mathbf{T}_{v,\pi(j)}\| \le O(\gamma)$ for some bijection $\pi = \pi(u, v) : [N] \mapsto [N]$. This is precisely the bijection defining the Unique Games constraint on edge (u, v) and also ensures that the SDP objective is high, i.e. $1 - O(\gamma^2)$.

Another key observation is that in the graph $G(V, E)$, any *local* neighborhood can be given a consistent labeling; in fact, once an arbitrary label for a vertex is fixed, it uniquely determines labels to all other vertices in a local neighborhood. Specifically, fix a small positive constant $p \le 0.1$. A set $C \subseteq V$ is called *p-local* if $\|\mathbf{T}_u - \mathbf{T}_v\| \le p \;\forall\, u, v \in C$. As observed in [KS09], for any p-local set C, there is a set $L(C)$ of N labelings, such that each labeling $\tau \in L(C)$ satisfies all the induced edges inside C. The j^{th} labeling is obtained by fixing the label of one vertex in C to be $j \in [N]$ and then uniquely fixing labels to all other vertices in C. This gives a natural way to define Lasserre vectors for all subsets $S \subseteq C$. Fix an arbitrary vertex $w \in C$. Consider any subset $S \subseteq C$, and a labeling σ to the vertices in S. We wish to construct a vector $\mathbf{y}_{S,\sigma}$. If σ is not consistent with any of the N labelings $\tau \in L(C)$ then set $\mathbf{y}_{S,\sigma} = 0$. Otherwise, let $\mathbf{y}_{S,\sigma} = \frac{1}{\sqrt{N}}\mathbf{T}_{w,j}$ where the labeling σ is consistent with a labeling $\tau \in L(C)$ which assigns j to w. It can be seen that this is a valid Lasserre SDP solution for all subsets of C. All edges that are inside C contribute well (i.e. $1 - O(\gamma^2)$) towards the SDP objective.

We now try to extend the above strategy to the whole set V. Even though the following naive approach does not work, it helps illustrate the main idea behind the construction. We partition V into local sets and construct Lasserre vectors that are a tensor product of vectors constructed for each local set. Towards this end, we think of the set of vectors $\{\mathbf{T}_u\}_{u \in V}$ as embedded on the unit sphere $\mathbb{S}^{|V|-1}$. Partition the unit sphere into clusters of diameter at most p. This naturally partitions the set of vertices V into disjoint p-local subsets $C_1, \ldots, C_m$. As before, fix w_i to be any arbitrary vertex in C_i for $i = 1, \ldots, m$. Now consider a subset $S \subseteq V$, and a labeling σ to the vertices in S, for which we wish to construct a vector $\mathbf{x}_{S,\sigma}$. Suppose that there is a subset C_i such that $\sigma|_{S \cap C_i}$ is not consistent with any labeling in $L(C_i)$; in this case set $\mathbf{x}_{S,\sigma} = 0$. Otherwise, construct vector $\mathbf{y}^i_{S,\sigma}$ as follows: if $|S \cap C_i| = \emptyset$, then let $\mathbf{y}^i_{S,\sigma} = \mathbf{T}$; else set $\mathbf{y}^i_{S,\sigma} = \frac{1}{\sqrt{N}}\mathbf{T}_{w_i,j}$, where $\sigma|_{S \cap C_i}$ is consistent with a labeling in $L(C_i)$ that assigns label j to w_i. Finally, let $\mathbf{x}_{S,\sigma} := \bigotimes_{i=1}^{m} \mathbf{y}^i_{S,\sigma}$. It can be seen that this construction is a valid SDP Lasserre solution. The tensor product is a vector analogue of assigning labeling to different clusters independently.

However, the above construction does not work because the unit sphere has dimension $|V|-1$ and partitioning such a high-dimensional sphere into local clusters necessarily means that almost all edges of $G(V, E)$ will have two endpoints in different clusters, and therefore the two endpoints get labels independently. This results in a very low SDP objective. A natural approach is to use dimensionality reduction that w.h.p. preserves the geometry of any set points that is not too large.

We therefore first randomly project the vectors $\{\mathbf{T}_u\}_{u \in V}$ onto $\mathbb{S}^{d-1}$ for an appropriate constant d. The Johnson-Lindenstrauss lemma implies that for a set $S \subseteq V$ of at most t vertices, w.h.p. the mapping approximately preserves all pairwise distances between the vectors $\{\mathbf{T}_u\}_{u \in S}$. This is followed, as before, by a (randomized) partition of $\mathbb{S}^{d-1}$ into low-diameter clusters that induces a partition of V into subsets $C_1, \ldots, C_m$. The dimension d is low enough to ensure that most of the edges in E fall inside some cluster. However, since the projection fails to preserve distances with some non-zero probability, the subsets C_i $(1 \leq i \leq m)$ are *not* guaranteed to be p-local. Nevertheless, for any set S of at most t vertices, if the projection preserves all distances between vectors $\{\mathbf{T}_u\}_{u \in S}$, then each of the sets $S \cap C_i$ for $i = 1, \ldots, m$ is a p-local set. For a fixed projection and a partition, a vector $\mathbf{x}_{S,\sigma}$ for the set S and its labeling σ can then be constructed as described earlier, except that there is no fixed representative vertex w_i for each C_i. Instead, an arbitrary vertex is chosen from the set $S \cap C_i$ to serve as the representative vertex w_i, and the set of labelings $L(S \cap C_i)$ is used. Since the projection and the partitioning are randomized, we implement the construction for each choice of random string and let the final vectors to be a (weighted) direct sum of the vectors constructed for each random string.

The above approach yields Lasserre vectors which have a good SDP objective value but only approximately satisfy the Lasserre constraints. There are two sources of error. One is that the random projection preserves distances within a set $S, |S| \leq t$, w.h.p. but not with probability 1. Secondly, since an arbitrary vertex from $S \cap C_i$ is chosen as a representative, for $T \subseteq S$, the representative for $S \cap C_i$ need not coincide with the representative for $T \cap C_i$. Still, since $S \cap C_i$ and $T \cap C_i$ are local sets (provided that the random projection has succeeded in preserving distances in S), their representative vectors are close enough.

5.2.2 Obtaining a δ-approximate Lasserre solution

As stated earlier, once we have the SDP vectors to the instance of [KV05], we apply the reduction of [KKMO07] and obtain a new instance of Unique Games with a constant label set $[k]$. We also obtain vectors which constitute the δ-approximate Lasserre solution to the new instance of Unique Games. We ensure that the objective value of the vectors remains high.

5.2.3 Organization

In Section 5.3 we formally define a formulation of the Lasserre hierarchy for Unique Games. In Section 5.4 we describe the basic Unique Games instance from [KV05] along with the reduction from [KKMO07] to obtain a new Unique Games instance with a constant label set $[k]$. In Section 5.3.3, we formally state our main theorem with quantitative parameters. Finally, in Section 5.5 we construct Lasserre vectors for the new Unique Games instance and prove that they form a δ-approximate Lasserre solution.

In Section 5.8 we define another formulation of the Lasserre hierarchy which is more standard in the literature, and prove that it is essentially equivalent to the formulation we use.

5.3 Preliminaries

5.3.1 Basic SDP relaxation

Let $\mathcal{U}(G(V,E),[k],\{\pi_e\}_{e\in E})$ be an instance of Unique Games (Definition 1.3.3). Note that we don't restrict the graph G to be bipartite. Figure 5.1 gives a natural SDP relaxation `SDP-UG`. The relaxation is over the vector variables $\mathbf{x}_{u,i}$ for every vertex u of the graph G and label $i \in [k]$.

$$\max \sum_{e=(u,v)\in E} \sum_{i\in[k]} \left\langle \mathbf{x}_{u,i}, \mathbf{x}_{v,\pi_e^{uv}(i)} \right\rangle \; wt(e)$$

Subject to,

$$\forall u \in V \qquad \sum_{i\in[k]} \|\mathbf{x}_{u,i}\|^2 = 1 \qquad \text{(I)}$$

$$\forall u \in V,\ i,j \in [k],\ i \neq j \qquad \langle \mathbf{x}_{u,i}, \mathbf{x}_{u,j} \rangle = 0 \qquad \text{(II)}$$

$$\forall u,v \in V,\ i,j \in [k] \qquad \langle \mathbf{x}_{u,i}, \mathbf{x}_{v,j} \rangle \geq 0 \qquad \text{(III)}$$

Figure 5.1: Relaxation `SDP-UG` for Unique Games

5.3.2 Lasserre relaxation

One can write a natural integer quadratic program for solving Unique Games, where the set of variables is $\mathbf{x}_{S,\sigma}$ for every $S \subseteq V$ and every assignment $\sigma : S \mapsto [k]$ to vertices in S. The solution to this quadratic program would ensure $\mathbf{x}_{S,\sigma} = 1$ if the global labeling of V induces the assignment σ on S and $\mathbf{x}_{S,\sigma} = 0$ otherwise.

The Lasserre semi-definite relaxation of Unique Games `L'-UG`(t) in Figure

5.3 (Section 5.8) is obtained by relaxing the variables of this quadratic program to vectors instead of integers and replacing the multiplication of two numbers by dot products of the corresponding vectors. In the t-round Lasserre relaxation, we consider sets of size up to t. Notice that SDP-UG is contained in L'-UG(2). In this chapter, we work with another relaxation L-UG(t) in Figure 5.2 which is essentially equivalent to L'-UG(t), but rephrases the constraints in terms of vector sums instead of dot-products. The two relaxations have the exact same objective function. In Lemma 5.8.1, we show that the two relaxations are essentially equivalent.

We say $\sigma|_T$ to mean assignment σ restricted to set T. We say $(S, \sigma) \simeq (S', \sigma')$ to mean that the assignments σ and σ' are consistent i.e. $\sigma|_{S\cap S'} = \sigma'|_{S\cap S'}$. Otherwise, we say $(S, \sigma) \not\simeq (S', \sigma')$. Let $\mathbf{x}_{u,i} := \mathbf{x}_{S,\sigma}$ for $S = \{u\}$ and $\sigma(u) = i$.

$$\max \sum_{e=(u,v)\in E} \sum_{i\in[k]} \left\langle \mathbf{x}_{u,i}, \mathbf{x}_{v,\pi_e^{uv}(i)} \right\rangle \ wt(e)$$

Subject to,

$$\|\mathbf{x}_\phi\|^2 = 1 \qquad \text{(IV)}$$

$$\forall\ S, |S| \le t, \sigma \neq \sigma' \qquad \langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S,\sigma'} \rangle = 0 \qquad \text{(V)}$$

$$\forall\ T \subseteq S, \tau \in [k]^T \qquad \sum_{\sigma:\sigma|_T=\tau} \mathbf{x}_{S,\sigma} = \mathbf{x}_{T,\tau} \qquad \text{(VI)}$$

Figure 5.2: Relaxation L-UG(t) for Unique Games

Thus, we want to construct $k^{|S|}$ orthogonal vectors for each set S of size up to t, such that the vectors for different sets are consistent with each other in the sense of Equation (VI).

5.3.3 Main Theorem

Theorem 5.3.1. *Fix an arbitrarily small constant $\epsilon > 0$ and integer $k \in \mathbb{Z}^+$. Then for all sufficiently large N (that is a power of 2), there is an instance $\mathcal{U}$ of* Unique Games *on $\frac{2^N}{N} \cdot k^{N-1}$ vertices with label set $[k]$ such that,*

1. *There exist vectors $\mathbf{x}_{S,\sigma}$ for every set S of vertices of $\mathcal{U}$ of size at most t, and every assignment of labels $\sigma : S \mapsto [k]$ such that it is a $O(t \cdot \eta^{1/16})$-approximate solution for $\eta := (\log N)^{-0.99}$ to the SDP relaxation with t-round Lasserre hierarchy of constraints.*

2. *The SDP objective value of the above approximate vector solution is at least $1 - O(\epsilon)$.*

3. *Any labeling to the vertices of $\mathcal{U}$ satisfies at most $k^{-\epsilon/2}$ fraction of edges.*

Proof. The construction is presented in Section 5.5 and properties (1), (2) and (3) are proved in Lemmas 5.6.2, 5.6.3, 5.4.4 respectively.

□

5.4 The instance

5.4.1 Basic instance

The starting point of our reduction is a Unique Games integrality gap instance $\mathcal{U}_\eta$ for `SDP-UG` constructed in [KV05]. Our presentation of the Unique Games instance $\mathcal{U}_\eta$ follows that in [KS09].

For $\eta > 0$ and $N = 2^m$ for some $m \in \mathbb{Z}^+$, Khot and Vishnoi [KV05] construct the Unique Games instance $\mathcal{U}_\eta(G'(V', E'), [N], \{\pi_e\}_{e \in E})$ where the number of vertices $|V'| = 2^N/N$. The instance has no good labeling, i.e. has low optimum.

lemma 5.4.1. *Any labeling to the vertices of the* Unique Games *instance* $\mathcal{U}_\eta(G'(V', E'), [N], \{\pi_e\}_{e\in E})$ *satisfies at most* $\frac{1}{N^\eta}$ *fraction of the edges.*

In the construction of [KV05] the elements of $[N]$ are identified with the additive group $(\mathbb{F}_2^m, \oplus)$. The authors construct a vector solution that consists of unit vectors $\mathbf{T}_{u,i}$ for every vertex $u \in V'$ and label $i \in [N]$. These vectors (up to a normalization) form the solution to the Unique Games SDP relxation `SDP-UG`. We highlight the important properties of the SDP solution below:

Properties of the Unique Games SDP Solution

- **(Orthonormal basis)** $\forall\ u \in V', \ \ \forall\ i \neq j \in [N]$,

$$\|\mathbf{T}_{u,i}\| = 1, \quad \langle \mathbf{T}_{u,i}, \mathbf{T}_{u,j}\rangle = 0. \tag{5.4}$$

- **(Non-negativity)** $\forall\ u, v \in V', \ \ \forall\ i, j \in [N]$,

$$\langle \mathbf{T}_{u,i}, \mathbf{T}_{v,j}\rangle \geq 0. \tag{5.5}$$

- **(Symmetry)** $\forall\ u, v \in V', \ \ \forall\ i, j, s \in [N]$,

$$\langle \mathbf{T}_{u,i}, \mathbf{T}_{v,j}\rangle = \langle \mathbf{T}_{u,s\oplus i}, \mathbf{T}_{v,s\oplus j}\rangle \tag{5.6}$$

  where '$\oplus$' is the group operation on $[N]$ as described above.

- **(High SDP Value)** For every edge $e = (u, v) \in E'$,

$$\forall\ i \in [N], \quad \left\langle \mathbf{T}_{u,i}, \mathbf{T}_{v,\pi_e^{uv}(i)}\right\rangle \geq 1 - 4\eta. \tag{5.7}$$

In fact, there is $s_e^{uv} \in [N]$ such that $\forall\ i \in [N],\ \ \pi_e^{uv}(i) = s_e^{uv} \oplus i$.

- **(Sum to a Constant Vector)** For every vertex $u \in V'$,

$$\frac{1}{\sqrt{N}} \sum_{i=1}^{N} \mathbf{T}_{u,i} = \mathbf{T} \tag{5.8}$$

where $\mathbf{T}$ is a fixed unit vector.

- **(Local Consistency)** A set $W \subseteq V'$ of vertices is p-local if $||\mathbf{T}_u - \mathbf{T}_v|| \leq p \leq 0.1$ for all $u, v \in W$. Here, $\mathbf{T}_u := \frac{1}{\sqrt{N}} \sum_{j\in[N]} .\mathbf{T}_{u,j}^{\otimes 4}$

lemma 5.4.2 ([KS09])**.** *Suppose a set $W \subseteq V'$ is p-local. Then there is set $L(W)$ of N* locally consistent *assignments to vertices in W such that if $\mu : W \mapsto [N] \in L(W)$ then*

$$\forall u, v \in W :\ \left\langle \mathbf{T}_{u,\mu(u)}, \mathbf{T}_{v,\mu(v)} \right\rangle \geq 1 - O(p^2). \tag{5.9}$$

The assignments in $L(W)$ are disjoint i.e. if $\mu \neq \mu' \in L(W)$ then $\forall\ u \in W, \mu(u) \neq \mu'(u)$.

The authors in [KS09] define for every vertex $u \in V'$ a unit vector $\mathbf{T}_u$

$$\mathbf{T}_u := \frac{1}{\sqrt{N}} \sum_{i\in[N]} \mathbf{T}_{u,i}^{\otimes 4}. \tag{5.10}$$

and prove that that the Euclidean distances between the vectors $\{\mathbf{T}_u\}_{u\in V'}$ are a measure of the 'closeness' between the orthonormal tuples $\{\mathbf{T}_{u,i} \mid i \in [N]\}_{u\in V'}$.

lemma 5.4.3 ([KS09]). *For every $u, v \in V'$,*

$$\min_{i,j \in [N]} \|\mathbf{T}_{u,i} - \mathbf{T}_{v,j}\| \;\leq\; \|\mathbf{T}_u - \mathbf{T}_v\| \;\leq\; 2 \cdot \min_{i,j \in [N]} \|\mathbf{T}_{u,i} - \mathbf{T}_{v,j}\| \tag{5.11}$$

5.4.2 Reduction to constant label size

In this section we transform the instance $\mathcal{U}_\eta(G'(V', E'), [N], \{\pi_e\}_{e \in E'})$ described in the previous section to another UNIQUE GAMES instance $\mathcal{U}_\epsilon(G(V, E), [k], \{\pi_e\}_{e \in E})$ using a reduction presented in [KKMO07]. Here $[k]$ is to be thought of as the set $\{0, 1, \ldots, k-1\}$ with the group operation of addition modulo k.

We start with the UNIQUE GAMES instance $\mathcal{U}_\eta(G'(V', E'), [N], \{\pi_e\}_{e \in E'})$ and replace each vertex $v \in V'$ by a block of k^{N-1} vertices $(v, \mathbf{s})$ where $\mathbf{s} \in [k]^N$ and $\mathbf{s}_1 = 0$.

For every pair of edges $e = (v, w), e' = (v, w') \in E'$, there are (all possible) weighted edges between the blocks $(w, \cdot)$ and $(w', \cdot)$ in the instance $\mathcal{U}_\epsilon(G(V, E), [k], \{\pi_e\}_{e \in E})$. The edge between $a := (w, \mathbf{s})$ and $b := (w', \mathbf{s}')$ is constructed as follows:-

1. Pick $\mathbf{p}$ uniformly at random from $[k]^N$ and $\mathbf{p}' \in [k]^N$ such that each coordinate ${\mathbf{p}'}_i$ is chosen to be $\mathbf{p}_i$ with probability $1 - \epsilon$ and is chosen uniformly at random from $[k]$ with probability ϵ for all $i \in [N]$.

2. Define $\mathbf{q}, \mathbf{q}' \in [k]^N$ as $\mathbf{q} := \mathbf{p} \circ \pi_e^{wv}$, $\mathbf{q}' := \mathbf{p}' \circ \pi_{e'}^{w'v}$ where $\mathbf{p} \circ \pi := (\mathbf{p}_{\pi(1)}, \ldots, \mathbf{p}_{\pi(N)})$.

3. Define $\mathbf{r}, \mathbf{r}' \in [k]^N$ as $\mathbf{r}_i := \mathbf{q}_i - \mathbf{q}_1$ and ${\mathbf{r}'}_i := {\mathbf{q}'}_i - {\mathbf{q}'}_1$ for all i from 1 through N.

4. Add an edge e^* between $a = (w, \mathbf{s})$ and $b = (w', \mathbf{s}')$ such that $\pi_{e^*}^{ab}(i) := (i + \mathbf{q'}_1 - \mathbf{q}_1)$ for all $i \in [k]$ and $wt(e^*) := \Pr[\mathbf{s} = \mathbf{r}, \mathbf{s}' = \mathbf{r}']$.

The third step in the construction incorporates a PCP trick called folding. To prove that the instance constructed has low optimum, we need the property that any labelling to vertices in $\mathcal{U}_\epsilon$ is balanced on every block of vertices arising out of some vertex in $\mathcal{U}_\eta$ i.e. it assigns each label in every block equally often.

We achieve this by reducing the number of vertices in each block by a factor of $\frac{1}{k}$, and then extend any labelling on the reduced vertex set to a balanced labelling on the original vertex set. In our case, we only consider strings $\mathbf{s}$ with $\mathbf{s}_1 = 0$ and as a mental exercise we extend any labeling σ to all strings as

$$\sigma(\mathbf{s'}_1, \mathbf{s'}_2, \ldots, \mathbf{s'}_N) := \sigma(0, \mathbf{s'}_2 - \mathbf{s'}_1, \ldots, \mathbf{s'}_N - \mathbf{s'}_1) + \mathbf{s'}_1$$

The following is a reformulation of Theorem 12 and corollary 13 of [KKMO07].

lemma 5.4.4. *Any labeling to the vertices of the* Unique Games *instance* $\mathcal{U}_\epsilon(G(V,E), [k], \{\pi_e\}_{e \in E})$ *satisfies at most* $k^{-\epsilon/2}$ *fraction of the edges provided the optimum of the instance* $\mathcal{U}_\eta$ *(which is at most* $N^{-\eta}$*)) is sufficiently small as a function of* ϵ *and* k*.*

5.5 Approximate Vector Construction

In this section we construct Lasserre vectors for the Unique Games instance $\mathcal{U}_\epsilon(G(V,E), [k], \{\pi_e\}_{e \in E})$ described in the previous section. Our construction will be randomized, i.e. we first create vectors $\mathbf{y}^r_{S,\sigma}$ for every choice of random bits r

and then set

$$\mathbf{x}_{S,\sigma} := \bigoplus_r \sqrt{\Pr[r]}\ \mathbf{y}^r_{S,\sigma} \tag{5.12}$$

where $\Pr[r]$ is the probability of choosing the random bit-sequence r (vectors for different choices of randomness live in independent, mutually orthogonal spaces).

Our construction will use Theorems 5.7.1 and 5.7.3 along with corollary 5.7.2 which are stated in Section 5.7.

5.5.1 Construction

We intend to construct vectors $\mathbf{x}_{S,\sigma}$ for every set $S \subseteq V$, $|S| \leq t$, and every assignment $\sigma : S \mapsto [k]$. Set $p = \eta^{1/16}$ and $d = 8\ln(2t^2/\eta)/p^2$.

1. **Projection:**

   Use corollary 5.7.2 to obtain a mapping $\mathbf{T}_u \mapsto \mathbf{T}'_u \in \mathbb{S}^{d-1} \;\; \forall\, u \in V'$.

2. **Partition:**

   Use Theorem 5.7.3 to randomly partition $\mathbb{S}^{d-1}$ with diameter p. Let C_1, C_2, $\ldots$, C_m denote this partition of $\mathbb{S}^{d-1}$ as well as the induced partition of V' (by a slight abuse of notation).

3. **Constructing vectors for a fixed set** $S \subseteq V$, $|S| \leq t$:

   Recall that every vertex of S is of the form $a = (v, \mathbf{s})$ for some $v \in V'$ and $\mathbf{s} \in [k]^N, \mathbf{s}_1 = 0$. Let $S = \cup_{l=1}^m S_\ell$ be a partition of S such that

   $$S_\ell := \{a = (v, \mathbf{s}) \in S \mid v \in C_\ell\}.$$

Also define for the sake of notational ease,

$$S'_\ell := \{v \mid \exists\ a = (v, \mathbf{s}) \in S_\ell\} \subseteq C_\ell \qquad \text{and} \qquad S' := \cup_{\ell=1}^m S'_\ell.$$

Since $|S| \leq t$, at most t of the sets S_ℓ (and hence S'_ℓ) are non-empty. Let r be the randomness used in Steps (1) and (2). If the Projection succeeds for the entire set S' (see corollary 5.7.2), go to Step 4.

Otherwise set $\mathbf{y}^r_{S,\sigma} := 0$ for all $\sigma : S \mapsto [k]$ and go to Step 5.

4. Since $S = \cup_{\ell=1}^m S_\ell$ is a partition, an assignment $\sigma : S \mapsto [k]$ can be split into assignments $\sigma_\ell : S_\ell \mapsto [k]$ for $\ell = 1, \ldots, m$. The construction below is the vector analogue of choosing an assignment σ_ℓ for set S_ℓ from a certain distribution, but independently for all $\ell = 1, \ldots, m$.

   For each ℓ such that $S_\ell = \emptyset$, let $\mathbf{y}^{r,l}_{S_\ell,\sigma_\ell} := \mathbf{T}$.

   For each ℓ such that $S_\ell \neq \emptyset$, observe that the set S'_ℓ is $O(p)$-local since the projection succeeded for S' and since the diameter of C_ℓ is at most p. Let $L(S'_\ell)$ denote the set of N locally consistent assignments to S'_ℓ as in Lemma 5.4.2, Equation (5.9).

   We partition the set $L(S'_\ell)$ of locally consistent assignments into different classes depending on how they behave w.r.t. assignments $\sigma_\ell : S_\ell \mapsto [k]$. Towards this end, let

$$L^{r,\ell}_{S_\ell,\sigma_\ell} := \left\{\mu \mid \mu \in L(S'_\ell) \text{ such that } \forall\ a = (v, \mathbf{s}) \in S_\ell,\ \ \mathbf{s}_{\mu(v)} = \sigma_\ell(a)\right\}.$$

Now arbitrarily pick a *representative* element $u \in S'_\ell$ and set

$$\mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} := \frac{1}{\sqrt{N}} \sum_{\mu \in L^{r,\ell}_{S_\ell,\sigma_\ell}} \mathbf{T}_{u,\mu(u)}.$$

Finally define,

$$\mathbf{y}^{r}_{S,\sigma} := \bigotimes_{\ell=1}^{m} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} \tag{5.13}$$

5. Construct vectors $\mathbf{x}_{S,\sigma} := \bigoplus_{r} \sqrt{\Pr[r]}\ \mathbf{y}^{r}_{S,\sigma}$ as in Equation (5.12).

5.6 Analysis

We begin with the following lemma.

lemma 5.6.1. *In the Step (4) of the construction in Section 5.5.1, for any fixed r and ℓ,*

$$\sum_{\sigma_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} = \mathbf{T}.$$

Proof.

$$\sum_{\sigma_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} = \frac{1}{\sqrt{N}} \sum_{\sigma_\ell} \sum_{\mu \in L^{r,\ell}_{S_\ell,\sigma_\ell}} \mathbf{T}_{u,\mu(u)} = \frac{1}{\sqrt{N}} \sum_{\mu \in L(S'_\ell)} \mathbf{T}_{u,\mu(u)} = \mathbf{T},$$

from Equation (5.8). □

lemma 5.6.2. *The vectors $\mathbf{x}_{S,\sigma}$ constructed in the previous section satisfy the constraints of the SDP* L-UG(t) *up to the following errors:-*

1. *Equations (IV) and (V) are satisfied exactly*

2. Equation (VI) is satisfied up to an error of $O(tp)$, i.e. for any sets $T \subseteq S$, $|S| \le t$ and an assignement $\tau : T \mapsto [k]$,

$$\left\| \sum_{\sigma:\sigma|_T=\tau} \mathbf{x}_{S,\sigma} - \mathbf{x}_{T,\tau} \right\| \le O(tp). \tag{5.14}$$

Proof. **(1):** It is clear from the construction that

$$\mathbf{x}_\phi = \bigoplus_r \sqrt{\Pr[r]} \bigotimes_{l=1}^{m} \mathbf{T}$$

which is a unit vector since $\mathbf{T}$ is a unit vector. Hence, Equation (IV) is satisfied. Also, it is easy to check that for a fixed set S, for every choice of the randomness, we always assign orthogonal vectors for different assignments σ, hence Equation (V) is satisfied.

(2): We will show that with probability $(1-\eta)$ over the choice of randomness r,

$$\left\| \sum_{\sigma:\sigma|_T=\tau} \mathbf{y}^r_{S,\sigma} - \mathbf{y}^r_{T,\tau} \right\| \le O(tp). \tag{5.15}$$

Using Equation (5.12), this implies that the desired claim that

$$\left\| \sum_{\sigma:\sigma|_T=\tau} \mathbf{x}_{S,\sigma} - \mathbf{x}_{T,\tau} \right\|^2 \le O(\eta) + O(t^2p^2) = O(t^2p^2).$$

Now we prove Equation (5.15). The Projection in Step 1 of the construction succeeds for S' (and hence also for T') with probability at least $1-\eta$ by corollary 5.7.2. Now fix the randomness r such that the projection succeeded for S'.

Let $(S = \cup_{\ell=1}^m S_\ell,\ \sigma = \{\sigma_\ell\}_{\ell=1}^m,\ S' = \cup_{\ell=1}^m S'_\ell)$ and

$(T = \cup_{\ell=1}^m T_\ell,\ \tau = \{\tau_\ell\}_{\ell=1}^m,\ T' = \cup_{\ell=1}^m T'_\ell)$ be the splitting of sets and their assignments respectively as described in Steps 3 and 4 of the construction in Section 5.5.1. Note that

$$\mathbf{y}^r_{T,\tau} = \bigotimes_{l=1}^{m} \mathbf{y}^{r,\ell}_{T_\ell,\tau_\ell} \tag{5.16}$$

and

$$\sum_{\sigma|_T=\tau} \mathbf{y}^r_{S,\sigma} = \sum_{\sigma|_T=\tau} \bigotimes_{l=1}^{m} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} = \bigotimes_{l=1}^{m} \left(\sum_{\sigma_\ell|_{T_\ell}=\tau_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} \right) \tag{5.17}$$

In the tensor product on right hand side in above Equations (5.16, 5.17), all but at most t of the sets S_ℓ (and hence T_ℓ) are empty in which case $\sum_{\sigma_\ell|_{T_\ell}=\tau_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} = \mathbf{y}^{r,\ell}_{T_\ell,\tau_\ell} = \mathbf{T}$. Thus it suffices to prove that for all ℓ such that $S_\ell \neq \emptyset$, we have

$$\left\| \sum_{\sigma_\ell|_{T_\ell}=\tau_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} - \mathbf{y}^{r,\ell}_{T_\ell,\tau_\ell} \right\| \leq O(p).$$

If $T_\ell = \emptyset$, then $\mathbf{y}^{r,\ell}_{T_\ell,\tau_\ell} = \mathbf{T}$ and $\sum_{\sigma_\ell|_{T_\ell}=\tau_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} = \mathbf{T}$ as well by Lemma 5.6.1, so we are done. Assume therefore that $T_\ell \neq \emptyset$. In that case, for some representative vertices $u^* \in S'_\ell$ and $v^* \in T'_\ell$, we have

$$\mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} = \frac{1}{\sqrt{N}} \sum_{\mu \in L^{r,\ell}_{S_\ell,\sigma_\ell}} \mathbf{T}_{u^*,\mu(u^*)}, \qquad \mathbf{y}^{r,\ell}_{T_\ell,\tau_\ell} = \frac{1}{\sqrt{N}} \sum_{\nu \in L^{r,\ell}_{T_\ell,\tau_\ell}} \mathbf{T}_{v^*,\nu(v^*)}.$$

Since $T'_\ell \subseteq S'_\ell$ are both $O(p)$-local sets, every locally consistent assignment ν

to T'_ℓ has a unique extension to a locally consistent assignment to S'_ℓ. Also,

$$\begin{aligned}
\bigcup_{\sigma_\ell|_{T_\ell}=\tau_\ell} L^{r,\ell}_{S_\ell,\sigma_\ell} &= \bigcup_{\sigma_\ell|_{T_\ell}=\tau_\ell} \{\mu \mid \mu \in L(S'_\ell) \text{ such that } \forall\, a=(v,\mathbf{s}) \in S_\ell,\ \ \mathbf{s}_{\mu(v)} = \sigma_\ell(a)\} \\
&= \Big\{\mu \mid \mu \in L(S'_\ell) \text{ such that } \forall\, a=(v,\mathbf{s}) \in T_\ell,\ \ \mathbf{s}_{\mu(v)} = \tau_\ell(a)\Big\} \\
&= \Big\{\nu \mid \nu \in L(T'_\ell) \text{ such that } \forall\, a=(v,\mathbf{s}) \in T_\ell,\ \ \mathbf{s}_{\nu(v)} = \tau_\ell(a)\Big\} \\
&= L^{r,\ell}_{T_\ell,\tau_\ell}.
\end{aligned}$$

Hence,

$$\sum_{\sigma_\ell|_{T_\ell}=\tau_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} = \frac{1}{\sqrt{N}} \sum_{\mu\,\in\,\bigcup_{\sigma_\ell|_{T_\ell}=\tau_\ell} L^{r,\ell}_{S_\ell,\sigma_\ell}} \mathbf{T}_{u^*,\mu(u^*)} = \frac{1}{\sqrt{N}} \sum_{\nu \in L^{r,\ell}_{T_\ell,\tau_\ell}} \mathbf{T}_{u^*,\nu(v^*)}.$$

Writing $L = L^{r,\ell}_{T_\ell,\tau_\ell}$, $|L| \le N$, it follows that

$$\begin{aligned}
\left\| \sum_{\sigma_\ell|_{T_\ell}=\tau_\ell} \mathbf{y}^{r,\ell}_{S_\ell,\sigma_\ell} - \mathbf{y}^{r,\ell}_{T_\ell,\tau_\ell} \right\|^2 &= \frac{1}{N} \left\| \sum_{\nu\in L} \mathbf{T}_{u^*,\nu(u^*)} - \sum_{\nu\in L} \mathbf{T}_{v^*,\nu(v^*)} \right\|^2 \\
&\le \frac{2}{N} \sum_{\nu\in L} \big(1 - \big\langle \mathbf{T}_{u^*,\nu(u^*)}, \mathbf{T}_{v^*,\nu(v^*)} \big\rangle\big) \\
&\le O(p^2) \qquad \text{using Equations (5.9) and (5.5).}
\end{aligned}$$

□

lemma 5.6.3. *The objective value achieved by the vectors $\mathbf{x}_{S,\sigma}$ constructed in the previous section for the SDP* `L-UG`(t) *is at least* $(1 - O(\epsilon))|E|$ *where* $|E| := \sum_e wt(e)$.

Proof. Consider edges of the form $e^* = (a, b)$ where $a := (w, \mathbf{s})$ and $b := (w', \mathbf{s}')$ as described in Section 5.4.2. Observe that

$$\sum_{\mathbf{s},\mathbf{s}'} wt(a,b) = 1.$$

We will prove that

$$\sum_{\mathbf{s},\mathbf{s}'} \sum_{i\in[k]} \left\langle \mathbf{x}_{a,i}, \mathbf{x}_{b,\pi^{ab}_{e^*}(i)} \right\rangle \; wt(e^*) \geq 1 - O(\epsilon)$$

which suffices to prove the lemma.

Notice that $\|\mathbf{T}_w - \mathbf{T}_{w'}\| = O(\sqrt{\eta})$ by Equations (5.7) and (5.11). With probability at least $1-\eta$, the projection in Step 1 ensures (see corollary 5.7.2) that $\|\mathbf{T}'_w - \mathbf{T}'_{w'}\| = O(\sqrt{\eta})$. Hence, by Theorem 5.7.3, the probability that the partitioning in Step 2 puts $\mathbf{T}'_w$ and $\mathbf{T}'_{w'}$ in different clusters is at most $O(\sqrt{\eta}\cdot d/p)$ which is at most $O(\eta^{1/4})$ and our choice of parameters. Now fix a choice of randomness r such that w and w' lie in the same cluster, say the cluster C_ℓ. We will prove that

$$\sum_{\mathbf{s},\mathbf{s}'} \sum_{i\in[k]} \left\langle \mathbf{y}^r_{a,i}, \mathbf{y}^r_{b,\pi^{ab}_{e^*}(i)} \right\rangle \; wt(e^*) \geq 1 - O(\epsilon). \tag{5.18}$$

Using Equation (5.12), this implies

$$\sum_{\mathbf{s},\mathbf{s}'} \sum_{i\in[k]} \left\langle \mathbf{x}_{a,i}, \mathbf{x}_{b,\pi^{ab}_{e^*}(i)} \right\rangle \; wt(e^*) \geq (1-O(\epsilon))(1-O(\eta^{1/4})) \geq 1 - O(\epsilon)$$

by our choice of parameters.

It remains to prove Equation (5.18). Let

$$L_i := L^{r,\ell}_{a,i} = \{\nu \mid \nu \in L(\{w\}) \text{ and } \mathbf{s}_{\nu(w)} = i\}$$

and

$$L'_i := L^{r,\ell}_{b,\pi^{ab}_{e^*}(i)} = \{\nu' \mid \nu' \in L(\{w'\}) \text{ and } \mathbf{s}'_{\nu'(w')} = \pi^{ab}_{e^*}(i)\}.$$

Observe that the left hand side of the last equation is the same as

$$\frac{1}{N}\sum_{\mathbf{s},\mathbf{s}'}\sum_{i\in[k]}\sum_{\substack{\nu\in L_i\\ \nu'\in L'_i}} \left\langle \mathbf{T}_{w,\nu(w)}, \mathbf{T}_{w',\nu'(w')}\right\rangle \; wt(e^*).$$

We lower bound this expression by restricting the inner summation to only those pairs $(\nu,\nu') \in L_i \times L'_i$ for which there exists a (necessarily unique) assignement $\mu \in L(\{w,w'\})$ such that $\mu(w) = \nu(w)$ and $\mu(w') = \nu'(w')$. Note that since $\{w,w'\}$ is $O(\eta^{1/2})$-local, the set $L(\{w,w'\})$ of locally consistent assignments is well-defined. Thus a lower bound on the above expression is

$$\frac{1}{N}\sum_{\mathbf{s},\mathbf{s}'}\sum_{i\in[k]}\sum_{\substack{\mu\in L(\{w,w'\}),\\ \mu|_w\in L_i,\mu|_{w'}\in L'_i}} \left\langle \mathbf{T}_{w,\mu(w)}, \mathbf{T}_{w',\mu(w')}\right\rangle \; wt(e^*)$$

Noting that the inner product is at least $1 - O(\eta)$ (see Equation (5.9)), and using the definition of L_i and L'_i, we further lower bound the expression by

$$(1-O(\eta))\frac{1}{N}\sum_{\mathbf{s},\mathbf{s}'}\sum_{i\in[k]}\sum_{\mu\in L(\{w,w'\})} \mathrm{IND}\left[\mathbf{s}_{\mu(w)} = i, \mathbf{s}'_{\mu(w')} = \pi^{ab}_{e^*}(i)\right] \; wt(e^*)$$

where IND$[\cdot]$ is an indicator function. Let $\pi := \pi^{vw}_e$ and $\pi' := \pi^{vw'}_{e'}$. Then except the $(1-O(\eta))$ factor, the above expression is same as,

$$\begin{aligned}
&\frac{1}{N}\sum_{\mathbf{s},\mathbf{s}'}\sum_{\mu\in L(\{w,w'\})}\sum_{i\in[k]}\text{IND}\left[\mathbf{s}_{\mu(w)}=i,\mathbf{s}'_{\mu(w')}=\pi^{ab}_{e^*}(i)\right]wt(e^*)\\
=\ &\frac{1}{N}\sum_{\mathbf{s},\mathbf{s}'}\sum_{\mu\in L(\{w,w'\})}\text{IND}\left[\mathbf{s}_{\mu(w)}=\mathbf{s}'_{\mu(w')}+\mathbf{q}'_1-\mathbf{q}_1\right]wt(e^*)\\
=\ &\frac{1}{N}\sum_{\mu\in L(\{w,w'\})}\Pr_{\mathbf{p},\mathbf{p}'}[\mathbf{r}_{\mu(w)}=\mathbf{r}'_{\mu(w')}+\mathbf{q}'_1-\mathbf{q}_1]\\
=\ &\frac{1}{N}\sum_{\mu\in L(\{w,w'\})}\Pr_{\mathbf{p},\mathbf{p}'}[\mathbf{q}_{\mu(w)}=\mathbf{q}'_{\mu(w')}]=\frac{1}{N}\sum_{\mu\in L(\{w,w'\})}\Pr_{\mathbf{p},\mathbf{p}'}[\mathbf{q}_{\pi(\mu(v))}=\mathbf{q}'_{\pi'(\mu(v))}]\\
=\ &\frac{1}{N}\sum_{\mu\in L(\{w,w'\})}\Pr_{\mathbf{p},\mathbf{p}'}[\mathbf{p}_{\mu(v)}=\mathbf{p}'_{\mu(v)}]\geq 1-\epsilon
\end{aligned}$$

where the second last equality uses Equations (5.7, 5.9) and the last equality uses the definition of $\mathbf{p}$ and $\mathbf{p}'$.

□

5.7 Projecting and partitioning on a unit sphere

We state the Theorems 5.7.1 and 5.7.3, and prove Corollary 5.7.2 which are used in Section 5.5. Theorem 5.7.1 can be inferred from [[DG99], Lemma 2.2] while Theorem 5.7.3 can be inferred from [[GKL03], Theorem 3.2] applied to the Euclidean unit sphere.

Theorem 5.7.1 ([JL84],[DG99])**.** *Let each entry of an $d \times n$ matrix P be chosen independently from $N(0,1)$. Let $Q := \frac{1}{\sqrt{d}}P$ and $v = Qu$ for $u \in \mathbb{R}^n$. Then for any $0 \leq \theta \leq \frac{1}{2}$,*

$$(1-\theta)\|u\| \leq \|Qu\| \leq (1+\theta)\|u\| \tag{5.19}$$

with probability at least $1-2e^{-\theta^2 \frac{d}{8}}$. *We say that* $v \in \mathbb{R}^d$ *is the projection of* $u \in \mathbb{R}^n$.

Corollary 5.7.2. *There is a randomized mapping* $\Gamma : \mathbb{S}^{n-1} \mapsto \mathbb{S}^{d-1}$ *with* $d = 8\ln(2t^2/\eta)/p^2$, *such that for any set* $X \subseteq \mathbb{S}^{n-1}$, $|X| \leq t$, *with probability* $1-\eta$, *we have*

$$\forall x, y \in X, \quad \frac{1}{32}\|\Gamma(x) - \Gamma(y)\| \leq \|x-y\| \leq 4p + 2\ \|\Gamma(x) - \Gamma(y)\|.$$

If this conclusion holds, we say that the randomized mapping (projection) succeeded.

Proof. Let Q be the random matrix as in Theorem 5.7.1, $\theta = p$, and define $\Gamma(x) = \frac{Qx}{\|Qx\|}$. Then by a union bound, with probability $1-\eta$, Equation (5.19) holds for all $u \in X \cup \{x-y | x, y \in X\}$. In that case, for any $x, y \in X$, letting $a = \|Qx\|, b = \|Qy\|$, we see that $a, b \in [1-\theta, 1+\theta]$ and $|a-b| \leq \|Qx - Qy\| = \|Q(x-y)\| \leq (1+\theta)\cdot\|x-y\|$. Hence,

$$ab \cdot \|\Gamma(x) - \Gamma(y)\| = \|bQx - aQy\| \leq b\|Qx - Qy\| + |b-a|\|Qy\| \leq 2\cdot(1+\theta)^2\|x-y\|.$$

This proves the left inequality. For the right inequality, we have:

$$\begin{aligned}(1-\theta)\cdot\|x-y\| &\leq \|Q(x-y)\| \leq \|Qx - \Gamma(x)\| + \|\Gamma(x) - \Gamma(y)\| + \|\Gamma(y) - Qy\| \\ &= |\|Qx\| - 1| + \|\Gamma(x) - \Gamma(y)\| + |\|Qy\| - 1| \leq 2\theta + \|\Gamma(x) - \Gamma(y)\|.\end{aligned}$$

□

Theorem 5.7.3 ([GKL03])**.** *Let* $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : ||x|| = 1\}$ *denote the* $(d-1)$ *dimensional unit sphere. For every choice of diameter* $p > 0$ *there is a randomized partition* $\tilde{P}$ *of* $\mathbb{S}^{d-1}$ *into disjoint clusters such that,*

1. *For every cluster* $C \in \tilde{P}$, $C \subseteq \mathbb{S}^{d-1}$, $\mathtt{diam}(C) \leq p$.

2. *For any pair of points* $u, v \in \mathbb{S}^{d-1}$ *such that* $\|u - v\| = \beta \leq \frac{p}{4}$,

$$\Pr_{\tilde{P}}\Big[u \text{ and } v \text{ fall into different clusters}\Big] \leq \frac{100\beta d}{p}.$$

5.8 Equivalence of Lasserre relaxations

$$\max \sum_{e=(u,v)\in E} \sum_{i\in[k]} \left\langle \mathbf{x}_{u,i}, \mathbf{x}_{v,\pi_e^{uv}(i)} \right\rangle \; wt(e)$$

Subject to,

$$\begin{array}{llr} \forall S \subseteq V, |S| \leq t & \displaystyle\sum_{\sigma\in[k]^S} \|\mathbf{x}_{S,\sigma}\|^2 = 1 & \text{(VII)} \\ \forall (S,\sigma) \not\simeq (S',\sigma') & \langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S',\sigma'} \rangle = 0 & \text{(VIII)} \\ \forall (S,\sigma) \simeq (S',\sigma'), (T,\tau) \simeq (T',\tau') & & \\ (S \cup S', \sigma \cup \sigma') = (T \cup T', \tau \cup \tau') & \langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S',\sigma'} \rangle = \langle \mathbf{x}_{T,\tau}, \mathbf{x}_{T',\tau'} \rangle & \text{(IX)} \end{array}$$

Figure 5.3: Relaxation `L'-UG`(t) for Unique Games

lemma 5.8.1. *The constraints of the semi-definite program (SDP)* `L'-UG`(t) *imply the constraints of the SDP* `L-UG`(t) *and the constraints of the SDP* `L-UG`$(2t)$ *imply the constraints of the SDP* `L'-UG`(t).

Proof. Let $\mathbf{x}_{S,\sigma}$ be vectors satisfying `L'-UG`(t). We will show that they satisfy Equation (VI) of `L-UG`(t). Note that Equation (V) is contained in Equation (VIII) for $S = S'$ and Equation (IV) is contained in Equation (VII) with $S = \phi$.

As a first step, we prove Equation (VI) for $T = \phi$. Fix a set S then $\langle \mathbf{x}_\phi, \mathbf{x}_{S,\sigma} \rangle = \langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S,\sigma} \rangle$ by Equation (IX) which means

$$\left\langle \mathbf{x}_\phi, \sum_{\sigma \in [k]^S} \mathbf{x}_{S,\sigma} \right\rangle = \sum_{\sigma \in [k]^S} \langle \mathbf{x}_\phi, \mathbf{x}_{S,\sigma} \rangle = \sum_{\sigma \in [k]^S} \langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S,\sigma} \rangle = 1$$

by using Equation (VII). Also, note that $\sum\limits_{\sigma \in [k]^S} \mathbf{x}_{S,\sigma}$ is a unit vector by Equations (VII) and (VIII) and so is $\mathbf{x}_\phi$. Since the dot products of these two unit vectors is 1 it must be that they are equal which proves Equation (VI) for $T = \phi$.

Now we prove Equation (VI) for $|T| = |S| - 1$ and it is easy to see that this implies Equation (VI) for all $T \subseteq S$ by repeated application. So fix S, T as described, fix $\tau \in [k]^T$ and let σ_i, $i \in [k]$ be the k assignments on S which are consistent with τ. We know that

$$\sum_{\sigma \in [k]^S} \mathbf{x}_{S,\sigma} = \mathbf{x}_\phi$$

Taking the dot product of $\mathbf{x}_{T,\tau}$ with both sides of the previous equation and using Equations (VIII) and (IX) gives us $\langle \mathbf{x}_{T,\tau}, \mathbf{x}_{T,\tau} \rangle = \sum\limits_{i=1}^{N} \langle \mathbf{x}_{S,\sigma_i}, \mathbf{x}_{S,\sigma_i} \rangle$ and it is also similarly easy to see that

$$\left\langle \mathbf{x}_{T,\tau}, \sum_{i=1}^{N} \mathbf{x}_{S,\sigma_i} \right\rangle = \sum_{i=1}^{N} \langle \mathbf{x}_{S,\sigma_i}, \mathbf{x}_{S,\sigma_i} \rangle$$

Thus, $\mathbf{x}_{T,\tau}$ and $\sum\limits_{i=1}^{N} \mathbf{x}_{S,\sigma_i}$ are vectors of norm $\sum\limits_{i=1}^{N} \langle \mathbf{x}_{S,\sigma_i}, \mathbf{x}_{S,\sigma_i} \rangle$ whose dot product is also equal to the same number which means they must be equal. This proves the first part of the lemma.

Conversely, let $\mathbf{x}_{S,\sigma}$ be a solution for $\texttt{L-UG}(2t)$. We will show that it satisfies Equations (VII),(VIII) and (IX) of SDP $\texttt{L'-UG}(t)$.

To prove Equation (VII), fix a set $S \subseteq V$. We have,

$$1 = \langle \mathbf{x}_\phi, \mathbf{x}_\phi \rangle = \left\langle \sum_{\sigma \in [k]^S} \mathbf{x}_{S,\sigma}, \sum_{\sigma \in [k]^S} \mathbf{x}_{S,\sigma} \right\rangle = \sum_{\sigma,\sigma' \in [k]^S} \langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S,\sigma'} \rangle = \sum_{\sigma \in [k]^S} \langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S,\sigma} \rangle ,$$

where second equality uses Equation (VI) with $T = \phi$ and the the fourth equality uses Equation (V).

To prove Equation (VIII), fix S, S', σ, σ' such that $(S,\sigma) \not\simeq (S', \sigma')$. Then,

$$\langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S',\sigma'} \rangle = \left\langle \sum_{\substack{\tau \in [k]^{S \cup S'} \\ \tau|_S = \sigma}} \mathbf{x}_{S \cup S', \tau}, \sum_{\substack{\tau' \in [k]^{S \cup S'} \\ \tau'|_{S'} = \sigma'}} \mathbf{x}_{S \cup S', \tau'} \right\rangle = 0$$

where the last equality uses Equation (V) and the fact that the two summations consist of disjoint assignments since $(S,\sigma) \not\simeq (S', \sigma')$.

To prove Equation (IX), fix $S, S', T, T' \subseteq V$ of size at most t and their corresponding assignments $\sigma, \sigma', \tau, \tau'$ respectively such that $(S,\sigma) \simeq (S',\sigma'), (T,\tau) \simeq (T',\tau'), (S \cup S', \sigma \cup \sigma') = (T \cup T', \tau \cup \tau')$. Now,

$$\langle \mathbf{x}_{S,\sigma}, \mathbf{x}_{S',\sigma'} \rangle = \left\langle \sum_{\substack{\sigma'' \in [k]^{S \cup S'} \\ \sigma''|_S = \sigma}} \mathbf{x}_{S \cup S', \sigma''}, \sum_{\substack{\sigma'' \in [k]^{S \cup S'} \\ \sigma''|_{S'} = \sigma'}} \mathbf{x}_{S \cup S', \sigma''} \right\rangle = \langle \mathbf{x}_{S \cup S', \sigma \cup \sigma'}, \mathbf{x}_{S \cup S', \sigma \cup \sigma'} \rangle$$

where the first equality uses Equation (VI) and the second equality uses Equation (V) combined with the observation that $\sigma \cup \sigma'$ is the only assignment appearing in both the summations.

Similarly,

$$\langle \mathbf{x}_{T,\tau}, \mathbf{x}_{T',\tau'} \rangle = \langle \mathbf{x}_{T\cup T',\tau\cup\tau'}, \mathbf{x}_{T\cup T',\tau\cup\tau'} \rangle$$

and since $(S \cup S', \sigma \cup \sigma') = (T \cup T', \tau \cup \tau')$ Equation (IX) is proved as desired.

$\square$

Bibliography

[ABLT06] S. Arora, B. Bollobás, L. Lovász, and I. Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(1):19–51, 2006.

[ABSS97] S. Arora, L. Babai, J. Stern, and Z. Sweedyk. Hardness of approximate optima in lattices, co des, and linear systems. *Journal of Computer and System Sciences*, 54(2):317–331, 1997.

[AKKV05] Mikhail Alekhnovich, Subhash Khot, Guy Kindler, and Nisheeth K. Vishnoi. Hardness of approximating the closest vector problem with pre-processing. In *FOCS*, pages 216–225. IEEE Computer Society, 2005.

[AKKV12] Mikhail Alekhnovich, Subhash A Khot, Guy Kindler, and Nisheeth K Vishnoi. Hardness of approximating the closest vector problem with pre-processing. *Computational Complexity*, pages 1–13, 2012.

[ALM+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.

[AR05] Dorit Aharonov and Oded Regev. Lattice problems in NP $\cap$ co-NP. *J. ACM*, 52:749–765, September 2005.

[Aro94] Sanjeev Arora. Probabilistic checking of proofs and the hardness of approximation problems. *Ph.D. Thesis*, 1994.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *J. ACM*, 45(1):70–122, 1998.

[AS03] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.

[BB06] Maria-Florina Balcan and Avrim Blum. Approximation algorithms and online mechanisms for item pricing. In Joan Feigenbaum, John C.-I. Chuang, and David M. Pennock, editors, *ACM Conference on Electronic Commerce*, pages 29–35. ACM, 2006.

[BBCH07] Maria-Florina Balcan, Avrim Blum, T.-H. Hubert Chan, and MohammadTaghi Hajiaghayi. A theory of loss-leaders: Making money by pricing below cost. In Xiaotie Deng and Fan Chung Graham, editors, *WINE*, volume 4858 of *Lecture Notes in Computer Science*, pages 293–299. Springer, 2007.

[BBH+12] Boaz Barak, Fernando GSL Brandão, Aram W Harrow, Jonathan Kelner, David Steurer, and Yuan Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the 44th symposium on Theory of Computing*, pages 307–326. ACM, 2012.

[BD06] Andrei Bulatov and Víctor Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM Journal on Computing*, 36(1):16–27, 2006.

[BK06] P. Briest and P. Krysta. Single-minded unlimited supply pricing on sparse instances. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 1093–1102. ACM, 2006.

[Bor85] C. Borell. Geometric bounds on the Ornstein-Uhlenbeck velocity process. *Z. Wahrsch. Verw. Gebiete*, 70(1):1–13, 1985.

[CMM06] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 205–214, 2006.

[CMM09] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Integrality gaps for sherali-adams relaxations. In Michael Mitzenmacher, editor, *STOC*, pages 283–292. ACM, 2009.

[DG99] S. Dasgupta and A. Gupta. An elementary proof of the Johnson-Lindenstrauss lemma. Technical Report TR-99-006, U. C. Berkeley, 1999.

[DGKR05] Irit Dinur, Venkatesan Guruswami, Subhash Khot, and Oded Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *SIAM Journal on Computing*, 34(5):1129–1146, 2005.

[DHFS06] E.D. Demaine, M.T. Hajiaghayi, U. Feige, and M.R. Salavatipour. Combination can be hard: Approximability of the unique coverage problem. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, pages 162–171. ACM, 2006.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12, 2007.

[DKRS03] Irit Dinur, Guy Kindler, Ran Raz, and Shmuel Safra. Approximating CVP to within almost-polynomial factors is NP-hard. *Combinatorica*, 23(2):205–243, 2003.

[DMR09] Irit Dinur, Elchanan Mossel, and Oded Regev. Conditional hardness for approximate coloring. *SIAM J. Comput.*, 39(3):843–873, 2009.

[DS05] Irit Dinur and Samuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, pages 439–485, 2005.

[ERRS09] K. Elbassioni, R. Raman, S. Ray, and R. Sitters. On profit-maximizing pricing for the highway and tollbooth problems. *Algorithmic Game Theory*, pages 275–286, 2009.

[ESZ07] K. Elbassioni, R. Sitters, and Y. Zhang. A quasi-PTAS for profit-maximizing pricing on line graphs. *ESA 2007*, 2007.

[FGL+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.

[FGRW09] V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu. Agnostic learning of monomials by halfspaces is hard. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science*, October 2009.

[FJ12] Uriel Feige and Shlomo Jozeph. Universal factor graphs. *Automata, Languages, and Programming*, pages 339–350, 2012.

[FM04] U. Feige and D. Micciancio. The inapproximability of lattice and coding problems with preprocessing. *Journal of Computer and System Sciences*, 69(1):45–67, 2004.

[GHK+05] Venkatesan Guruswami, J.D. Hartline, A.R. Karlin, David Kempe, Claire Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1164–1173. Society for Industrial and Applied Mathematics, 2005.

[GKL03] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proc.* 44^{th} *IEEE FOCS*, 2003.

[GKO+10] Venkatesan Guruswami, Subhash Khot, Ryan O'Donnell, Preyas Popat, Madhur Tulsiani, and Yi Wu. SDP gaps for 2-to-1 and other label-cover variants. In *ICALP (1)*, pages 617–628, 2010.

[GMPT07] K. Georgiou, A. Magen, T. Pitassi, and I. Tourlakis. Integrality gaps of 2 - o(1) for vertex cover SDPs in the Lovész-Schrijver hierarchy. In *Proc.* 48^{th} *IEEE FOCS*, pages 702–712, 2007.

[GMR08] V. Guruswami, R. Manokaran, and P. Raghavendra. Beating the random ordering is hard: Inapproximability of maximum acyclic subgraph. In *Proc.* 49^{th} *IEEE FOCS*, 2008.

[GR11] F. Grandoni and T. Rothvoß. Prizing on paths: A PTAS for the highway problem. *SODA*, 2011.

[GRSW10] Venkatesan Guruswami, Prasad Raghavendra, Rishi Saket, and Yi Wu. Bypassing UGC from some optimal geometric inapproximability results. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:177, 2010.

[GS09] Venkatesan Guruswami and Ali Kemal Sinop. Improved inapproximability results for maximum k-colorable subgraph. In *Proceedings of the 12th International Workshop on Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques (APPROX)*, pages 163–176, 2009.

[GS10] I. Gamzu and D. Segev. A sublogarithmic approximation for highway and tollbooth pricing. *Automata, Languages and Programming*, pages 582–593, 2010.

[GS11] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for quadratic integer programming with PSD objectives. In *FOCS*, 2011.

[GT06] A. Gupta and K. Talwar. Approximating unique games. In *SODA '06: Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete algorithm*, 2006.

[GVLSU06] A. Grigoriev, J. Van Loon, R. Sitters, and M. Uetz. How to sell a graph: Guidelines for graph retailers. In *Graph-Theoretic Concepts in Computer Science*, pages 125–136. Springer, 2006.

[GvLU10] A. Grigoriev, J. van Loon, and M. Uetz. On the complexity of the highway pricing problem. *SOFSEM 2010: Theory and Practice of Computer Science*, pages 465–476, 2010.

[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, 1995.

[Hås01] J. Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001.

[HK05] J.D. Hartline and V. Koltun. Near-optimal pricing in near-linear time. *Algorithms and Data Structures*, pages 422–431, 2005.

[JL84] W. Johnson and J. Lindenstrauss. Extensions of lipschitz maps into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

[Kho02a] S. Khot. Hardness results for coloring 3-colorable 3-uniform hypergraphs. In *43rd IEEE FOCS*, pages 23–32, 2002.

[Kho02b] Subhash Khot. On the power of unique 2-prover 1-round games. In *STOC*, pages 767–775, 2002.

[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, 2007.

[KKMS09] Rohit Khandekar, Tracy Kimbrel, Konstantin Makarychev, and Maxim Sviridenko. On hardness of pricing items for single-minded bidders. In Irit Dinur, Klaus Jansen, Joseph Naor, and José D. P.

Rolim, editors, *APPROX-RANDOM*, volume 5687 of *Lecture Notes in Computer Science*, pages 202–216. Springer, 2009.

[KPS10] Subhash Khot, Preyas Popat, and Rishi Saket. Approximate Lasserre integrality gap for unique games. In *APPROX-RANDOM*, pages 298–311, 2010.

[KPV12] Subhash A Khot, Preyas Popat, and Nisheeth K Vishnoi. $2^{\log^{1-\epsilon} n}$ hardness for the closest vector problem with preprocessing. In *Proceedings of the 44th symposium on Theory of Computing*, pages 277–288. ACM, 2012.

[KS09] Subhash Khot and Rishi Saket. SDP integrality gaps with local l_1-embeddability. In *FOCS*, pages 565–574, 2009.

[KS11] Subhash Khot and Rishi Saket. On the hardness of learning intersections of two halfspaces. *J. Comput. Syst. Sci.*, 77(1):129–141, 2011.

[KV05] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into l_1. In *FOCS*, pages 53–62, 2005.

[Las01] Jean B. Lasserre. An explicit exact SDP relaxation for nonlinear 0-1 programs. In *IPCO*, pages 293–303, 2001.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[LLS90] J.C. Lagarias, H.W. Lenstra, and C.P. Schnorr. Korkine-Zolotarev bases and successive minima of a lattice and its reciprocal lattice. *Combinatorica*, 10:333–348, 1990.

[MOO05] E. Mossel, R. O'Donnell, and K. Oleszkiewicz. Noise stability of functions with low infuences invariance and optimality. In *Proc.* 46^{th} *IEEE FOCS*, 2005.

[MR10] Dana Moshkovitz and Ran Raz. Two-query pcp with subconstant error. *J. ACM*, 57(5), 2010.

[MV10] Daniele Micciancio and Panagiotis Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. In *STOC*, pages 351–358, 2010.

[OW09] Ryan O'Donnell and Yi Wu. Conditional hardness for satisfiable 3-csps. In *STOC*, pages 493–502, 2009.

[OWZ11] Ryan O'Donnell, Yi Wu, and Yuan Zhou. Hardness of Max-2Lin and Max-3Lin over integers, reals, and large cyclic groups. In *Proc. 26th CCC*, 2011.

[PW12] Preyas Popat and Yi Wu. On the hardness of pricing loss-leaders. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 735–749. SIAM, 2012.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *STOC*, pages 245–254, 2008.

[Rag09] Prasad Raghavendra. *Approximating NP-hard problems: efficient algorithms and their limits.* PhD thesis, University of Washington, 2009.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998.

[Reg04] Oded Regev. Improved inapproximability of lattice and coding problems with preprocessing. *IEEE Transactions on Information Theory*, 50(9):2031–2037, 2004.

[RR06] Oded Regev and Ricky Rosen. Lattice problems and norm embeddings. In *STOC*, pages 447–456, 2006.

[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. Integrality gaps for strong SDP relaxations of Unique Games. In *FOCS*, pages 575–585, 2009.

[Sch87] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theor. Comput. Sci.*, 53:201–224, 1987.

[Sch08] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 593–602, 2008.

[STT07a] G. Schoenebeck, L. Trevisan, and M. Tulsiani. A linear round lower bound for Lovasz-Schrijver SDP relaxations of vertex cover. In *IEEE Conference on Computational Complexity*, pages 205–216, 2007.

[STT07b] G. Schoenebeck, L. Trevisan, and M. Tulsiani. Tight integrality gaps for Lovasz-Schrijver LP relaxations of vertex cover and max cut. In *Proc. 39th ACM STOC*, pages 302–310, 2007.

[Tal94] Michel Talagrand. On Russo's approximate zero-one law. *The Annals of Probability*, pages 1576–1587, 1994.

[Tan09] Linqing Tang. Conditional hardness of approximating satisfiable Max 3CSP-q. *Algorithms and Computation*, pages 923–932, 2009.

[Tre05] L. Trevisan. Approximation algorithms for Unique Games. In *Proc. 46th IEEE FOCS*, 2005.

[Tul09] Madhur Tulsiani. CSP gaps and reductions in the Lasserre hierarchy. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, pages 303–312, 2009.

[Wu11] Yi Wu. Pricing loss leaders can be hard. In *Innovations in Computer Science*, 2011.