Approximability and Mathematical Relaxations

Rajsekar Manokaran

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

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Sanjeev Arora

November 2012

Abstract

The thesis ascertains the approximability of classic combinatorial optimization problems using mathematical relaxations. The general flavor of results in the thesis is: a problem P is hard to approximate to a factor better than one obtained from the R relaxation, unless the Unique Games Conjecture is false.

Almost optimal inapproximability is shown for a wide set of problems including METRIC LABELING, MAX. ACYCLIC SUBGRAPH, various packing and covering problems. The key new idea in this thesis is in coverting hard instances of relaxations (a.k.a integrality gap instances) into a proof of inapproximability (assuming the UGC). In most cases, the hard instances were discovered prior to this work; our results imply that these hard instances are possibly strong bottlenecks in designing approximation algorithms of better quality for these problems.

For ordering problems such as MAX. ACYCLIC SUBGRAPH and FEEDBACK ARC SET, such hard instances were previously unknown. For these problems (see chapter 6), we construct such hard instance by using the reduction designed to prove the inapproximability. The hard instances show that all ordering problems are hard to approximate to a factor larger than the expected fraction satisfied by a random ordering: i.e., all ordering CSPs are approximation resistant.

Techniques involve using mathematical relaxations to obtain local distributions, converting them into low degree functions defined over the boolean cube and using the invariance principle to analyse such function.

I believe the thesis will be a good reference, both for the results proven therein, and for the framework designed in ascertaining approximability from mathematical relaxations.

Acknowledgements

I would like to thank my parents, Bama and Manokaran, and my sister Nithya for providing a wonderful family. The values imbibed in me by them has always been crucial throughout my life.

I thank my adviser, Prof. Sanjeev Arora, for his constant support and advice during my stay here at Princeton. My colloborations with him not only led to this thesis but has also refined my research methodologies. I am grateful to the complete independence in decisions regarding research directions, letting me grow a taste for research in theoretical computer science, while fully supporting me financially. NSF grants MSPA-MCS 0528414, 0830673, 0832797, 528414 and ITR 0205594 were instrumental in this academic freedom.

I thank Princeton University and the Computer Science department for providing a placid environment, ideal for pursuing research. Special thanks to Melissa Lawson and Mitra Kelly for handling travel reimbursements, and various other paper work making travel to conferences and hence public dissemination of my research work easier.

I would like to thank my numerous friends, both in Princeton and elsewhere for providing a second family, while away from home. This list is by no means complete: special thanks to Anirudh Badam, Vijay Krishnamurthy, Abishek Kumarasubramanian, Ana Bell, Arul Shankar, Ila Varma and Aravindan Vijayaraghavan. They were always around during the toughest of my times, nudging me towards my goals.

I am fortunate to have been mentored by Dr. Ramarathnam Venkatesan. His thoughts on things both pertaining to research and otherwise have been a guiding light to me.

Almost my entire knowledge of techniques used in theoretical computer science stems from my collaboration during and before I started pursuing my doctoral degree. I would like to thank all my collaborators, an incomplete list of whom would be Prasad Raghavendra, Nisheeth Vishnoi, Konstantin Makarychev, Moses Charikar, Aravindan Vijarayaghavan, Aditya Bhaskara, Ravishankar Krishnaswamy, Venkatesan Guruswami, Johan Håstad, Dana Moshkovitz, Sushant Sachdeva and Arnab Bhattacharya.

Lastly, I would like to thank someone initialled D. N., for teaching me how to live.

Dedicated to D. N.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

A standard approach to coping with the NP-hardness of a combinatorial optimization problem involves designing algorithms that guarantee a solution within a factor α of the optimum (a.k.a an α-approximation algorithm). Assuming $\mathsf{P} \neq \mathsf{NP}$, this is the best one can hope for, and the design is aimed at minimizing α. In this context, *approximability* refers study and design of the best approximation algorithms. Approximability is, by now a large field and draws techniques from a wide variety of mathematical tools like metric embeddings, matroid theory, etc. The analysis has made important and fundamental contributions to the fields it draws from.

In this thesis, we will focus on one of the most powerful and widely used tool: mathematical relaxations. We develop techniques to ascertain the approximability of a large number of locally constrained optimization problems. The general flavor of results in the thesis is: a problem P is hard to approximate to a factor better than one obtained from the R relaxation, unless the Unique Games Conjecture is false. Figure 1.1 tabulates all the new results shown in this thesis.

The Unique Games conjecture, introduced by Khot [38], hypothesizes that a particular optimization problem known as the Unique Games is NP-hard to approximate upto any constant factor. This conjecture has proven very useful in obtaining tight inapproximability results. However, being more recent than traditional complexity assumptions (such as $\mathsf{P} \neq \mathsf{NP}$), the conjecture is not as widely believed.

Problem	Result	Remarks
3-MULTIWAY CUT	$^{12}/_{11} - \varepsilon$ inapprox.	Tight inapprox. (see [33].)
k-MULTIWAY CUT	$\theta - \varepsilon$ inapprox.	θ is the integrality gap of the earthmover relaxation (see fig. 4.1) and hence tight.
0-EXTENSION	$O(\sqrt{\log k})$-inapprox	Tight upto factors hidden in $O(\cdot)$ [10]
k-METRIC LABELING	$O(\log k)$-inapprox	Tight upto factors hidden in $O(\cdot)$ [34]
MAX. ACYCLIC SUBGRAPH	$1/2 + \varepsilon$-inapprox.	Tight. A random ordering is a 1/2-approx.
FEEDBACK ARC SET	$\omega(1)$-inapprox.	Improves over a small const. hardness.
BETWEENNESS	$1/3 + \varepsilon$-inapprox.	Tight. A random ordering is a 1/3-approx.
STRICT CSP	$\theta - \varepsilon$ inapprox.	Tight. θ is the integrality gap of the relaxation (see fig. 5.1)
ORDERING CSP	$\theta + \varepsilon$ inapprox.	Tight. θ is the expected payoff of a random ordering.

Figure 1.1: New Results in this thesis

1.2 Approximation Algorithm Design

For the sake of this exposition, we will consider a simple optimization problem, explaining key ingredients in designing an approximation algorithm using a mathematical relaxation and explaining our approach to determining its approximability. The example is the 3-WAY CUT problem, defined below.

Example 1.2.1 (3-WAY CUT). *Given a graph $\mathcal{J} = (\mathcal{V}, \mathcal{E})$, along with three specific* terminal *vertices, $t_1, t_2, t_3 \in \mathcal{V}$, the* 3-WAY CUT *problem asks for the partitioning of the vertices—i.e., disjoint sets X, Y and Z— that separates the three terminals from each other with minimum number of edges crossing partitions.*

This simple extension of the well-known MIN CUT problem is already NP-hard. Towards designing an approximation algorithm, one first writes an integer program to solve the problem. Integer programming, being a non-convex optimization problem is NP-hard. One would then *relax* the non-convex constraints in the integer program, obtaining a *mathematical relaxation*. There are two families of relaxations that are widely used: *linear* (LP) and *semi-definite* (SDP) relaxations.

$$\mathrm{OPT}(\mathcal{J}) \stackrel{\text{def}}{=} \min \sum_{e=(w,v)\in\mathcal{E}} \frac{1}{2}\left[\, |x_w - x_v| + |y_w - y_v| + |z_w - z_v|\, \right] \tag{1.1}$$

subject to

$$\forall_{v\in\mathcal{V}} \qquad x_v + y_v + z_v = 1; \quad x_v, y_v, z_v \in \{0,1\} \tag{1.2}$$

$$x_{t_1} = 1; \quad y_{t_1} = 0; \quad z_{t_1} = 0$$

$$x_{t_2} = 0; \quad y_{t_2} = 1; \quad z_{t_2} = 0$$

$$x_{t_3} = 0; \quad y_{t_3} = 0; \quad z_{t_3} = 1$$

Figure 1.2: Integer Program for 3-WAY CUT

The integer program for 3-WAY CUT is in fig. 1.2. Equation (1.2) in fig. 1.2 is the non-convex "integer" constraint, which can be relaxed to

$$x_v + y_v + z_v = 1; \qquad x_v, y_v, z_v \in [0,1]$$

giving us a *linear program* (see fig. 1.3 for the complete description).

The motive behind this relaxation is to be able to compute the optimum in polynomial time. The optimum value of the LP denoted by $\mathrm{LP}(\mathcal{J})$ is indeed tractable while the optimum of the IP (denoted by $\mathrm{OPT}(\mathcal{J})$) is NP-hard to compute. Further, since the constraints where only relaxed, a feasible solution to the IP is already a feasible solution to the LP. Thus, we know that $\mathrm{LP}(\mathcal{J}) \le \mathrm{OPT}(\mathcal{J})$. On the other hand, if there was an α such that

$$\mathrm{LP}(\mathcal{J}) \le \mathrm{OPT}(\mathcal{J}) \le \alpha \cdot \mathrm{LP}(\mathcal{J}) \tag{1.5}$$

$$\mathrm{LP}(\mathcal{J}) \stackrel{\text{def}}{=} \min \sum_{e=(w,v)\in\mathcal{E}} \frac{1}{2}\,[\ |x_w - x_v| + |y_w - y_v| + |z_w - z_v|\ ] \tag{1.3}$$

subject to

$$\begin{aligned} \forall_{v\in\mathcal{V}} \qquad & x_v + y_v + z_v = 1; \quad 0 \le x_v, y_v, z_v \le 1 \qquad (1.4)\\ & x_{t_1} = 1; \quad y_{t_1} = 0; \quad z_{t_1} = 0\\ & x_{t_2} = 0; \quad y_{t_2} = 1; \quad z_{t_2} = 0\\ & x_{t_3} = 0; \quad y_{t_3} = 0; \quad z_{t_3} = 1 \end{aligned}$$

Figure 1.3: Linear Program for 3-WAY CUT

then, we have an *approximation* to the intractable optimum of 3-WAY CUT. The value α measures the *quality* of the relaxation and is known as the *integrality gap* of the relaxation. An integrality gap instance of factor β is an instance $\mathcal{J}$ such that $\mathrm{OPT}(\mathcal{J}) \geq \beta \cdot \mathrm{LP}(\mathcal{J})$. In other words, an integrality gap instance potrays the limits to which this relaxation can yield approximations. For example, the work of Karger et al [33] shows that $\alpha = \beta = 12/11$ for the above relaxation.

Aside: the relaxation described above generalizes to the so called earthmover relaxation (see fig. 4.1) for a large class of "cut" problems known as the metric labeling problems (refer [16, 10, 41]). From the work of Karger et al [33], we know that for the 3-WAY CUT problem, $\alpha = 12/11$ ($\alpha = \log(n)$ for the general metric labeling problems, where n is the number of vertices in $\mathcal{J}$ [34]).

Rounding Procedure. In the integer program, x_v represents vertex v being assigned to partition 1 (and y_v, z_v, to partitions 2 and 3 resp.) However, the optimum solution to the LP might assign real values in $[0, 1]$ to the variable, which is not readily amenable to being read off as a parititon. To obtain a partitioning of the vertices, one designs a *rounding procedure* that converts the solution to the LP into a partition. Of course, since $\mathrm{LP}(\mathcal{J})$ is smaller than $\mathrm{OPT}(\mathcal{J})$, the rounding procedure can only produce a partition by increasing (i.e., with a loss in) the objective. Following the terminology above, we say a procedure is a factor γ rounding if the cost of the solution produced is at most $\gamma \cdot \mathrm{LP}(\mathcal{J})$. For example, Kleinberg and Tardos [41] designed a simple 2-approximation by rounding this relaxation, which was later improved to $^{12}/_{11}$ [33].

Together, *relax-and-round* is a powerful paradigm in approximation algorithm design.

1.3 Inapproximability Reductions

The study of approximation algorithms has also lead to the following natural contra-positive (or dual) question: what is the best polynomial time approximation admitted by a problem? Such an inapproximability is proven via a polynomial time reduction from 3-SAT. Consider a polynomial time reduction from 3-SAT to an optimization problem P such that satisfiable instances of 3-SAT map to instances of P whose optimum is at most ρ while unsatisfiable instances map to ones whose optimum is at least $\beta \cdot \rho$. Now, assuming $\mathsf{P} \neq \mathsf{NP}$, no algorithm can efficiently approximate instances of P by a factor strictly smaller than β.

The pioneering work on the proof of the PCP theorem showed a self-reduction of 3-SAT such that unsatisfiable instances map to instances where at most a fraction $(1 - \rho)$ of the clauses are simultaneously satisfiable while satisfiable instances remain completely satisfiable [22, 3, 2]. This work spurred a large body of work, the notable of which include the parallel repetition theorem [60] and the 3-bit PCP showing "tight" inapproximability of 3-SAT (among other problems [29]). Here, *tight* refers to a factor $\rho + \varepsilon$ inapproximability while a ρ-approximation algorithm is also known (ε is a constant which can be chosen to be arbitrarily small, while incurring a larger running time for the reduction).

Recently, the conjecture of Khot [38], the Unique Games conjecture has lead to tight inapproximability for fundamental problems such as Max Cut and Vertex Cover which had resisted proofs of inapproximability (see [18] for a standard inapproximability proof of Vertex Cover; [39] for a UG based inapproximability). In all these tight inapproximability results, the reduction looks somewhat magical: although inspired by the structure of instances that are hard to approximate, it is not at all evident from the reduction.

1.4 Systematic Design in Approximability

This thesis puts forth the following systematic design of approximation algorithms: given a optimization problem P, one first writes a mathematical relaxation to approximate (the optimum of instances of) P. Relaxations enforce local constraints, providing us locally "integral" distributions. For example, the LP solution (call it $\{(x_v, y_v, z_v)\}_v$) in fig. 1.3 gives the a distribution $\mathcal{D}_e$ over $[3] \times [3]$ for every edge $e = (v, w)$. such that

$$\mathop{\mathbf{P}}_{(a,b)\in\mathcal{D}_e}[x \neq y] = \frac{1}{2}\left[\ |x_w - x_v| + |y_w - y_v| + |z_w - z_v|\ \right] \tag{1.6}$$

Thus, we have a local distribution (i.e., one for each edge) whose cost is exactly the cost of the corresponding edge in the linear program. Of course, unless the linear program has an integrality gap equal to 1, there is no such global distribution.

Now, we compose an integrality gap instance for the linear program with an instance of Unique Games (UG), producing instances of 3-WAY CUT such that:

- if the UG instance is almost completely satisfiable, the instance of 3-WAY CUT whose optimum is at most $\mathrm{LP}(\mathcal{J}) + \varepsilon$.
- on the other hand, almost unsatisfiable instances of UG produce instances whose optimum is at least $\mathrm{OPT}(\mathcal{J}) - \varepsilon$.
- the size of the instance is polynomial in the size of the UG instance.

The Unique Games conjecture says that the UG instances in items 1 and 2 above are NP-hard to distinguish between. Here, ε is a constant that can be made arbitrarily small while incurring a increased running time for the reduction. Thus, choosing ε small enough shows a inapproximability (assuming the UGC) of a factor almost $\mathrm{OPT}(\mathcal{J})/\mathrm{LP}(\mathcal{J})$, the integrality gap ratio.

Note that when the integrality gap of the problem is some constant β (which is the case here: $\beta = 12/11$), one can always find an instance $\mathcal{J}$ of size a function of ε such that the integrality gap of $\mathcal{J}$ is $\beta - \varepsilon$. Composing with this instance gives almost optimal inapproximability for 3-way Cut. Further, the only fact needed here is that β is a constant, and the inapproximabiltiy is in fact, oblivious to the value of β!

As mentioned earlier, the relaxation described above generalizes to a fairly large class of labeling problems known as Metric Labeling. The reduction, which only uses the the local distributions, also generalizes to this wide class, giving tight inapproximabilities for all cases of Metric Labeling where the integrality gap is bounded by a constant (see chapter 4 for more details). This is a trait of our reduction: since a relaxation generalizes (or specializes) to particular subcases of a general problem, the reduction also automatically proves inapproximability results for these special cases.

1.5 Unique Games Conjecture

The Unique Games Conjecture (UGC) is an auxillary hardness assumption similar to $\mathsf{P} \neq \mathsf{NP}$, introduced by Khot [38]. The conjecture has been instrumental in obtaining tight inapproximability results for a wide number of problems [38, 39, 58, 27, 48, 55, 5, 4]. The UGC hypothesizes that a constraint satisfaction problem known as the UG is hard to approximate in the following strong sense: even instances where almost all the constraints can be satisfied are NP-hard to distinguish from instances where almost none of the constraints can be satisfied.

An instance of UG, Υ of label size R is simply a graph where edges are augmented by permutations $\pi_e : [R] \to [R]$. An assignment refers to labeling each vertex with a number between 1 and R. Such an assignment is said to satisfy an edge if the end points are mapped to each other by the permutation corresponding to the edge. The UG problem asks to find an assignment that satisfies the most number of edges.

1.6 Overview of the Reduction

At the outset, our reduction follows the paradigm of using dictatorship tests introduced in [29]. Given a UG instance, Υ, we replace each vertex with a gadget, $[3]^R$ where R represents the size of the label set of Υ. Next, we add edges between neigboring gadgets using a distribution called the dictatorship test: a R-wise direct product of the local distribution induced by the solution on edges of the integrality gap instance (this is a simplified presentation. Refer chapter 3 and section 4.4 for a precise description).

Given a labeling of Υ, we use *dictator* functions: $f_i : [3]^R \to [3]$ given by $f(x) = x_i$ to form an assignment for the instance output from the reduction. For each vertex, i is chosen as the label assigned to it. In the analysis of the cost, the R-wise product has no effect since the assignment to a point in the gadget is only dictated by its $i\hat{}$th coordinate. Thus the cost of the assignment can be bounded by an equation similar to eq. (1.6), which then relates to $\mathrm{LP}(\mathcal{J})$.

The meat of the proof is in lower bounding the cost when Υ is far from being satisfiable. Here, the key idea is to prove that most pairs of neighboring gadgets can not have correlated coordinates controlling the assignment (where i and j are said to be correlated if assigning i and j to the two neigboring vertices of Υ satisfies the permutation constraint on the edge connecting the vertices). An averaging argument then says that for most edges of Υ, the pair of gadgets either is either dictated by coordinates that are uncorrelated or that the assignment on the gadget does not have a "few" coordinates that dictate the assignment.

In the former case, the assignment can be shown to cost as much as a random assignment to the instance (which is much larger than $\mathrm{LP}(\mathcal{J})$). The latter case is handled by the R-wise product structure of the distribution. The behaviour of R-wise product distributions is studied using the *invariance principle* introuced in [51, 50, 31], proving that the cost is high and in fact almost as large as $\mathrm{OPT}(\mathcal{J})$.

1.7 Organization of this thesis

We begin by setting up notation, and other preliminaries needed in reading the proofs in this thesis (see chapter 2). Refer here for a quick primer on writing mathematical relaxations and theorems on the structure of boolean functions. Chapter 3 describes the basic building block in all our inapproximability reductions: the dictatorship test gadget. The chapter also contains analysis of this gadget, used in the remaining chapters in proving the inapproximability. Our results on metric labeling problems are in chapter 4. Strict constraint satisfaction problems are analysed in chapter 5 and ordering constraint satisfaction problems in chapter 6.

Chapter 2

Preliminaries

In this chapter, we will setup the notation and other groundwork needed to read the rest of the thesis. Section 2.1 describes the notation. Section 2.2 is a short primer on mathematical relaxations. Section 2.3 describes definitions and a few basic theorems form harmonic analysis that come in handy in our application of the *invariance principle* of Mossel et al [51, 50, 31].

2.1 Setup and Notation

We use boldface letters $\mathbf{z}$ to denote vectors $\mathbf{z} = (z_1, \ldots z_k)$. For an integer k, $\blacktriangle_k$ or $\blacktriangle_{[k]}$ denotes all probability distributions on k elements; that is, the convex hull of the set $\{(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, \ldots, 0, 1)\}$. In general, for a set S, $\blacktriangle_S$ denotes the set of probability distributions over the sets.

2.2 Mathematical Relaxations

In our setting, an *integer program* is an optimization procedure written over a set of variables $Y_1, Y_2, \ldots Y_n$ each constrained to belong to the $\{0, 1\}$; constrained by another set of linear functions and minimizes a linear objective. For example, fig. 2.1 is an integer program over a set of variables, indexed by $\mathcal{V}$ (and constraints indexed by $\mathcal{E}$). For every $e = (v, w) \in \mathcal{E}$, (Y_v, Y_w) takes values in $\{(0,1),(1,0),(1,1)\}$. (enforced in eq. (2.2)). When $\mathcal{E}$ is the edge set of a graph on (the vertex set) $\mathcal{V}$, the objective measures the size of the minimum Vertex Cover of the graph.

$$\mathrm{OPT}(\mathcal{J}) \stackrel{\text{def}}{=} \min \sum_{v \in \mathcal{V}} Y_v \tag{2.1}$$

$$\text{subject to}$$

$$\forall_e \qquad Y_v + Y_w \geq 1 \quad Y_v \in \{0,1\} \tag{2.2}$$

$$\tag{2.3}$$

Figure 2.1: Integer Program (for Vertex Cover)

The value of a solution—collection of values Y_v one for each $v \in \mathcal{V}$, satisfying the above constraints—$\mathsf{val}(\mathcal{J}, \{Y_v\})$ and the optimum, $\mathrm{OPT}(\mathcal{J})$ is the minimum over all solutions. Integer programming is in general NP-hard and in fact hard to approximate to any polynomial factor because of non-convex constraint forcing the variables to take on discrete values [62]. For this reason, these constraints are relaxed into a linear program. For instance, a linear program (used in the design of algorithms for the Vertex Cover problem) is such a relaxation (refer fig. 2.2).

$$\mathrm{LP}(\mathcal{J}) \stackrel{\text{def}}{=} \min \sum_{v \in \mathcal{V}} Y_v \tag{2.4}$$

$$\text{subject to}$$

$$\forall_e \qquad Y_v + Y_w \geq 1 \quad Y_v \in [0,1] \tag{2.5}$$

$$\tag{2.6}$$

Figure 2.2: Linear Program (for Vertex Cover)

Linear programs can be optimized in time polynomial in the number of variables, number of constraints and the number of bits of accuracy (refer [24] for a good introduction). Accuracy will not be an issue in the problems we consider: polynomial bits of accuracy suffices. As before, we denote the value of a solution Y_v to the above relaxation by $\mathsf{val}(\mathcal{J}, \{Y_v\})$ and the minimum such solution by $\mathrm{LP}(\mathcal{J})$.

Hierarchies of Linear Programs. We will use sophisticated linear relaxations where the variables are locally constrained to be a convex combination of valid solutions of the integer program (while only focussed on the variables in the constraint).

For example, in the linear relaxation above, the constraint in eq. (2.5) says that for every (v, w), the variables (Y_v, Y_w) takes values in the convex hull of the set $((0,1),(1,1),(1,0))$. However, for a set of three variables (v, w, u), the solution $(1/2, 1/2, 1/2)$ is valid for the linear program while not for the integer program if every pair of vertices is an edge (this example is instructive to verify).

On the other hand, such constraints can be enforced by LP hierarchies (refer [46, 64]). Given a integer program (say, over n variables and $\mathrm{poly}(n)$ constraints) and an integer k, one can write a linear program in $O(n^k)$ variables and constraints such that for every subset S of the variables of size at most k, a valid solution to the elements in S, (say Y_S) is a convex combination of valid integer solutions (projected onto the set S). Thus, if such a set is known before hand, say $C \subset \{0,1\}^k$, then we write the constraint:

$$Y_S \in \blacktriangle(C) \tag{2.7}$$

Semidefinite Programming. A more sophisiticated family of mathematical relaxations are semidefinite relaxations (SDP) (a.k.a vector programming relaxation). In a vector program, the variables are replaced with vectors $(\mathbf{Y_v})$ and constraints with linear constraints over the inner product.

Hierarchies for semidefinite programs [44] are known, and will be used as a building block in writing complex SDPs.

$$\mathrm{SDP}(\mathcal{J}) \stackrel{\text{def}}{=} \min \sum_{v \in \mathcal{V}} \langle \mathbf{Y}_v, \mathbf{Y}_0 \rangle \tag{2.8}$$

subject to

$$\forall_e \qquad \langle \mathbf{Y}_v, \mathbf{Y}_0 \rangle + \langle \mathbf{Y}_w, \mathbf{Y}_0 \rangle \geq 1 \tag{2.9}$$

$$\forall_v \qquad \langle \mathbf{Y}_v, \mathbf{Y}_0 \rangle \geq 0 \tag{2.10}$$

Figure 2.3: Semidefinite Program (for VERTEX COVER)

Integrality Gap. The integrality gap of a relaxation (for a certain problem $\mathcal{P}$) is the supremum (or inf. if the objective is to max.) over instances $\mathcal{J}$ of $\mathcal{P}$, of the ratio of the optimum to the optimum of the relaxation: $\mathrm{OPT}(\mathcal{J})/\mathrm{SDP}(\mathcal{J})$ (and similarily for LP relaxations). An integrality gap instance for a relaxation is simply an instance of the problem, $\mathcal{J}$, and set of solution to the relaxation for the instance. $\mathbf{Y}_v$. The relaxation one is interested in, for a particular problem, will be clear from the context.

2.3 Fourier Analysis over the Boolean Cube

Given a measure Ω over a finite probability space $\mathcal{X}$, we will work with random variables Z from $(\mathcal{X}, \Omega)$ to $\mathbb{R}$ that are square integrable. The space of such random variables has an orthonormal basis $\{\chi_0, \chi_1, \ldots \chi_{t-1}\}$ where t is the size of $\mathcal{X}$ ($\chi_0 = 1$ is customary). The R-wise product $\mathcal{X}^R$ with a product measure has a orthonormal basis indexed by sets $S \in [t]^R$. The fourier decomposition is this multilinear polynomial of Z:

$$Z = \sum_S \hat{Z}(S) \chi_S \tag{2.11}$$

The norm of Z is $\|Z\|_2^2 = \sum_S \hat{Z}_S^2$. The influence of coordinate $j \in [R]$ in Z is:

$$\mathrm{Inf}_j(Z) = \sum_{\{S \in [t]^R | S_j \neq 0\}} \hat{Z}^2(S)$$

where $\hat{Z}$ is the coefficient of χ_S when Z is written in the orthonormal basis. The *degree* of a random variable is $D = \max_S |S|$ where the size of a set S is the number of coordinates j such that $S_j \neq 0$.

The following simple lemmas will come in handy in our analysis.

Lemma 2.3.1. *For every D-degree random variable Z, there are at most $\|Z\|_2^2 D/\tau$ coordinates such that* $\mathrm{Inf}_j(Z) \geq \tau$.

Proof. Every coordinate whose influence is at least τ contributes at least τ/D towards the norm of the random variable. □

Lemma 2.3.2. *Let $\{Z_k\}$ be a finite collection of random variables and let $Z = \mathbf{E}_k[Z_k]$ be the average random variable (where the expectation is over a uniform k from the collection. Then,*

$$\mathop{\mathbf{E}}_k[\mathrm{Inf}_j(Z_k)] \geq \mathrm{Inf}_j(\mathop{\mathbf{E}}_k(Z_k)) = \mathrm{Inf}_j(Z).$$

In particular, if coordinate j has influence at least τ, then at least a fraction $\tau/2$ of Z_k have influence at least $\tau/2$.

Proof. The first statement follows from convexity:

$$\tau = \mathrm{Inf}_j(Z) =\leq \sum_{S|S_j \neq 0} \hat{Z}^2(S) \leq \sum_{S|S_j \neq 0} \mathop{\mathbf{E}}_k[\hat{f}^2(S)] = \mathop{\mathbf{E}}_k[\mathrm{Inf}_j(Z_k)].$$

The second statement follows from an averaging argument. □

Correlated Prob. Space. A product space such as $\mathcal{X} \times \mathcal{Y}$ is said to be a correlated space if it is endowed with a measure Ω. The maximum correlation coefficient of the space is:

$$\rho(\mathcal{X} \times \mathcal{Y}, \Omega) = \max_{Z_1, Z_2} \mathop{\mathbf{E}}_{(x,y) \in \mathcal{X} \times \mathcal{Y}} Z_1(x) Z_2(y) \tag{2.12}$$

where Z_1 and Z_2 are random variables in $\mathcal{X}$ and $\mathcal{Y}$ respectively, and have mean 0 and variance 1 (in their respective marginal measures).

Definition 2.3.3 (Correlation Coeff.)**.** *Let $\mathcal{X}_1 \times \mathcal{X}_2 \times \ldots \times \mathcal{X}_k$ denote a probability space with a measure Ω. The maximum correlation coefficient of this space is:*

$$\rho(\mathcal{X}_1 \times \ldots \times \mathcal{X}_k, \Omega) = \max_{\substack{S,T \subseteq [k] \\ S \cup T = [k]; S \cap T = \emptyset}} \rho(\mathcal{X}_S \times \mathcal{X}_T, \Omega) \tag{2.13}$$

Noise Operator and Low Degree Random Variables. The noise operator of parameter ρ acts on a random variable Z to give:

$$T_\rho(Z) = \sum_S \hat{Z}(S) \cdot \rho^{|S|} \cdot \chi_S \tag{2.14}$$

The truncation operator of order D acts on a random variable Z producing:

$$Z^{(\leq D)} = \sum_{S||S| \leq D} \hat{Z}(S) \cdot \chi_S, \tag{2.15}$$

a D-degree truncation of the random variable Z.

2.3.1 Notion of Influence for orderings

As before, let Ω denote the finite probability space corresponding to the uniform distribution over $[t]$. Let $\{\chi_0 = 1, \chi_1, \chi_2, \ldots, \chi_{t-1}\}$ be an orthonormal basis for the space $L_2(\Omega)$ - of real valued functions over $[m]$ with the inner product $\langle f, g \rangle = \mathbf{E}_{x \in [t]}[f(x)g(x)]$. For $\sigma \in [t]^R$, define $\chi_\sigma(\mathbf{z}) = \prod_{k \in [R]} \chi_{\sigma_i}(z^{(k)})$.

An ordering of $[t]^R$ is a permutation of $[t]^R$, denoted by $\mathcal{O} : [t]^R \to [t]^R$. Such a function can be expressed as a multilinear polynomial as $\mathcal{O}(\mathbf{z}) = \sum_\sigma \hat{\mathcal{O}}(\sigma)\chi_\sigma(\mathbf{z})$. The L_2 norm of $\mathcal{O}$ in terms of the coefficients of the multilinear polynomial is $||\mathcal{O}||_2^2 = \sum_\sigma \hat{\mathcal{O}}^2(\sigma)$

Definition 2.3.4. *For a function* $\mathcal{O} : \Omega^R \to \mathbb{R}$*, define* $\mathrm{Inf}_k(\mathcal{O}) = \mathbf{E}_{\mathbf{z}}[\mathbb{V}_{z^{(k)}}[\mathcal{O}]] = \sum_{\sigma_k \neq 0} \hat{\mathcal{O}}^2(\sigma)$.

Here $\mathbb{V}_{z^{(k)}}[\mathcal{O}]$ *denotes the variance of* $\mathcal{O}(\mathbf{z})$ *over the choice of the* k^{th} *coordinate* $z^{(k)}$.

Definition 2.3.5. *For a function* $\mathcal{O} : \Omega^R \to \mathbb{R}$*, define the function* $T_\rho \mathcal{O}$ *as follows:*

$$T_\rho \mathcal{O}(\mathbf{z}) = \mathbf{E}[\mathcal{O}(\tilde{\mathbf{z}}) \mid \mathbf{z}] = \sum_{\sigma \in [m]^R} \rho^{|\sigma|} \hat{\mathcal{O}}(\sigma) \chi_\sigma(\mathbf{z})$$

where each coordinate $\tilde{z}^{(k)}$ *of* $\tilde{\mathbf{z}} = (\tilde{z}^{(1)}, \ldots, \tilde{z}^{(R)})$ *is equal to* $z^{(k)}$ *with probability* ρ *and with the remaining probability,* $\tilde{z}^{(k)}$ *is a random element from the distribution* Ω.

Lemma 2.3.6. *For every* $\varepsilon > 0$*, there exists a* $\mu_0 > 0$ *such that for all* $\mu < \mu_0$ *the following holds: Let* $\mathcal{O}, \mathcal{Q} : [m]^R \to [0,1]$ *be any two functions with* $\mathbf{E}[\mathcal{O}] = \mathbf{E}[\mathcal{Q}] = \mu$*, and*

$$\mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{O}), \mathrm{Inf}_k(T_{1-\varepsilon}\mathcal{Q}) \leq \tau$$

for all k*. Let* $\mathbf{x}, \mathbf{y}$ *be random vectors in* $[m]^R$ *whose marginal distributions are uniform over* $[m]^R$ *but are arbitrarily correlated. Then,*

$$\mathop{\mathbf{E}}_{\mathbf{x},\mathbf{y}}[T_{1-2\varepsilon}\mathcal{O}(\mathbf{x}) T_{1-2\varepsilon}\mathcal{Q}(\mathbf{y})] \leq \mu^{1+\varepsilon/2} + o_\tau(1)$$

Proof. The lemma essentially follows from the Majority is Stablest theorem (see Theorem 4.4 in [51]). We bound each factor individually as follows:

$$\begin{aligned} ||T_{1-2\varepsilon}\mathcal{O}||_2^2 = \sum_{\sigma \in [k]^R} (1-2\varepsilon)^{2|\sigma|} \hat{\mathcal{O}}^2(\sigma) &\leq \sum_{\sigma \in [k]^R} (1-\varepsilon)^{|\sigma|} \hat{\mathcal{O}}(\sigma) (1-\varepsilon)^{2|\sigma|} \hat{\mathcal{O}}(\sigma) \\ &\leq \mathbf{E}[(T_{1-\varepsilon}\mathcal{O})(\mathbf{x}) T_{1-\varepsilon}(T_{1-\varepsilon}\mathcal{O})(\mathbf{x})]\,. \end{aligned}$$

Since the influences of $\mathrm{T}_{1-\varepsilon}\mathcal{O}$ are low, we can apply Theorem 4.4 from [51] to bound the last expression by noise stability in Gaussian space $\Gamma_{1-\varepsilon}(\mu)$.

$$\mathbf{E}[(\mathrm{T}_{1-\varepsilon}\mathcal{O})\mathrm{T}_{1-\varepsilon}(\mathrm{T}_{1-\varepsilon}\mathcal{O})] \leq \Gamma_{1-\varepsilon}(\mu) + o_\tau(1)$$

By Theorem B.2 from [51], $\Gamma_{1-\varepsilon}(\mu)$ is bounded by $\mu^{1+\varepsilon/2}$ for μ small enough compared to ε. Applying a similar bound for $\mathcal{O}$ and applying Cauchy-Schwartz gives the result:

$$\mathop{\mathbf{E}}_{x}[\mathrm{T}_{1-2\varepsilon}\mathcal{O}(x)\mathrm{T}_{1-2\varepsilon}\mathcal{O}(y)] \leq \sqrt{||\mathrm{T}_{1-2\varepsilon}\mathcal{O}||_2^2||\mathrm{T}_{1-2\varepsilon}\mathcal{O}||_2^2} \leq \mu^{1+\varepsilon/2} + o_\tau(1)$$

(for μ small enough)

□

The following lemma is useful in bounding the number of influential coordinates of a function.

Lemma 2.3.7 (Sum of Influences Lemma)**.** *Given a function* $\mathcal{O} : [m]^R \to [0,1]$*, if* $\mathcal{O} = T_{1-\varepsilon}\mathcal{O}$ *then* $\sum_{k=1}^{R} \mathrm{Inf}_k(\mathcal{O}) \leq \frac{1}{2e\ln 1/(1-\varepsilon)} \leq \frac{1}{\varepsilon}$

Proof. Let $\mathcal{O}(\mathbf{x}) = \sum_\sigma \hat{\mathcal{O}}(\sigma)\chi_\sigma(\mathbf{x})$ denote the multilinear expansion of $\mathcal{O}$. The function $\mathcal{O}$ is given by $\mathcal{O}(\mathbf{x}) = \sum_\sigma (1-\varepsilon)^{|\sigma|}\hat{\mathcal{O}}(\sigma)\chi_\sigma(x)$. Hence we get,

$$\begin{aligned}\sum_{i=1}^{R} \mathrm{Inf}_i(\mathcal{O}) &= \sum_{i=1}^{R}\sum_{\sigma,\sigma_i\neq 0}(1-\varepsilon)^{2|\sigma|}\hat{\mathcal{O}}^2(\sigma) = \sum_\sigma (1-\varepsilon)^{2|\sigma|}|\sigma|\hat{\mathcal{O}}^2(\sigma)\\ &\leq \max_{\sigma\in[m]^R}\Big((1-\varepsilon)^{2|\sigma|}|\sigma|\Big)\cdot\sum_\sigma \hat{\mathcal{O}}(\sigma)^2 \leq \max_\sigma (1-\varepsilon)^{2|\sigma|}|\sigma|\end{aligned}$$

The function $h(x) = x(1-\varepsilon)^{2x}$ achieves a maximum at $x = -1/2\ln(1-\varepsilon)$. Substituting we get $\sum_{i=1}^{R} \mathrm{Inf}_i(\mathcal{O}) \leq \frac{1}{2e\ln 1/(1-\varepsilon)} \leq \frac{1}{\varepsilon}$. □

2.4 Unique Games Conjecture

Definition 2.4.1. *An instance of* UNIQUE GAMES *represented as* $\Upsilon = (\mathcal{A}, \mathcal{B}, E, \Pi, R)$*, consists of a bipartite graph over node sets* $\mathcal{A}$*,* $\mathcal{B}$ *with the edges* E *between them. Also part of the instance is a set of labels* $[R] = \{1, \ldots, R\}$*, and a set of permutations* $\pi_{a\to b} : [R] \to [R]$ *for each edge* $e = (a, b) \in E$*. An assignment* A *of labels to vertices is said to satisfy an edge* $e = (a, b)$*, if* $\pi_{a\to b}(A(a)) = A(b)$*. The objective is to find an assignment* A *of labels that satisfies the maximum number of edges.*

A subset of vertices is said to be *strongly satisfied* by an assignment if every edge incident on the subset is satisfied by the assignment. For sake of convenience, we shall use the following version of the Unique Games Conjecture which was shown to be equivalent to the original conjecture [39].

Conjecture 2.4.2 (Unique Games Conjecture)**.** *For every* $\eta > 0$*, the following problem is* NP*-hard for a sufficiently large choice of* R*: Given a bipartite* UNIQUE GAMES *instance* $\Upsilon = (\mathcal{A}, \mathcal{B}, E, \Pi, R)$ *with number of labels* R*, distinguish between the following two cases:*

- $(1-\eta)$-strongly satisfiable instances: *There exists an assignment* A *of labels such that for* $1-\eta$ *fraction of vertices* $w \in \mathcal{A}_\Upsilon$ *are strongly satisfied, i.e.,* all *the edges* (w, v) *are satisfied.*
- Instances that are not η-satisfiable: *No assignment satisfies more than a* η*-fraction of the edges* E*.*

2.5 Gaussian Stability Estimates

The Gaussian noise stability Γ_ρ is defined as follows:

Definition 2.5.1. *Given $\mu \in [0,1]$, let $t = \Phi^{-1}(\mu)$ where Φ denotes the distribution function of the standard Gaussian. Then,*

$$\Gamma_\rho(\mu) = \mathbf{P}[X \leq t, Y \leq t]$$

where (X,Y) is a two-dimensional Gaussian vector with covariance matrix $\begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}$.

Lemma 2.5.2. *There is constants ε_0, c_1, and c_2 such that, for every $\varepsilon < \varepsilon_0$, and for every $\mu < 1 - c_1\sqrt{\varepsilon}$,*

$$\mu - \Gamma_{1-\varepsilon}(\mu) \geq c_2 \cdot (1-\mu) \cdot \sqrt{\varepsilon}$$

Proof. Let g, h denote independent gaussian random variables with mean 0 and variance 1 and let t be such that $\mathbf{P}[g < t] = \mu$. Then,

$$\begin{aligned}
\mu - \Gamma_{1-\varepsilon}(\mu) &= \underset{X \sim_{1-\varepsilon} Y}{\mathbf{P}}[X < t \wedge Y > t] = \underset{g,h}{\mathbf{P}}[g < t \wedge (1-\varepsilon)g + \sqrt{2\varepsilon - \varepsilon^2}h > t] \\
&= \underset{g,h}{\mathbf{P}}\left[\frac{t - \sqrt{2\varepsilon-\varepsilon^2}h}{1-\varepsilon} < g < t\right] = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-h^2/2} dh \int_{\frac{t-\sqrt{2\varepsilon-\varepsilon^2}h}{1-\varepsilon}}^{t} e^{-g^2/2} dg \\
&\geq \frac{1}{2\pi}\int_{t\sqrt{\frac{\varepsilon}{2-\varepsilon}}}^{\infty} e^{-h^2/2} dh \left[e^{-t^2/2}\left(t - \frac{t - \sqrt{2\varepsilon-\varepsilon^2}h}{1-\varepsilon}\right)\right] \qquad \text{(for small enough } \varepsilon\text{)} \\
&= \frac{e^{-t^2/2}\sqrt{2\varepsilon-\varepsilon^2}}{2\pi(1-\varepsilon)}\left\{\int_{t\sqrt{\frac{\varepsilon}{2-\varepsilon}}}^{\infty} h e^{-h^2/2} dh\right\} - \frac{e^{-t^2/2}(\varepsilon)t}{\sqrt{2\pi}(1-\varepsilon)}\left\{\frac{1}{\sqrt{2\pi}}\int_{t\sqrt{\frac{\varepsilon}{2-\varepsilon}}}^{\infty} e^{-h^2/2} dh\right\} \\
&\geq \frac{e^{-t^2/2}\sqrt{2\varepsilon-\varepsilon^2}}{2\pi(1-\varepsilon)}\left(e^{\frac{t^2(\varepsilon)}{2(2-\varepsilon)}}\right) - \frac{e^{-t^2/2}(\varepsilon)t}{\sqrt{2\pi}(1-\varepsilon)}\left[\frac{1}{2}\right] \\
&= \frac{e^{-\frac{t^2(1-\varepsilon)}{(2-\varepsilon)}}}{2\pi(1-\varepsilon)}\sqrt{2\varepsilon-\varepsilon^2} - \frac{t\varepsilon e^{-t^2/2}}{2\sqrt{2\pi}(1-\varepsilon)} \\
&\geq \frac{(1-\mu)\cdot\sqrt{\varepsilon}}{10} - t^2\theta\varepsilon \geq c_2 \cdot (1-\mu)\cdot\sqrt{\varepsilon} \qquad \left(\frac{t}{1+t^2} \leq \frac{\theta}{e^{-t^2/2}} \leq \frac{1}{t} \leq \frac{1}{10}\right) \square
\end{aligned}$$

Chapter 3

Dictatorship Gadget

In this chapter, we describe the dictatorship test and analyse its properties. The test (and the accompanying gadget) form the core of our inapproximability reductions. Dictatorship gadgets have been widely used and have proven to be extremely powerful in proving tight inapproximability results (both assuming the UGC and otherwise [30]). Almost all known unique games based inapproximability results use this gadget and the study of this gadget has led to important results in analysis [51].

Our contribution is a design of the dictatorship test directly using local distributions guaranteed by the relaxation that is written down for a problem. Recollect the general flavor of the reduction: each vertex in the unique game by a gadget: $[t]^R$. Depending on the problem in hand, a distribution over k-tuples of points in $[t]^R$ is sampled from and a constraint spanning the gadgets of multiple vertices is added. The *dictatorship test* is a distribution over k-tuples of elements in $[t]^R$ which is used to design the final distribution over different cubes; the final distribution is obtained by a somewhat standard procedure from the dictatorship test.

The overall output of the reduction, including the interplay between the gadgets corresponding to different vertices and the implications to inapproximability of various problems arising from the use of the gadget is described in the chapters corresponding to the problem in hand in their respective chapters.

Depending on the problem, we have a payoff (or a cost) function on assignments to the cube. The analysis of the gadget involves studying the payoff of certain canonical assignments known as the *dictator assignment*, compared to assignments "far" from being like a dictator assignment. The payoff is also part of the specification, when designing the gadget, and goes hand-in-hand with the distribution used to design the gadget. In particular, the payoff directly encodes either the objective function of the relaxation, or in some cases, constraints set by the problem in hand.

3.1 Construction of the Gadget

Our dictatorship tests have the following simple form: it is obtained by a direct-product of a distribution obtained from the local constraints enforced by the relaxation.

Definition 3.1.1. *Given a positive integer, k denoting the* arity *of the constraints; a positive integer, t denoting the* alphabet size*; a positive integer R and a probability distribution $\mathcal{D}$ over $[t]^k$, the gadget is the following distribution $\mathsf{Dict}^R(\mathcal{D})$ over k-tuples of elements from $[t]^R$:*

- *Pick R independent samples $(z_{i1}, z_{i2}, \ldots, z_{ik})$ from $\mathcal{D}$; $1 \leq i \leq R$.*

- *Output $((z_{11}, z_{21}, \ldots, z_{R1}), (z_{12}, z_{22}, \ldots, z_{R2}), \ldots, (z_{1k}, z_{2k}, \ldots, z_{Rk}))$.*

Canonical Dictator Assignments. An assignment or labeling of the gadget is a function $f : [t]^R \to \blacktriangle_{[t]}$. The assginment is said to be *proper*, if f maps every point in its domain to $[t]$ (as opposed to a probability distribution over $[t]$).

The gadget is designed such that certain canonical assignments of labels, known as the dictator assignments are preferred over others. There is one dictator assignment per coordinate; the dictator "along" coordinate i is the function f_i defined as:

$$f_i(\mathbf{z}) = z_i.$$

3.1.1 Payoff Functions

As mentioned before, the analysis of the gadget involves studying the performance of a dictator assignment (along any coordinate), comparing it with assignments far from a dictator. The performance is measured by a payoff function (also referred to as a cost function depending on the context.)

Definition 3.1.2. *A payoff (or a cost) function of degree k is a function, $p : [t]^k \to [-1, 1]$, used to measure the performance of an assignment f on the gadget. The payoff of a proper assignment f is defined to be:*

$$\mathcal{P}(f) = \mathbf{E}[p(f(\mathbf{z}_1), f(\mathbf{z}_2), \dots f(\mathbf{z}_k))].$$

Our application will require us to work on improper assignments (arising from averaging over a bunch of proper assignments). The payoff or cost of a (general) assignment: $f : [t]^R \to \blacktriangle_{[t]}$ is:

$$\mathcal{P}(f) = \mathop{\mathbf{E}}_{\mathbf{z}} \sum_{\mathbf{x} \in [t]^k} p(\mathbf{x}) \prod_j f(\mathbf{z}_j)_{x_j} \tag{3.1}$$

Note that the general definition reduces to the simpler one when f is a proper assignment.

Since $\mathsf{Dict}(\cdot)$ is simply a direct-product of the input distribution, the assignment of any dictator assignment is simply a function of the input distribution.

Lemma 3.1.3. *For every k, R, $\mathcal{D}$, $\mathcal{P}$ as above, and any index i, if f_i denotes the dictator along coordinate i, then the payoff of f_i on $\mathsf{Dict}^R(\mathcal{D})$ is:*

$$\mathcal{P}(f_i) = \mathop{\mathbf{E}}_{\mathbf{x}\in\mathcal{D}}[p(\mathbf{x})]$$

Proof.

$$\mathcal{P}(f_i) = \mathop{\mathbf{E}}_{\mathsf{Dict}(\mathcal{D})}[p(f_i(\mathbf{z}_1), f_i(\mathbf{z}_2), \ldots f_i(\mathbf{z}_k))] = \mathbf{E}[p(z_{1i}, z_{2i}, \ldots, z_{ki})] = \mathop{\mathbf{E}}_{\mathbf{x}\in\mathcal{D}}[p(\mathbf{x})]\,.$$

$\square$

3.2 Analysis of Gadget Assignments

The bulk of the analysis of the gadget involves assignments that are far from being like the dictator assignments described above. The main tool used here is the *invariance principle* (introduced in [51, 50]), a powerful method used in analysing low-degree function defined over a probability space.

3.2.1 Invariance Principle

Invariance principle, introduced in [51, 50] is a tool to analyse low degree functions on a probability space. A simple invariance theorem is the following restatement of Berry-Esseen's theorem (see 16.5 in [23]).

Theorem 3.2.1 (Berry-Esseen's theorem)**.** *Let $x_1, x_2, \ldots x_n$ be i.i.d. bernoulli rand. variables (i.e., ± 1 with equal probability), and let $x = (x_1 + x_2 + \ldots + x_n)/\sqrt{n}$. Similarily, let $g_1, g_2, \ldots g_n$ be i.i.d standard gaussian variables (i.e. gaussian: mean 0, variance 1), and let $g = (g_1 + g_2 + \ldots g_n)/\sqrt{n}$. Then, the total variational distance between the distribution of x*

and g is bounded as follows:

$$\|x - g\|_{TV} \leq O(1/\sqrt{n}) \tag{3.2}$$

The theorem shows that the (linear) formal polynomial $(x_1+x_2+\ldots+x_n)/\sqrt{n}$ is invariant under the two different ensemble of random variables. Invariance principle gives a general method to obtain such bounds on any low-degree polynomial. Our application of the principle is on the payoff of an assignment to the dictatorship gadget. Consider the 3-WAY CUT problem introduced in chapter 1. The gadget used for reductions to this problem is $[3]^R$ for some large R. Let f be an assignment to the cut problem; $f : [3]^R \to [3]$. Suppose we are interested in the probability that $f(x) = f(y)$ when x is picked uniformly at random and y is obtained by randomly picking εR coordinates of x and replacing the entries there with a independent uniform entry (i.e., $y_j \in_R [3]$ if j is among the coordinates picked). Such a quantity arises in the analysis of the cost of an assignment (the edges are picked to match the weights from the above distribution).

Note that the problem attempts to minimize the size of a cut, and therefore the above probability is to be minimized in picking an assignment. For example, the dictator assignment (along any coordinate) "cuts" with probability $O(\varepsilon)$. As mentioned in the overview, our reduction prohibits such assignments from occurring (on an average) when the UG instance is almost completely unsatisfiable.

Using the invariance principle, one can show that any non-influential assignment which splits the gadget into 3 equal components cuts with probability at least $\sqrt{\varepsilon}$. Let $f = (f_1, f_2, f_3)$ be an assignment (where f_i is the indicator of f assining i). The probability an edge is cut is $\mathbf{P}_{(x,y)}[\sum_{i \neq j} f_i(x) f_j(y)]$. Consider a single term, $\mathbf{P}_{x,y}[f_1(x) f_2(y)]$. Now, f_1 (and similarily f_2) can be written in a orthnormal basis of $[3]^R$ endowed with the uniform measure as:

$$f_1(x) = \sum_{S \in [3]^R} \hat{f}_1(S) \chi_S(x) \tag{3.3}$$

and suppose the degree of this expansion is at most D.

Let $\{G_S\}_S$ denote a gaussian ensemble with matching moments as $\{\chi_S\}_S$. and let $\tilde{f}_1(x) = \sum_S \hat{f}_1(S)\mathcal{J}_S(x)$. Now, the invariance principle says that if for a parameter τ, every coordinate in f_1 are at most τ, then

$$\|\tilde{f}_1 - f_1\|_{\mathrm{TV}} \leq \delta(\tau, D)$$

where $\|\cdot\|_{\mathrm{TV}}$ denotes total variational distance and $\delta(\tau, D)$ is a quantity that tends to 0 as $\tau \to 0$ or if $D \to \infty$. Further, $\mathbf{E}[f_1 f_2]$ is equal to $\mathbf{E}[\tilde{f}_1 \tilde{f}_2]$ since the two ensembles have matching moments of degree upto 2. This allows one to equate the minimum cut value to the best partitions in gaussian space, where estimates are known about such isoperimetric questions (see section 3.2.3 for a complete analysis of such assignments.)

3.2.2 Low Degree Assignments

In the above discussion, the f had to be a low-degree polynomial (that is $<< R$, the dimension of the gadget) which is not necessarily true of an arbitrary assignment. However, as we show below, under certain mild conditions, the assignment can always be approximated by one of low-degree. Fortunately, our applications satisfy these constraints, hence allowing this approximation while only losing a small arbitrary constant in the final guarantees of the reduction.

Lemma 3.2.2. *For every $\varepsilon > 0$, $\rho < 1$, and integer k, if $p : [t]^k \to [-1, 1]$ is a degree-k payoff function, and $\mathcal{D}$ is a distribution on $[t]^k$ whose max-correlation coefficient is at most ρ, then, for every assignment f on $[t]^R$ for the dictatorship test $\mathsf{Dict}^R(\mathcal{D})$, there exists an assignment f' of degree at most $c_1 \cdot \frac{k \log^2(1/\varepsilon)}{\varepsilon(1-\rho)}$ such that,*

$$\left|\mathcal{P}_{\mathsf{Dict}^R(\mathcal{D})}(f) - \mathcal{P}_{\mathsf{Dict}^R(\mathcal{D})}(f')\right| \leq \varepsilon$$

Proof. Let $f^j(\mathbf{z}) = 1$ if $f(\mathbf{z}) = j$ and 0 otherwise.

$$\mathcal{P}_{\mathsf{Dict}^R(\mathcal{D})}(f) = \sum_{(j_1,j_2,\dots j_k)\in[t]^k} p(j_1, j_2, \dots j_k) \mathop{\mathbf{E}}_{\mathbf{z}}[f^{j_1}(\mathbf{z}_1) f^{j_2}(\mathbf{z}_2) \dots f^{j_k}(\mathbf{z}_k)] \tag{3.4}$$

$$= \sum_{(j_1,j_2,\dots j_k)} p(j_1, j_2, \dots j_k) \sum_{S_1,S_2,\dots S_k\in[t]^k} \hat{f^{j_1}}(S_1) \dots \hat{f^{j_k}}(S_k) \mathop{\mathbf{E}}_{\mathbf{z}}[\chi_{S_1}(\mathbf{z}_1)\chi_{S_2}(\mathbf{z}_2) \dots \chi_{S_k}(\mathbf{z}_k) \tag{3.5}$$

Each expectation in eq. (3.5) is at most ρ^D if any of the sets S_j is of size D. Thus, the total contribution of such terms is at most ε if $D \geq \Omega({}^{k\log^2(1/\varepsilon)}/_{\varepsilon(1-\rho)})$. Setting $f' = f^{(\leq D)}$ shows the result. ☐

3.2.3 Metric Like Payoffs on Noise Distributions

A simple, but already useful setup is when the cost function is *metric-like* and the distribution is a very lazy random walk on a probability space. We use this setup in the analysis of our reductions to metric labeling problems (see chapter 4.)

Definition 3.2.3. *A function, $p : [t]^k \to [-1, 1]$, is said to be* metric-like *if it is non-negative and:*

$$p(\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_k) = 0 \Leftrightarrow \mathbf{x}_1 = \mathbf{x}_2 = \dots = \mathbf{x}_k.$$

The interesting parameter of the cost function is the minimum non-zero cost assigned:

$$\beta = \min_{\mathbf{x}_1,\mathbf{x}_2,\dots\mathbf{x}_k} p(\mathbf{x}_1, \mathbf{x}_2, \dots \mathbf{x}_k).$$

Note that β fixes the *aspect ratio* of the cost function since the range is already bounded by 1 in magnitude.

Definition 3.2.4. *For a parameter ε, given a measure Ω on $[t]$, the lazy random walk distribution $\mathcal{D}^{(\Omega)}_{1-\varepsilon}$ (also known as the noise operator distribution) over $[t]^k$ is obtained by:*

– *sampling x from Ω*

– *for each* i *from* 1 *to* k*, set* $z_i = x$ *with probabilty* $1-\varepsilon$ *and to a new sample* y_i *from* Ω*; this choice is made independent of every other choice, for each* i*.*

– *output* $\mathbf{z} = (z_1, z_2, \ldots, z_k)$

Consider the dictatorship test constructed using the above distribution, $\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})$. A dictator assignment has a cost of at most $k \cdot \varepsilon$ since the cost is bounded by 1 and the distribution outputs the same points with probability $1 - k\varepsilon$. However, as we will see next, the cost of assignments far from dictator is significantly larger.

Theorem 3.2.5. *Let* Ω *be a measure on* $[t]$ *such that the minimum probability of any atom is* $\alpha \in (0,1)$ *and let* $\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})$ *be the dictatorship test generated from the noise operator distribution of* Ω*. For an assignment* f*, let* $\mu_j = \mathbf{E}_{\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})}[f(\mathbf{z}_i) = j]$ *and let* $\mu = \max_j \mu_j$*. Then, for there is an* ε_0*,* c_1 *and* c_2 *such that for all* $\varepsilon < \varepsilon_0$*,* $\alpha, \beta > 0$ *and integers* k, D *there exist a* τ *such that if* f *is a degree* D *assignment such that every coordinate has influence at most* τ *and* $\mu \leq 1 - c_1\sqrt{\varepsilon}$*; and if* p *is a* k*-degree metric-like cost function whose minimum non-zero cost is atleat* β*, then the cost of* f *on* $\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})$*:*

$$\mathcal{P}(f) \geq 2\beta c_2 \sqrt{\varepsilon} \cdot (1 - \mu)$$

Proof. First, note that μ_j are well defined since the marginal distribution of each $\mathbf{z}_i$ is the same. For $1 \leq j \leq t$, let f^j be the indicator of the set of points where f assigns j. The gadget $\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})$ is a distribution over k tuples of elements in $[t]^R$. Fix two indices i_1, i_2 in $\{1, \ldots k\}$. Now, using the invariance principle, the probability that $f(\mathbf{z}_{i_1}) = f(\mathbf{z}_{i_2})$ is:

$$\sum_j \mathbf{E}\left[f^j(\mathbf{z}_{i_1}) = f^j(\mathbf{z}_{i_2}) = 1\right] = \mathbf{E}\left[f^j \cdot T_{1-\varepsilon} f^j\right] \leq \Gamma_{1-\varepsilon}(\mu_j) + o_{\tau,D}(1)$$

where the final term, $o_{\tau,D}(1)$ is an expression that decays to 0 as $\tau \to 0$ and $D \to \infty$.

The payoff of the labeling is then,

$$\begin{aligned}\mathcal{P}(f) &\geq \beta \cdot \underset{(\mathbf{z}_1,\mathbf{z}_2,\ldots\mathbf{z}_k)\in\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})}{\mathbf{P}}\left[f(\mathbf{z}_{i_1}) \neq f(\mathbf{z}_{i_2})\right] \\ &\geq \beta \sum_j \left[\mu_j - \Gamma_{1-\varepsilon}(\mu_j)\right]\end{aligned}$$

The function Γ is superadditive and hence the minimum value the above expression attains is when one of the μ_j is large. Now, using lemma 2.5.2, we know that the last expression is at least $\Omega((1-\mu_j)\cdot\sqrt{\varepsilon})$ if ε is small enough. Now, picking τ and D appropriately gives the required result. $\square$

From lemma 3.2.2, the constraint on the degree of f can be removed if the influence is measured on the low-degree truncation of f.

Corollary 3.2.6. *For ε, μ, k, α, β and p as described above (theorem 3.2.5), there exist τ and D such that if an assignment f is such that $f^{(\leq D)}$ has influences at most τ, then the cost of f on $\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})$ is:*

$$\mathcal{P}_{\mathsf{Dict}(\mathcal{D}^{(\Omega)}_{1-\varepsilon})}(f) \geq 4\beta c_2\sqrt{\varepsilon}\cdot(1-\mu)$$

3.2.4 Connected Payoff Functions

The analysis of assignments under a metric-like cost can be generalized as follows. Suppose $([t]^k, \mathcal{D})$ is a correlated space whose correlation coefficient, $\rho(\mathcal{D})$ is strictly smaller than 1. Let p be a cost function that assigns no cost to points in the support of $\mathcal{D}$.

A dictator assignment to $\mathsf{Dict}^R(\mathcal{D})$ would cost nothing at all. However, assignments with low influence and low degree have a strictly positive cost unless no significant fraction takes that value that leads to a positive cost.

This forms the basis of the reduction to STRICT CSP in chapter 5. These problems have a set of strict constraints which need to satisfied in any solution. For these problems, we construct a dictatorship test from distributions arising from a "local" relaxation. Here, assignments far from being dictator-like can be shown to be almost a constant assignment unless they violate a few constraints.

Theorem 3.2.7. *For every integers k, D, numbers $\rho < 1$, $0 < \alpha < 1/2$, and every $\mu > 0$, there is a $\delta > 0$ such that, if $\mathcal{D}$ is a distribution with max. correlation coefficient at most ρ, f a D-degree labeling of $\mathsf{Dict}^R(\mathcal{D})$ and $\mathbf{x} \in [t]^k$ is a pattern such that if:*

$$\min_j \mathop{\mathbf{P}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} [f(z_j) = x_j] \geq \mu, \tag{3.6}$$

then the probability that

$$\mathop{\mathbf{P}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} [\wedge_i f(z_i) = x_i] \geq \delta \tag{3.7}$$

Proof. As before, let f_j denote the indicator of the assignment f taking the value j in $[t]$. The probability f takes the patter $\mathbf{x}$ is:

$$\mathop{\mathbf{P}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} [f(\mathbf{z}_1) = x_1 \wedge f(\mathbf{z}_2) = x_2 \wedge \ldots \wedge f(\mathbf{z}_k) = x_k] = \mathop{\mathbf{P}}_{\mathbf{z}}[\Pi_i f_{x_i}(\mathbf{z}_i)]. \tag{3.8}$$

Since f is of degree D, the components f_j are too; hence, applying the invariance principle, we have that

$$\mathop{\mathbf{P}}_{\mathbf{z}}[\Pi f_{x_i}(\mathbf{z}_i)] \geq \min_{\{S_1, S_2, \ldots S_t\}} \mathop{\mathbf{P}}_{\mathbf{g}}[\wedge_i \{g_i \in S_{x_i}\}] - \delta_{\tau,D}(1) \tag{3.9}$$

where S_j are partitions of $\mathbb{R}^t$ such that $\mu(S_j) = \mu(f_j) = \mathbf{P}[f_j(\mathbf{z}) = 1]$; g_i are the analogously correlated gaussian random variables; and $\delta_{\tau,D}$ is a quantity that tends to 0 as $D \to \infty$ or $\tau \to 0$. Although the exact partition that minimizes the expression in eq. (3.9), the minimum can be lower bounded by a function $\Delta(\rho, \alpha, \mu)$. Picking τ small enough proves the theorem. $\square$

As before, we prove a version where f is not restricted to be of low-degree (using lemma 3.2.2); the influences are measured for a truncation of f.

Corollary 3.2.8. *For every k, ρ, α and μ, there is a $\delta > 0$, τ and D such that, if $\mathcal{D}$ is a distribution with max. correlation coefficient at most ρ, f a labeling for $\mathsf{Dict}^R(\mathcal{D})$ such that $\mathrm{Inf}_j f^{(\leq D)} \leq \tau$ for every coordinate j and $\mathbf{x} \in [t]^k$ is a pattern such that if:*

$$\min_j \mathop{\mathbf{P}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} [f(\mathbf{z}_j) = x_j] \geq \mu, \tag{3.10}$$

then the probability that

$$\mathop{\mathbf{P}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} [\wedge_i f(\mathbf{z}_i) = x_i] \geq \delta \tag{3.11}$$

3.2.5 Alphabet Reduction

Semidefinite relaxations are widely used in the design of approximation algorithms for constraint satisfaction problems (CSPs). The seminal work of Raghavendra [56], shows that a semidefinite program gives the best approximation assuming the Unique Games Conjecture.

An instance of a CSP of arity k and alphabet size t is $\mathcal{J} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a collection of ordered k-tuples $(v_1, \ldots v_k)$ along with constraint / payoff: $p : [t]^k \to \mathbb{R}$. The goal is to find an assignment $L : \mathcal{V} \to [t]$ that maximizes:

$$\sum_{(v_1, \ldots v_k) \in \mathcal{E}} p(L_1(v_1), L_2(v_2), \ldots L_k(v_k)) \tag{3.12}$$

The following theorem is a restatement of the inapproximability proven in [56].

Theorem 3.2.9 (Theorem 7.1 in [56])**.** *For every CSP $\mathcal{J}$, there is a SDP relaxation such that, for every $\eta > 0$, it is hard (assuming the UGC) to distinguish instances of the CSP, $\mathcal{I}$ such that $\mathrm{OPT}(\mathcal{I}) \geq \mathrm{SDP}(\mathcal{J}) - \eta$ from instances with $\mathrm{OPT}(\mathcal{I}) \leq \mathrm{OPT}(\mathcal{J}) + \eta$.*

In the theorem above, it is necessary that the $\mathcal{J}$ is hard-coded in the algorithm performing the reduction: the running time is only fixed-parameter tractable in the parameters describing $\mathcal{J}$. In particular, the running time is exponential in the label or alphabet size of the CSP, t.

On the other hand, ordering problems are examples of CSPs if the label size is allowed to depend on the total number of points. Here, we show that such extended CSPs (CSPs in the sense described in [56] have a constant alphabet size), can be analysed in an interesting manner.

An ordering of $[t]^R$ is a permutation of $[t]^R$, denoted by $\mathcal{O} : [t]^R \to [t]^R$. (Aside: the notion of influence for ordering is not intuitive; we refer the readers to section 2.3.1) for a quick reference of the definition and theorems on this notion). We show that when the ordering is not influential, it might as well be approximated by the t-bucket version described below. On the other, when the distribution $\mathcal{D}$ is itself from an ordering problem, dictator orderings have a better payoff.

Payoffs for orderings are defined as follows. Let $\Pi_{k\to\mathbb{N}}$ denote the set of one to one maps from $[k] \to \mathbb{N}$. The domain of a payoff function P can be extended naturally from the set of permutations Π_k to $\Pi_{k\to\mathbb{N}}$. In particular, an injective map $f \in \Pi_{k\to\mathbb{N}}$, along with the standard ordering on the range $\mathbb{N}$ induces a permutation π_f on $[k]$. To extend the payoff, just define $p(f) = p(\pi_f)$ for all $f \in \Pi_{k\to\mathbb{N}}$.

To explain this method, consider the following *bucket version* of an ordering problem. Given an instance $\mathcal{J}$ of Ordering CSP, and a parameter t, the t-bucket version of $\mathcal{J}$ asks for an assignment $\sigma_t : \mathcal{V} \to [t]$. The payoff of a labeling σ_t is the expected payoff of a random ordering that is consistent with σ_t in the following sense: for every v_1 and v_2 such that $\sigma_t(v_1) > \sigma_t(v_2)$, v_1 is placed after v_2 in the ordering. In other words, the random ordering is picked by randomly permuting the set of vertices assigned the same value in the bucket version, while preserving the ordering between such sets. Let $\mathrm{OPT}_t(\mathcal{J})$ denote the optimum of the t-bucket version. At the one end, setting $t = \infty$ gives us the original problem; while setting $t = 1$ gives a trivial instance where the optimum is exactly the payoff of a purely random ordering. Let p_t denote the payoff of the (proxy) t-CSP.

The key theorem is that if an ordering does not have influential variables (say, $\geq \tau$), then,

$$\mathcal{P}_{\mathsf{Dict}^R(\mathcal{D})}(\mathcal{O}) = \mathop{\mathbf{E}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} p(\mathcal{O}(\mathbf{z}_1), \ldots, \mathcal{O}(\mathbf{z}_k)) \tag{3.13}$$

$$\leq \mathop{\mathbf{E}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} p_t(\mathcal{O}(\mathbf{z}_1), \ldots, \mathcal{O}(\mathbf{z}_k)) + \theta \tag{3.14}$$

$$= \mathcal{P}^{(t)}_{\mathsf{Dict}^R(\mathcal{D})}(\mathcal{O}) + \theta \tag{3.15}$$

where $\theta \to 0$ as $t \to \infty$ or as $\tau \to 0$.

Theorem 3.2.10. *For every θ, there exists τ, D and t such that if $\mathcal{O}$ is an ordering such that if $\inf_j(\mathcal{O}^{(\leq D)}) \leq \tau$ for all j, then*

$$\mathcal{P}_{\mathsf{Dict}^R(\mathcal{D})}(\mathcal{O}) \leq \mathcal{P}^{(t)}_{\mathsf{Dict}^R(\mathcal{D})}(\mathcal{O}) + \theta$$

Proof. Given an ordering $\mathcal{O}$, construct $\mathcal{O}_t$ by bucketing into t (almost) equal pieces. Fix $i_1, i_2 \in [k]$. Since $\rho < 1$, applying lemma 2.3.6 with $\mu = 1/t$ gives that:

$$\mathop{\mathbf{P}}_{\mathbf{z} \in \mathsf{Dict}^R(\mathcal{D})} \{\mathcal{O}_t(\mathbf{z}_{i_1}) = \mathcal{O}_t(\mathbf{z}_{i_2}) + t^{-\varepsilon} + o_{\tau,D}(1)\} \tag{3.16}$$

Thus, the difference between $\mathcal{P}$ and $\mathcal{P}^{(t)}$ is at most $\binom{k}{2} t^{-\varepsilon}$. Setting t large enough shows that $\mathcal{P}$ and $\mathcal{P}^{(t)}$ differ by at most $\theta/2$, while setting τ small enough makes the second term smaller than $\theta/2$. $\square$

3.3 Overview of a reduction from UG

In the next chapters, we will show inapproximability of various class of problems using the dictatorship test we construct here. All the reductions involved transform a given UG instance by replacing each right hand vertex by the gadget, $[t]^R$. The key insight is that one can use dictator assignments along coordinates corresponding to the labels assigned to the vertices. This leads to aligned coordinates in most neighboring pair of vertices.

A dictatorship test is constructed out of a "input" distribution; a random pair (or generally, a random k tuple) having a common left neighbor is picked and the test is used to sample vertices from gadgets corresponding to the vertices picked. However, the order of the labels for each of these gadgets is also permuted by the permutation specified on the edge. Now, the dictator assignments inspired by the labeling of the instance gives a global labeling of all the gadgets such that most dictatorship tests see a dictator labeling. This is the reason we have a higher payoff (or a lower cost as the case maybe) when the UG instance is almost completely satisfiable. This higher payoff is often proportional to the objective of a relaxation.

In fact, a stronger converse can also be shown: if the payoff of some assignment is larger than a threshold, one can find a small set of "influential" coordinates for each vertex such that for a good fraction of the edges, some pair in the corresponding set of influential coordinates satisfy the constraint on the edge. This suggests a "list decoding" algorithm that samples one of these coordinates at random and assigns to the vertex. The expected payoff of this algorithm is then proven to be significant (at least a constant away from 0). Assuming the UGC, this is indistinguishable from when the instance is almost fully satisfiable. The argument is completed by showing that this threshold is indeed a integer optimum of the problem.

Chapter 4

Approximability of Labeling Problems

In this chapter, we will focus on the approximability of a fairly general class of labeling problems known as Metric Labeling. Using this framework developed in the previous chapters, we show that a simple linear relaxation already captures the approximability of these problems (and certain interesting special cases of them). The relaxation has been previously studied in designing approximation algorithms for these problems. We show that a gap instance of this relaxation implies an inapproximability of the integrality gap ratio upto an arbitrarily small additive constant.

Roadmap. We begin my describing metric labeling problems and their special cases, citing previous work in the design of approximation algorithms. We follow this up with the description of the relaxation, followed by our reduction and its analysis. Finally, we describe a simple rounding scheme for the relaxation and show its optimality. Most of the work presented in this chapter is from joint work with Joseph (Seffi) Naor, Prasad Raghavendra, and Roy Schwartz[48].

4.1 Description of Problems

Metric labeling problems fall under the class of edge deletion problems along with many other classic optimization problems. In an edge deletion problem, given an undirected graph $G = (V, E)$ and weights w on E, the goal is to find a minimum weight set of edges E' such that $G' = (V, E - E')$ satisfies certain properties. A special case is when the set of deleted edges form a *cut*. For example, the simplest and probably most familiar problem in this class is the minimum (s, t) cut problem. Given two terminals s and t the goal is to find a minimum weight cut that separates s and t, i.e., s and t belong to different connected components as a result of removing the cut from the graph. This problem can be solved precisely in polynomial time following the classic work of Ford and Fulkerson. METRIC LABELING is a more general edge deletion problem, with applications in computer vision, network design and clustering, to name a few ([41]).

We will describe a few special metric labeling problems, which are interesting in their own right, finally leading upto the general METRIC LABELING. Refer fig. 1.1 for a quick overview of our results for the problem and its special cases.

Multiway Cut. The MULTIWAY CUT problem is a natural generalization of the minimum (s, t) cut problem when more than two terminals are involved. An instance of MULTIWAY CUT is a graph $\mathcal{J} = (\mathcal{V}, \mathcal{E})$ and a set $\mathcal{X} \subseteq \mathcal{V}$ of terminals. The graph can be weighted, in which case, a probability distribution over the edges, $\Omega_{\mathcal{E}}$ is part of the specification. The problem asks for a minimum collection of edges separating *every* pair of terminals. This problem is NP-hard and a $(2 - \frac{2}{t})$-approximation algorithm that uses the classic min-cut algorithm as a subroutine is known [17] (here, $t = |\mathcal{X}|$). Based on a novel geometric relaxation, which was later generalized to the relaxation we work with, Calinescu, Karloff and Rabani [16] obtained a $\frac{3}{2} - \frac{1}{t}$ approximation for the problem. Continuing this line of work, Karger et al[33] obtained tight integrality gaps when the instance has exactly 3 terminals., and improved approximation factors for the general case. However, nothing better than APX-hardness [17] was known prior to this work.

0-Extension. The 0-Extension problem, introduced in [36, 37] is a generalization of Multiway Cut in which a metric d is defined on the terminal set $\mathcal{X}$. The goal is to assign to each vertex $v \in \mathcal{V}$ a terminal $x(v)$ in $\mathcal{X}$, while minimizing the total cost given by $\sum_{w,v \in \mathcal{E}} w(w,v)d(x(w),x(v))$. When the metric assigns a distance, 1 to all distinct terminal pairs (and 0 otherwise), the problem reduces to the Multiway Cut described above. Calinescu et al [10] obtained an $O(\log|\mathcal{X}|)$-approximation algorithm for 0-Extension. With a better analysis, the guarantee was improved to $O(\log|\mathcal{X}|/\log\log|\mathcal{X}|)$ in [20]. The ideas from the 0-Extension problem [20, 10] have found applications in metric embedding [42] and analysis [45]. Building on the work of [15], [34] showed that there is no polynomial time algorithm that approximates 0-Extension within a factor of $O((\log|\mathcal{X}|)^{\frac{1}{4}-\varepsilon})$, unless $NP \subseteq DTIME(n^{poly(\log n)})$

Metric Labeling. Motivated by applications in computer vision, Kleinberg and Tardos [41] introduced the Metric Labeling problem. An instance is a (weighted) graph $\mathcal{J} = (\mathcal{V}, \mathcal{E}, \Omega_{\mathcal{E}})$, a metric d on a set of labels, $\mathcal{X}$, and a non-negative assignment cost function, C, on vertex-label pairs. The objective is to find an assignment L of labels to the vertices while minimizing:

$$\sum_{v \in \mathcal{V}} C(v, x(v)) + \mathop{\mathbf{E}}_{(w,v)\in\Omega_{\mathcal{E}}} d(t(w), t(v)).$$

It is known that the cost function can be assumed to assign only 0 or ∞ to its domain without loss of generality. We will denote the cost by a function $C : w \to 2^{\mathcal{X}}$.

If the cost function is 0 for all vertices except for a selected bunch of terminals, where the cost is 0 when the terminals are assigned a pre-determined label and ∞ otherwise, the instance turns into an instance of the aforementioned 0-Extension problem.

Using an approximation of the metric as a combination of dominating tree metrics [8, 41], gave an approximation algorithm for Metric Labeling. Its approximation factor can be shown to be $O(\log|\mathcal{X}|)$ using the later improvement of [21] in embedding metrics into dominating tree metrics. On the other hand, Chuzhoy and Naor [15] showed that there is no polynomial time approximation, better than $O(\log^{1/2-\varepsilon}|\mathcal{X}|)$ unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\text{poly}\log n})$.

Another important special case is the Uniform Metric Labeling, where the metric is the uniform metric (assigning 1 to distinct labels). A 2-factor approximation algorithm[41] is known for Uniform Metric Labeling. Constant factor approximation algorithms [41, 25, 13, 1] are known for many other special cases of metrics.

4.2 Earthmover Relaxation

Now, we describe the linear relaxation used to tailor the inapproximability. Note that our goal in this chapter is to show that the particular linear relaxation provides the best approximation assuming the UGC.

Earthmover Metric. Given a metric d on a set of points, $\mathcal{X}$, the *earthmover* metric corresponding to d, denoted by $d_{\bowtie}$, specifies distances between *fractional points* from $\mathcal{X}$ (that is, points in $\blacktriangle_{\mathcal{X}}$).

Definition 4.2.1 (Earthmover Distance)**.** *Given a metric $d : \mathcal{X} \times \mathcal{X} \to R$, and two points $x, y \in \blacktriangle_{\mathcal{X}}$, the earthmover distance between x and y is:*

$$\begin{aligned} d_{\bowtie}(x,y) = &\textit{minimize} && \sum_{i,j} d(i,j) f_{ij} && \\ &\textit{subject to} && \sum_i f_{ij} = y_j; \quad \sum_j f_{ij} = x_i && \forall i,j \in \mathcal{X} \\ & && 0 \le f_{ij} \le 1 && \forall i,j \in \mathcal{X} \end{aligned}$$

In words, the earthmover distance is the minimum cost of moving the probability mass from distribution x to y, given the distance metric d on the labels. It is easy to see that this defines a metric on the simplex of the label set of the metric, $\blacktriangle_{\mathcal{X}}$. The distance is measured by optimizing a "flow" from the source point to the destination point in the simplex. Note that the distance can be calculated in a linear relaxation. We will use this fact in embedding this metric in the linear relaxation for Metric Labeling problems.

We now prove a simple and yet important property of the earthmover metric: given two points $x, y \in \blacktriangle_{\mathcal{X}}$, there is a distribution over $\mathcal{X} \times \mathcal{X}$ such that the expected distance is exactly the earthmover distance.

Theorem 4.2.2. *For every metric $d : \mathcal{X} \times \mathcal{X}$, and every two points $\mathbf{x}, \mathbf{y} \in \blacktriangle_{\mathcal{X}}$, there is a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{X}$ such that:*

$$\mathop{\mathbf{E}}_{(z_1,z_2)\in\mathcal{D}}[d(z_1, z_2)] = d_{\bowtie}(\mathbf{x}, \mathbf{y})$$

and for every $z \in \mathcal{X}$,

$$\mathop{\mathbf{P}}_{(z_1,z_2)\in\mathcal{D}}[z_1 = z] = (\mathbf{x})_z \qquad \mathop{\mathbf{P}}_{(z_1,z_2)\in\mathcal{D}}[z_2 = z] = (\mathbf{y})_z$$

Linear Relaxations for Metric Labeling Problems. As described above, the Metric Labeling is the most general case. We now describe the relaxation for an instance of Metric Labeling. Given an instance, $\mathcal{J} = (\mathcal{V}, \mathcal{E}, \Omega_{\mathcal{E}})$; a metric d on $\mathcal{X}$ and assignment constraints $C : \mathcal{V} \to 2^{\mathcal{X}}$, the relaxation asks to:

$$\begin{aligned} &\text{minimize} && \mathrm{LP}(\mathcal{J}) = \mathop{\mathbf{E}}_{(v,w)\in\Omega_{\mathcal{E}}} d_{\bowtie}(Y_v, Y_w) \\ &\text{such that} && Y_v \in \blacktriangle_{C(v)} \qquad \forall v \in \mathcal{V} \end{aligned} \tag{EMLP}$$

Figure 4.1: Earthmover Relaxation for Metric Labeling

Structured Integrality Gap. The inapproximability of METRIC LABELING problems involves a reduction from UNIQUE GAMES and a integrality gap instance of the above relaxation to another instance of METRIC LABELING. We will need minor "pre-processing" of the integrality gap instance for our reduction; we denote such a pre-processed integrality gap a structured integrality gap instance whose properties are listed below.

Definition 4.2.3 (Structured Integrality Gap). *A structured integrality gap of a* METRIC LABELING *problem is an instance, $\mathcal{J} = (\mathcal{V}, \mathcal{E}, \Omega_{\mathcal{E}})$; a metric d on $\mathcal{X}$; assignment constraints $C : \mathcal{V} \to 2^{\mathcal{X}}$ and a feasible solution of the earthmover relaxation(see fig. 4.1), $\{Y_v\}$. Further, the metric is such that:* (1) *distinct points in the metric have a positive distance between them;* (2) *the maximum distance between the points is at most* 1*. The structured integrality gap is specified by the tuple $(\mathcal{J} = (\mathcal{V}, \mathcal{E}, \Omega_{\mathcal{E}}), d, \mathcal{X}, C, \{Y_v\})$.*

$$\sum_{e \in E(\mathcal{J})} w_e = 1 \qquad\qquad d(i,j) \leq 1, \text{ for all } 1 \leq i,j \leq k$$

4.3 Results on Labeling Problems

Using our framework, we convert integrality gaps for all the problems mentioned above to inapproximability for the problem, upto a ratio equal to the integrality gap ratio discounting an arbitrarily small additive constant.

Theorem 4.3.1 (Main). *For every number $\theta > 0$, integers m, t, and every structured integrality gap instance, $\mathcal{J} = (\mathcal{V}, \mathcal{E}, \Omega_{\mathcal{E}}, \mathcal{X}, d, C), \{Y_v\}_{\mathcal{V}}$, where $|\mathcal{V}| = m$ and $|\mathcal{X}| = k$, , there is a $\eta > 0$ and δ, and a polynomial time reduction from a* UNIQUE GAMES *instance Υ to a* METRIC LABELING *instance $\mathcal{I} = (\mathcal{W}, \mathcal{E}, \Omega_{\mathcal{E}}, \mathcal{X}, d, F)$ such that:*

- *If Υ is $(1-\eta)$-strongly satisfiable, then $\mathrm{OPT}(\mathcal{I}) \leq \delta \cdot \mathrm{LP}(\mathcal{J}) \cdot (1+\theta)$,*
- *while, if Υ is at most η satisfiable, then $\mathrm{OPT}(\mathcal{I}) \geq \delta \cdot \mathrm{OPT}(\mathcal{J}) \cdot (1-\theta)$*

– *the output instance, $\mathcal{I}$ has the same metric d over the same space $\mathcal{X}$ as $\mathcal{J}$*

– *further, if $\mathcal{J}$ is a* 0-Extension *instance, then $\mathcal{I}$ is a* 0-Extension*; and similarily with* Multiway Cut *and* Uniform Metric Labeling *instances.*

Remark 4.3.2. *The size of the output instance, $\mathcal{I}$ is polynomial in n, the size of the UG instance, Υ; however, doubly exponential in m and k. This restricts the application to constant m and k.*

The constraint—m and k being fixed constants—is not too restrictive. For example, suppose the integrality gap of Metric Labeling restricted to a fixed metric d_0 on $\mathcal{X}_0$ is θ. That is,

$$\theta = \sup_{\mathcal{J}|d=d_0,\mathcal{X}=\mathcal{X}_0} \inf_{Y} \frac{\mathrm{OPT}(\mathcal{J})}{\mathrm{LP}(\mathcal{J},Y)}.$$

Then, for every $\varepsilon > 0$, there is a instance $\mathcal{J}$ whose size solely depends on ε and $|\mathcal{X}_0| = t$ that attains an integrality gap of $\theta - \varepsilon$. Further, such an instance can be searched for in time solely dependent on ε and k. This together with theorem 4.3.1 gives the following strong corollary.

Corollary 4.3.3 (Almost Optimal Inapproximability)**.** *For* Metric Labeling *or* 0-Extension *on any fixed metric d (on $\mathcal{X}$),* Multiway Cut *for a fixed k or the* Uniform Metric Labeling *on a fixed metric space, the earthmover relaxation fig. 4.1 gives an almost optimal polynomial-time approximation assuming the Unique Games Conjecture.*

Note that the exhaustive search is not always necessary. For Uniform Metric Labeling, Metric Labeling and 0-Extension on a metric of size k (but otherwise arbitrary) integrality gaps almost matching the corresponding approximation factor of the best approximation algorithm are already known. However, for Multiway Cut where $k > 3$, our exhaustive-search argument is necessary to show optimal approximability result. We end the discussion with a table displaying the approximability of the various metric labeling problems emphasizing results from this work (see fig. 4.2).

Problem	Metric Size	Best Approx.	LP Gap	Inapproximabilty
Multiway Cut	3	$^{12}/_{11}$	$^{12}/_{11} - \varepsilon$	$^{12}/_{11} - \varepsilon$[48]
Multiway Cut	k	1.2313	1.0123	$\theta - \varepsilon$[48]
Uniform Metric Labeling	k	2[41]	$2 - \varepsilon$[41]	$2 - \varepsilon$[48]
0-Extension	k	$O(\sqrt{\log k})$	$O(\sqrt{\log k})$	$O(\sqrt{\log k})$[48]
Metric Labeling	k	$O(\log k)$	$O(\log k)$	$O(\log k)$[48]
Metric Labeling	poly(n)	$O(\log n)$	$O(\log n)$	$O(\log^{1/2-\varepsilon}(n)$[15]

Figure 4.2: Approximability of Metric Labeling Problems

4.4 Reduction to Metric Labeling Problems

We now describe the reduction from Unique Games to Metric Labeling problems. Given a UG instance, $\Upsilon = (\mathcal{A}, \mathcal{B}, E, \Pi, R)$, an instance of the metric labeling problem, $\mathcal{J} = (\mathcal{V}, \mathcal{E}, \Omega_{\mathcal{E}}, \mathcal{X}, d, C)$, and a solution to the relaxation, Y; construct another instance, $\mathcal{I}$, of Metric Labeling as described below.

The label set and the distance metric on the labels remain the same: $\mathcal{X}$ and d. The vertex set of the instance, $\mathcal{W} = \mathcal{B} \times \mathcal{V} \times \mathcal{X}^R$. The assignment constraints extend from the assignment constraint of $\mathcal{J}$ as follows: vertex (b, v, x) can only be assigned labels from the set corresponding to v.

Edge Distribution. The probability distribution describing the edges (and their weights) — denoted by $\Omega_{\mathcal{E}}$ — uses the dictatorship test constructed from two types of distribution. The overall distribution is the random process described in fig. 4.3; parameters δ and ε are set in the analysis.

For Metric Labeling, the output of the reduction is the instance $\mathcal{I} = (\mathcal{W}, \mathcal{E}, \Omega_{\mathcal{E}}, \mathcal{X}, d, F)$. For 0-Extension and Multiway Cut, the instances produced by the reduction have too many terminals. We handle this by collapsing the gadgets corresponding to a terminal vertex. That is, for every terminal $t \in \mathcal{X}(\mathcal{J})$, contract the set $S_t = \{(b, t, z)\}$ into one vertex t and make it a terminal.

Random Process describing $\Omega_{\mathcal{E}}$ (parameters: δ, ε):

1. Pick $a \in \mathcal{A}$ and two neighbors of a, $b_1, b_2 \in \mathcal{B}$. Let π_1, π_2 denote the permutations of the constraint on edge (a, b_1) and (a, b_2) respectively.

2. With probablity δ,

   (a) pick an edge $e = (v, w) \in \mathcal{E}$,

   (b) sample $(\mathbf{z}_1, \mathbf{z}_2)$ from $\mathsf{Dict}^R(\mathcal{D}_{\bowtie}(Y(v), Y(w))$

   (c) output the edge $((b_1, v, \pi_1(\mathbf{z}_1)), (b_2, w, \pi_2(\mathbf{z}_2)))$.

3. otherwise (that is, with probability $1 - \delta$),

   (a) pick a vertex v from $\mathcal{V}$,

   (b) sample $(\mathbf{z}_1, \mathbf{z}_2)$ from $\mathsf{Dict}^R(\mathcal{D}_{1-\varepsilon}(Y(v)))$

   (c) output the edge $((b_1, v, \pi_1(\mathbf{z}_1)), (b_2, v, \pi_2(\mathbf{z}_2)))$

Figure 4.3: Reduction from UG to Metric Labeling problems

4.4.1 Canonical Labeling from UG Labelings

The first part of the main theorem is simple to prove. Given a labeling Λ of the UG instance, Υ, we design a labeling for Metric Labeling instance output by the reduction. This labeling uses dictator assignments.

Lemma 4.4.1. *For every θ, $\mathcal{J}$ and Y, there is exists η, ε and δ such that, if Λ strongly satisies at least a $1-\eta$ fraction of the constraints of a UG instance Υ, the Metric Labeling problem instance, $\mathcal{I}$ output by the reduction in fig. 4.3 is such that:*

$$\mathrm{OPT}(\mathcal{I}) \leq \delta\, \mathrm{LP}(\mathcal{J}) + \varepsilon + \eta$$

Proof. Construct a labeling $A : \mathcal{B} \times \mathcal{V} \times \mathcal{X}^R \to \mathcal{X}$ as follows: $A(b, v, \mathbf{z}) = (\mathbf{z})_{\Lambda(b)}$.

Assume a picked in item 1 of the reduction be such that Λ satisfies every edge incident on a. Since a is picked at random, this happens with probability $1 - \eta$.

With probablity $1-\varepsilon$, edges from item 3 of the reduction have cost 0 because they are assigned the same point in the metric. Otherwise, the cost is at most 1. Thus, the total cost from this step is at most ε.

Since both (a, b_1) and (a, b_2) are satisfied by Λ, $\pi_1(\Lambda(b_1)) = \pi_2(\Lambda(b_2))$. Using theorem 4.2.2, the expected cost of edges from item 2 is exactly the cost of the LP solution. Thus, the total cost of A is:

$$(1-\eta) \cdot [\delta \, \mathrm{LP}(\mathcal{J}) + (1-\delta)\varepsilon] + \eta \leq \delta \, \mathrm{LP}(\mathcal{J}) + \varepsilon + \eta$$

□

4.4.2 Soundness

The harder part of theorem 4.3.1 is the contrapositive of lemma 4.4.1: if the assignment costs less than $\delta \cdot \mathrm{OPT}(\mathcal{J}) \cdot (1-\theta)$, then the UG instance, Υ is at least η satisfiable. Fix such an assignment $A : \mathcal{B} \times \mathcal{V} \times \mathcal{X}^R \to \mathcal{X}$. We show that any such labeling A of $\mathcal{I}$ can be used to "decode" an assignment for Υ that is more than η-satisfiable. For each $b \in \mathcal{B}$ and each $v \in \mathcal{V}$, define functions $f_{b,v} : \mathcal{X}^R \to \mathcal{X}$ and $g_{a,v} : \mathcal{X}^R \to \blacktriangle_{\mathcal{X}}$ as follows:

$$\begin{aligned} f_{b,v}(\mathbf{z}) &= A(b, v, \mathbf{z}) \\ g_{a,v}(\mathbf{z}) &= \mathop{\mathbf{E}}_{b \in N(a)} \left[f_{b,v}(\pi_{(a,b)}(\mathbf{z})) \right] \end{aligned}$$

For each $a \in \mathcal{A}$, let $\mathsf{val}^{\mathsf{Vt}}(a)$ denote the cost of the edges from item 3 in fig. 4.3 when a is picked as the left vertex (item 1 in fig. 4.3). Similarily, let $\mathsf{val}^{\mathsf{Ed}}(a)$ denote cost from item 2 of the reduction. The total cost of the solution is:

$$\mathsf{val}(A) = \mathop{\mathbf{E}}_{a \in \mathcal{A}} \left[\delta \cdot \mathsf{val}^{\mathsf{Ed}}(a) + (1-\delta) \cdot \mathsf{val}^{\mathsf{Vt}}(a) \right] \tag{4.1}$$

Parameters. We describe how we set the paramters before delving into the analysis. Parameters that arise from the structured gap instance should be considered constants. These are: m, the size of $\mathcal{J}$; α, the minimum non-zero value taken by any coordinate of any variable $\mathbf{x}_v$ in the LP solution; β, the minimum non-zero cost assigned by the metric d; the LP optimum, $\mathrm{LP}(\mathcal{J})$; and the (integral) optimum, $\mathrm{OPT}(\mathcal{J})$.

We set ε and δ based on the error, θ, we can tolerate (see theorem 4.3.1). The rest of the parameters are fixed by these three. We would like $\varepsilon + \eta \leq \delta\theta\,\mathrm{LP}(\mathcal{J})$ to prove item 1 of theorem 4.3.1. Thus, $\delta = \Omega(\varepsilon/\theta)$. On the other hand, as we show below, we need to set $\delta = O(\theta\sqrt{\varepsilon})$ to prove item 2. With this in mind, we set $C = \sqrt{\frac{\mathrm{LP}(\mathcal{J})\beta}{10m}}$. Now, given the error parameter θ, set $\varepsilon = C^4\theta^4$. The main structure theorem follows.

Theorem 4.4.2. *For every $\theta, \varepsilon, \delta > 0$, there is τ, D such that, for every $a \in \mathcal{A}$, one of the three hold:*

- *There is a $v \in \mathcal{V}$ and a coordinate i such that $\mathrm{Inf}_i({g_{a,v}}^{(\leq D)}) \geq \tau$*
- $\mathsf{val}^{\mathsf{Vt}}(a) \geq \frac{\theta\sqrt{\varepsilon}\beta}{m}$
- $\mathsf{val}^{\mathsf{Ed}}(a) \geq \delta(1-\theta/2)\cdot\mathrm{OPT}(\mathcal{J})$

Proof. Set $\varepsilon = C^4\theta^4$ and $\mu = 1-\theta/2$. Let τ and D be as obtained from corollary 3.2.6. Fix an $a \in \mathcal{A}$. Suppose case 1 in the statement does not hold for any $v \in \mathcal{V}$. If there is a $v \in \mathcal{V}$ and a $j \in [t]$ such that $\mathbf{E}_z[(g_{a,v})_j] \leq \mu$, then, applying corollary 3.2.6, we know that:

$$\mathsf{val}^{\mathsf{Vt}}(a) \geq \frac{1}{m}\Omega(\theta\sqrt{\varepsilon}\beta).$$

On the other hand, if every v has a j_v such that the expectation is at least μ, then the assignment, $g_{a,v}$, induces an almost integral assignment. Each v is assigned the coordinate j_v; being an integral assignment, the cost of the edge tests is at least as much as the best integral solution:

$$\mathsf{val}^{\mathsf{Ed}}(a) \geq \delta\cdot\mathrm{OPT}(\mathcal{J})\cdot\mu \geq \delta\,\mathrm{OPT}(\mathcal{J})(1-\theta/2) \qquad \square$$

Note that the weights in a structured integrality gap are scaled so that $\mathrm{OPT}(\mathcal{J}) \leq 1$. Setting $\delta = O(\frac{\theta\sqrt{\varepsilon}\beta}{m})$, makes sure that if a vertex, a does not have any influential coordinates, then the total cost of edges corresponding to a is at least $\delta(1-\theta/2) \cdot \mathrm{OPT}(\mathcal{J})$.

The decoding procedure is to simply pick τ-influential coordinates of ${g_{a,v}}^{(\leq D)}$ (for any v) and $\tau/2$-influential coordinates of ${f_{b,v}}^{(\leq D)}$. That is, set S_a, S_b as follows:

$$S_a = \cup_v \left\{ j |\ \mathrm{Inf}_j({g_{a,v}}^{(\leq D)}) \geq \tau \right\} \quad \text{and} \quad S_b = \cup_v \left\{ j |\ \mathrm{Inf}_j({f_{b,v}}^{(\leq D)}) \geq \tau/2 \right\}$$

Suppose A is such that $\mathrm{OPT}(\mathcal{I}, A) \leq \delta\, \mathrm{OPT}(\mathcal{J}) \cdot (1-\theta)$. Then, from theorem 4.4.2, we know that at least a $\theta/2$ fraction of a have non-empty S_a. Further, using lemma 2.3.2, we know that for every such a, at least a $\tau/2$ fraction of b adjacent to it have at least one element j in S_b such that $j \in \pi_{a,b}(S_a)$.

Consider a (random) assignment Λ obtained by picking a random element out of S_a for each a and similarily for b. From lemma 2.3.1, we know that each of these sets are bounded by $2Dkm/\tau$ in size. Thus, this random assignment satisfies at least a $(\tau^2/4D^2k^2m^2) \cdot \theta/2$ fraction of the edges. Finally, picking η smaller than this quantity and ε proves theorem 4.3.1.

4.5 Concluding Remarks

In this chapter, we show a direct conversion of integrality gaps for a mathematical relaxation to a proof of inapproximability of Metric Labeling problem. This method was expounded in the work of Raghavendra [55], to show a similar coversion procedure for every constraint satisfaction problem (CSP). In his work, the relaxation used is stronger than a linear program; a semi-definite programming relaxation.

Chapter 5

Approximability of Strict CSPs

In this chapter, we will study the approximability of "strict" constraint satisfaction problems. This class is fairly general and includes important problems such as Vertex Cover, Hypergraph Vertex Cover, numerous covering and packing problems and certain scheduling problems. The flavor of the result is similar to the previous result: we describe a simple linear relaxation that capture the approximability of these problems. That is, an integrality gap instance of this relaxation can be turned into a proof of inapproximability[1]. Further, we show a rounding procedure inspired by the above conversion procedure. The work presented is based on a paper with Amit Kumar, Madhur Tulsiani, and Nisheeth Vishnoi [43].

Roadmap. We begin by describing the class of problems we study here, formalizing the notion of a Strict CSP (see section 5.1). The relaxation we analyse is described in section 5.2. Our results on strict constraint satisfaction problems are in section 5.3. The inapproximability reduction is in section 5.4.

[1]Caveat: the gap instance needs to satisfy a connectedness property; hence any general instance does not suffice.

5.1 Description of Problem

STRICT CSP is a generalization of the classic VERTEX COVER, HYPERGRAPH VERTEX COVER, INDEPENDENT SET problems. An instance of VERTEX COVER is a graph $\mathcal{J} = (\mathcal{V}, \mathcal{E})$. The goal is to find a subset of the vertices such that every edge is incident on the subset (i.e., at least one endpoint belongs to the subset). Modeled as a constraint satisfaction problem, the problem seeks an assignment $L : \mathcal{V} \to \{0, 1\}$ such that every edge $e = (v, w)$ satisfies the predicate $(L(v) \vee L(w))$.

Similarily, a STRICT CSP is specified by positive integers k and t denoting the *arity* and the *alphabet size*. An instance of a STRICT CSP is a tuple $\mathcal{J} = (\mathcal{V}, \mathcal{E}, C, \mathcal{X})$ where $\mathcal{V}$ is a set of vertices; $\mathcal{E}$ is a collection of k-tuples of vertices; C is a collection of objective functions $C_v : [t] \to [-1, 1]$, one per vertex; and $\mathcal{X}$ is a collection of constraints, one per edge: $\mathcal{X}_e \subseteq [t]^k$. The goal in a STRICT CSP is to figure out an assignment, $L : \mathcal{V} \to [t]$ such that:

- For *every* edge $e = (v_1, \ldots, v_k)$,$(L(v_1), \ldots, L(v_k))$ belongs to $\mathcal{X}_e$.
- The objective $\sum_v C_v(L(v))$ is *minimized*

For example, VERTEX COVER is a STRICT CSP with $k = t = 2$ where the objective is always the function: $f(1) = 1$; $f(0) = 0$; and the constraints are always the subset $\{(0, 1), (1, 0), (1, 1)\} \in \{0, 1\}^2$.

Fixing k and restricting the choice of the constraints A_e allowed in the specification (as opposed to allowing arbitrary subsets of $[q]^k$) gives raise to particular classes of strict-CSPs – we shall often abuse notation and refer to these classes as *problems*. Many important optimization problems are captured by this specification: VERTEX COVER, HYPERGRAPH VERTEX COVER, INDEPENDENT SET, covering and packing problems to name a few.

Note that strict-CSPs are different from the CSPs considered by Raghavendra [55] where the goal, given a set of constraints, is to find an assignment which maximizes a payoff function associated with whether a constraint is satisfied or not and, in particular, assignments which satisfy only part of the constraints are feasible, e.g., MAXIMUM CUT. We refer to them as strict-CSPs precisely for this reason. Even though optimal inapproximability and approximability for several problems such as MAXIMUM CUT which fell in Raghavendra's framework were known before (see [55]), the main feature of his result was the use of semi-definite programming (SDP)-integrality gaps to come up with Unique Games Conjecture (UGC)-based hardness reductions, complementing the result of Khot and Vishnoi [40] who show how to use UGC-based hardness reductions to come up with SDP-integrality gaps. He gave a generic SDP for this class of CSPs and showed how the approximability of each problem is determined by the corresponding SDP up-to an arbitrarily small additive error assuming the UGC. He noted in his paper that his techniques do not apply to strict-CSPs such as VERTEX COVER and GRAPH-3-COLORING.

In this chapter we present a framework similar to the one in [55] which applies to a large class of strict-CSPs. In particular, we show that a natural linear program (LP) captures precisely (up-to arbitrarily small additive error) the approximability of strict-CSPs such as covering-packing problems, which include VERTEX COVER, HYPERGRAPH VERTEX COVER and INDEPENDENT SET, as observed by Guruswami and Saket [28] - the k-partite-k-uniform-HYPERGRAPH VERTEX COVER problem, and the concurrent open shop problem in scheduling [49], [7]. We show how to convert integrality gap for the LP for these problems to a UNIQUE GAMES-based hardness of approximation result in a principled way. Thus, the above results are obtained by invoking known integrality gaps for the above-mentioned problems. In addition, for covering-packing problems we give a simple rounding algorithm which achieves the integrality gap, again up-to an arbitrarily small additive constant. The rounding result is an analogue in the strict-CSP world of that obtained by Raghavendra and Steurer [57].

We do not attempt to list all the corollaries and, rather, focus on providing a systematic framework to compose LP integrality gap instances for strict-CSPs with UNIQUE GAMES instances and to demonstrate how the rounding algorithm comes out as a natural by-product of the analysis.

5.2 Linear Relaxations for Strict CSPs

We describe the linear relaxation for a general k-STRICT CSP problem. The relaxation is inspired by the Sherali-Adams [64] relaxation and plays a crucial role in our results. Given a STRICT CSP instance $\mathcal{J} = (\mathcal{V}, \mathcal{E}, C, \mathcal{X})$, the relaxation we consider is described in fig. 5.1.

$$\mathrm{LP}(\mathcal{J}) \stackrel{\text{def}}{=} \min \sum_{v \in \mathcal{V}} C_v(Y_v) \tag{5.1}$$

$$\text{subject to}$$

$$\forall_{e=(v_1,v_2,\ldots,v_k)\in\mathcal{E}} \qquad (Y_{v_1}, Y_{v_2}, \ldots, Y_{v_k}) \in \blacktriangle(\mathcal{X}_e) \tag{5.2}$$

$$\forall_{v\in\mathcal{V}} \qquad Y_v \in \blacktriangle_{[t]} \tag{5.3}$$

$$\tag{5.4}$$

Figure 5.1: LP for k-STRICT CSP

Here, for a hyper-edge $e = (v_1, \ldots, v_l)$, $\texttt{ConvexHull}(A_e)$ denotes the convex hull of all assignments $\sigma \in \{0,1\}^l$ which satisfy the constraint A_e. For an instance $\mathcal{I}$, let $\mathrm{LP}(\mathcal{I})$ denote the optimum of the LP of Figure fig. 5.1 for $\mathcal{I}$. Let $\mathsf{val}(\mathcal{I}, x)$ denote the value of $\mathsf{LP}(\mathcal{I})$ for a feasible x to it. Also, let $\mathrm{OPT}(\mathcal{I})$ denote the value of the optimal integral solution for $\mathcal{I}$. For the sake of readability, we will assume that all the hyper-edges are exactly of size k.

Connected Integrality Gap Instance. Our inapproximability of STRICT CSP requires a gap instance of the relaxation with the following properties:

– *Satisfiability:* Every constraint is such that, for every assignment fixing all but one variable involved, there is one assignment for the unfixed variable that satisfies the constraint.

– *Connectedness:* The solution $Y_{\mathcal{V}}$ is such that the distribution induced on each edge has a max. correlation coefficient less than 1.

– *Scale:* The maximum cost of any assignment is at most 1.

A connected integrality gap instance is simply an instance $\mathcal{J} = (\mathcal{V}, \mathcal{E}, C, \mathcal{X})$ along with a solution $Y_{\mathcal{V}}$ satisfying the above properties.

5.3 Results on Strict CSP

The general theme of the results is that for a particular (and amenable) class of STRICT CSP, the linear relaxation in fig. 5.1 gives the best approximation to the optimum for problems in this class assuming the UGC. We state the result formally for interesting special cases.

We start with the VERTEX COVER and HYPERGRAPH VERTEX COVER problems. VERTEX COVER has a simple 2-approximation; and HYPERGRAPH VERTEX COVER a simple k-approximation. Assuming the UGC, these problems are known to be hard to approximate to a better factor [39]. However, the reduction in [39] does not give an insight into why a simple linear program obtains this approximation factor. Our results convert integrality gap instances of this linear program into inapproximability results.

Theorem 5.3.1 (VERTEX COVER inapprox.)**.** *For every $\delta > 0$, every* VERTEX COVER *instance $\mathcal{J} = (\mathcal{V}, \mathcal{E})$ and a solution, $Y_{\mathcal{V}}$ to the relaxation in fig. 5.1, there is a $\eta > 0$ and a polynomial time reduction from* UNIQUE GAMES *to* VERTEX COVER *such that:*

– *If Υ is $(1-\eta)$-strongly satisfiable, then $\mathrm{OPT}(\mathcal{I}) \leq \mathrm{LP}(\mathcal{J}) + \delta$,*

– *while, if Υ is at most η satisfiable, then $\mathrm{OPT}(\mathcal{I}) \geq \mathrm{OPT}(\mathcal{J}) + \delta$.*

Structure Preserving Reduction. While not improving on the results obtained in [39], our reduction shows a direct connection between the relaxation and the problem. To the best of our knowledge, this connection has not been explicit potrayed and used. The reduction in [6] obtains inapproximability of degree-bounded instances of Vertex Cover using a reduction similar to the one in this chapter.

Our reduction has the characteristic of preserving certain structures present in the integrality gap instance. For instance, for Hypergraph Vertex Cover, if the integrality gap instance is k-partite, the output is also k-partite. Incidentally, Hypergraph Vertex Cover on a k-partite, k-uniform instance has been studied and a $k/2$-approximation is known for this subclass. Using our reduction, [28] show that this factor is in fact tight.

Packing and Covering Problems. A covering problem is a Strict CSP with $t = 2$ where the constraints have a covering property: if an assignment $(x_1, x_2, \ldots x_k)$ to the incident vertices of an edge satisfies the constraint on it, then changing any x_j from 0 to 1 also produces an assignment that satisfies that constraint. The problem seeks an assignment with a minimum set of vertices set to 1 while satisfying all the covering constraints. Packing problems are the natural complement of this problem. For example, Vertex Cover and Independent Set are covering and packing problems that are complements of each other.

Theorem 5.3.2 (Covering Problems)**.** *For every $\delta > 0$, every coverint* Strict CSP *instance $\mathcal{J} = (\mathcal{V}, \mathcal{E}, C, \mathcal{X})$ and a solution, $Y_{\mathcal{V}}$ to the relaxation in fig. 5.1, there is a $\eta > 0$ and a polynomial time reduction from* Unique Games *to* Strict CSP *such that:*

– *If Υ is $(1-\eta)$-strongly satisfiable, then* $\mathrm{OPT}(\mathcal{I}) \leq \mathrm{LP}(\mathcal{J}) + \delta$,

– *while, if Υ is at most η satisfiable, then* $\mathrm{OPT}(\mathcal{I}) \geq \mathrm{OPT}(\mathcal{J}) + \delta$.

5.4 Reduction to Strict CSPs

We now describe the reduction from Unique Games to the Strict CSP in hand using a connected integrality gap instance. Given a UG instance, $\Upsilon = (\mathcal{A}, \mathcal{B}, E, \Pi, R)$, an instance of Strict CSP, $\mathcal{J} = (\mathcal{V}, \mathcal{E}, C, \mathcal{X})$ and a connected solution to the relaxation in fig. 5.1; construct another instance of Strict CSP as described below.

The vertex set of the instance, $\mathcal{W} = \mathcal{B} \times \mathcal{V} \times [t]^R$. The set of (strict) constraints of the instance is the support of the a "edge distribution" detailed in fig. 5.2. The reason for describing the constraints using a distribution even though strict constraints do not need weights will be evident in the analysis.

Vertices are weighted as follows. The weight of the vertex $(b, v, \mathbf{z})$ is given by the formula:

$$w(b, v, \mathbf{z}) = \prod_j (Y_v)_{(\mathbf{z}_j)}$$

1. Pick $a \in \mathcal{A}$ and k neighbors of a, $b_1, b_2, \ldots b_k \in \mathcal{B}$. Let $\pi_1, \pi_2, \ldots \pi_k$ denote the permutations of the constraint on corresponding edges.
2. Pick an edge $e = (v_1, v_2, \ldots v_k) \in e$. Let $\mathcal{X}_e$ be the subset representing the constraint on edge e.
3. Let $\mathcal{D} = \mathcal{D}(Y_{v_1}, Y_{v_2}, \ldots Y_{v_k})$ denote the correlated distribution over $[t]^k$ given by the relaxation. Sample $(\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_k)$ from $\mathcal{D}$.
4. Output the constraint $((b_1, v_1, \pi_1(\mathbf{z}_1)), (b_2, v_2, \pi_2(\mathbf{z}_2)), \ldots (b_k, v_k, \pi_k(\mathbf{z}_k)))$ and constraint $\mathcal{X}_e$.

Figure 5.2: Constraint Set of the output instance, $\mathcal{I}$

The followign inapproximability is proven in the rest of this section.

Theorem 5.4.1. *For every θ, and gap instance $\mathcal{J} = (\mathcal{V}, \mathcal{E}, C, \mathcal{X})$, there is a η such that:*

– *If Υ is $(1-\eta)$-strongly satisfiable, then* $\mathrm{OPT}(\mathcal{I}) \leq \mathrm{LP}(\mathcal{J}) + \theta$,

– *while, if Υ is at most η satisfiable, then* $\mathrm{OPT}(\mathcal{I}) \geq \mathrm{OPT}(\mathcal{J}) - \theta$.

– *further, the collection of constraints in $\mathcal{I}$ is the same as in $\mathcal{J}$.*

5.4.1 Deriving an Assignment from a UG Labeling

As in the previous chapter, we will derive an assignment to the instance from a labeling of the UG instance. This assignment will have a cost corresponding to the value of the solution to the relaxation.

Theorem 5.4.2 (Completeness)**.** *Given a labeling Λ that strongly satisies at least a $1-\eta$ fraction of the constraints of the UG instance Υ, the* Strict CSP *instance, $\mathcal{I}$ output by the reduction in fig. 5.2 is such that* $\mathrm{OPT}(\mathcal{I}) \leq \mathrm{LP}(\mathcal{J}) + \eta$.

Proof. Construct a labeling $A : \mathcal{B} \times \mathcal{V} \times [t]^R \to [t]$ as follows:$A(b, v, \mathbf{z}) = \mathbf{z}_{\Lambda(b)}$. The relaxation enforces local constraints such that $\mathcal{D}$ (see item 3 in fig. 5.2) is supported only on satisfying assignments for the constraint inducing $\mathcal{D}$. If a is such that every permutation constratint induced by a neighbor b of it is satisfied by Λ, then the assignment A satisfies all the constraints involving the those vertices. This handles the the neighborhood of a fraction $1-\eta$ of a. For the rest, assign arbitrarily to satisfy the constraints. It is easy to verify that the weights are designed such that the cost of this assignment is at most $\mathrm{LP}(\mathcal{J}) + \eta$. ☐

5.4.2 Soundness

Now, we delve into the harder part of the main theorem. As before, we prove this in the contrapositive: if $\mathrm{OPT}(\mathcal{I}) \leq \mathrm{OPT}(\mathcal{J}) - \theta$, then one can find a labeling Λ for Υ that satisfies at least $\theta^{\Omega(1)}$ fraction of the UG constraints.

Let A be such an assignment: $A : \mathcal{B} \times \mathcal{V} \times [t]^R \to [t]$. That is, $\mathsf{val}(\mathcal{I}, A) \leq \mathrm{OPT}(\mathcal{J}) - \theta$. For each $b \in \mathcal{B}$ and each $v \in \mathcal{V}$, define functions $f_{b,v} : [t]^R \to [t]$ and and $g_{a,v} : [t]^R \to \blacktriangle_{[t]}$ as follows:

$$f_{b,v}(\mathbf{z}) = A(b, v, \mathbf{z}); \qquad\qquad g_{a,v}(\mathbf{z}) = \mathop{\mathbf{E}}_{b \in N(a)} \left[f_{b,v}(\pi_{(a,b)}(\mathbf{z})) \right]$$

Let $\mathsf{val}(a)$ denote the cost of the neighborhood of a single a. The total cost of A is the average over $\mathsf{val}(a)$:

$$\mathsf{val}(\mathcal{I}, A) = \mathop{\mathbf{E}}_{a}[\mathsf{val}(a)]$$

Parameters. As before, the parameters arising out of the gap instance should be considered fixed constants. These are ρ, the maximum over the max. corr. coefficient of every edge distribution induced by the relaxation; m, the size of the instance, α, the minimum non-zero value taken by any coordinate of any variable Y_v in the LP solution; and k the arity of the problem. Let $\mu = \frac{\theta}{2k}$.

Theorem 5.4.3. *For every θ, there exists τ, D such that, of A is an assignment to $\mathcal{I}$ such that $\mathsf{val}(\mathcal{I}, A) \leq \mathrm{OPT}(\mathcal{J}) - \theta$, then one of the following hold for every $a \in \mathcal{A}$:*

- *There is a $v \in \mathcal{V}$ and a coordinate i such that $\mathrm{Inf}_i({g_{a,v}}^{(\leq D)}) \geq \tau$*
- $\mathsf{val}(a) \geq \mathrm{OPT}(\mathcal{J}) - \theta/2$

Proof. Let k, ρ, α and μ be as fixed above. Let δ, τ and D be as obtained by applying corollary 3.2.8. Consider an assignment to $\mathcal{J}$, L obtained as follows: v is assigned j with probability $p_{j,v} = \mathbf{P}_{(b,v,\mathbf{z}) \in N(a)}[A(\mathbf{z}) = j]$ if $p_{j,v}$ is at least μ and unassigned otherwise. Finally, assign the unassigned vertices arbitrarily. Corollary 3.2.8 says that the assignment satisfies all the constraints. Now, since the assignment was induced from A (unless when unassigned) we know that:

$$\mathsf{val}(a) = \mathsf{val}(\mathcal{I}, A) \geq \mathsf{val}(\mathcal{J}, L) - \mu k \geq \mathrm{OPT}(\mathcal{J}) - \mu k \geq \mathrm{OPT}(\mathcal{J}) - \theta/2 \qquad \square$$

The decoding procedure is as in the previous chapter: pick τ-influential coordinates of ${g_{a,v}}^{(\leq D)}$ (for any v) and $\tau/2$-influential coordinates of ${f_{b,v}}^{(\leq D)}$. Set S_a, S_b as follows:

$$S_a = \cup_v \left\{ j | \operatorname{Inf}_j({g_{a,v}}^{(\leq D)}) \geq \tau \right\} \quad \text{and} \quad S_b = \cup_v \left\{ j | \operatorname{Inf}_j({f_{b,v}}^{(\leq D)}) \geq \tau/2 \right\}$$

Suppose A is such that $\mathrm{OPT}(\mathcal{I}, A) \leq \mathrm{OPT}(\mathcal{J}) - \theta$ Then, from theorem 5.4.3, we know that at least a $\theta/2$ fraction of a have non-empty S_a. Further, using lemma 2.3.2, we know that for every such a, at least a $\tau/2$ fraction of b adjacent to it have at least one element j in S_b such that $j \in \pi_{a,b}(S_a)$.

Consider a (random) assignment Λ obtained by picking a random element out of S_a for each a and similarily for b. From lemma 2.3.1, we know that each of these sets are bounded by $2Dkm/\tau$ in size. Thus, this random assignment satisfies at least a $(\tau^2/4D^2k^2m^2) \cdot \theta/2$ fraction of the edges. Finally, picking η smaller than this quantity and θ proves the main theorem.

5.5 Inapproximabilty of Strict CSPs

We derive the approximation for Hypergraph Vertex Cover (incl. Vertex Cover and Independent Set) and packing and covering Strict CSP using the reduction shown above.

Vertex Cover and related problems. Any known integrality gap instance of these problems can be used. To make sure the LP solution is connected, we perturb the solution by a small amount.

Proof of theorem 5.3.1. Optimal integrality gaps are known for the relaxation. For example, even constant rounds of the integrality gaps constructed in [61] suffice. □

Further, theorem 5.4.1 directly proves theorem 5.3.2.

Chapter 6

Ordering Problems

In this chapter, we show that ordering constraint satisfaction problems are approximation resistant using our framework. Ordering CSP includes important problems such as Max. Acyclic Subgraph (MAS) and Feedback Arc Set (FAS) and the approximation resistance of ordering problems has been a significant open question. We show that all Ordering CSP are *approximation resistant* assuming the UGC. The results are from a collection of published work with Moses Charikar, Venkatesan Guruswami, Johan Hastad, and Prasad Raghavendra [27, 11, 26].

Roadmap. We will first describe ordering problems, and known study on the approximation of ordering problems (MAS and FAS in particular). The results are described in section 6.1. The reduction is more involved than the one in the previous chapters; a overview is in section 6.2.

6.1 Description of Problems

Ordering constraint satisfaction problems are a fairly large class of problems, containing classic problems such as the Max. Acyclic Subgraph, Feedback Arc Set and the

Betweenness problem; the first example being the simplest in this class. Below, we describe Max. Acyclic Subgraph, stating our results on approximating it, followed by the generalizations.

Maximum Acyclic Subgraph. Given a directed acyclic graph G, one can efficiently order ("topological sort") its vertices so that all edges go forward from a lower ranked vertex to a higher ranked vertex. But what if a few, say fraction ε, of edges of G are reversed? Can we detect these "errors" and find an ordering with few back edges? Formally, given a directed graph whose vertices admit an ordering with many, i.e., $1-\varepsilon$ fraction, forward edges, can we find a good ordering with fraction α of forward edges (for some $\alpha \to 1$)? This is equivalent to finding a subgraph of G that is acyclic and has as many edges as possible; hence this problem is called the Max. Acyclic Subgraph problem.

Max. Acyclic Subgraph is a classic optimization problem, figuring in Karp's early list of NP-hard problems [35]; the problem remains NP-hard on graphs with maximum degree 3, when the in-degree plus out-degree of any vertex is at most 3. Max. Acyclic Subgraph is also complete for the class of permutation optimization problems, MAX SNP[π], defined in [54], that can be approximated within a constant factor. It is shown in [52] that Max. Acyclic Subgraph is NP-hard to approximate within a factor greater than $\frac{65}{66}$.

On the positive side, the problem is known to be efficiently solvable on planar graphs [47, 32] and reducible flow graphs [59]. Berger and Shor [9] gave a polynomial time algorithm with approximation ratio $1/2 + \Omega(1/\sqrt{d_{\max}})$ where $d_{\max}$ is the maximum vertex degree in the graph. When $d_{\max} = 3$, Newman [52] gave a factor $8/9$ approximation algorithm. Note that a $1/2$-approximation is trivial: a random ordering of the vertices has a fraction $1/2$ of forward edges in expectation. In fact, the better of an arbitrary ordering and its reverse would have half the edges going forward.

Charikar, Makarychev, and Makarychev[12] gave a factor $(1/2+\Omega(1/\log n))$-approximation algorithm for Max. Acyclic Subgraph, where n is the number of vertices. In fact, their algorithm is stronger: given a digraph with an acyclic subgraph consisting of a fraction $(1/2 + \delta)$ of edges, it finds a subgraph with at least a fraction $(1/2 + \Omega(\delta/\log n))$ of edges. This algorithm, and in particular an instance showing tightness of its analysis from [12], plays a crucial role in our work. However, despite much effort, no efficient ρ-approximation algorithm for a constant $\rho > 1/2$ has been found for MAX. ACYCLIC SUBGRAPH.

The complementary objective of minimizing the number of back edges, or equivalently deleting the minimum number of edges in order to make the graph a DAG, leads to the FEEDBACK ARC SET (FAS) problem. This problem admits a factor $O(\log n \log \log n)$ approximation algorithm [63] based on bounding the integrality gap of the natural covering linear program for FAS; see also [19]. Using this algorithm, one can get an approximation ratio of $\frac{1}{2} + \Omega(1/(\log n \log \log n))$ for MAX. ACYCLIC SUBGRAPH.

We prove that this problem is hard to approximate to a factor better than 1/2 assuming the UGC.

Theorem 6.1.1. *For every ε, there is a η such that:*

– *$(1-\eta)$-strongly satisfiable UG instances can be reduced to a* MAX. ACYCLIC SUBGRAPH *instance whose optimum is at least $1 - \varepsilon$.*

– *The reduction converts $\leq \eta$ satisfiable instances into instances whose optimum is atmost $1/2 + \varepsilon$.*

– *The reduction runs in polynomial time, for fixed ε.*

For FEEDBACK ARC SET the above theorem leads to an equivalent "no constant approximation" inapproximability.

Corollary 6.1.2. *For every ε, unless the UGC is false, there is no algorithm that can distinguish instances whose FAS optimum (minimization problem) is at most ε from ones whose optimum is at least $1/2 + \varepsilon$.*

3-ary Ordering: Betweenness. Apart from Max. Acyclic Subgraph, the other Ordering CSP that has received some attention is the Betweenness problem. Betweenness is an Ordering CSP where all the constraints are of the form "X appears between Y and Z" for variables X, Y and Z. In [14], a $\frac{1}{2}$-approximation algorithm is presented for Betweenness on instances that are promised to be perfectly satisfiable.

A special case of our result is that the *Betweenness* problem is hard to approximate beyond a factor 1/3. The Betweenness problem consists of constraints of the form "j lies between i and k" corresponding to the subset $\{123, 321\}$ of S_3.

Indeed, our result holds in a more general setting where the OCSP could consist of a mixture of predicates. We refer the reader to section 6.4 for the formal statement.

Ordering CSPs: Approximation Resistance. Note that a random ordering would again satisfy at least a fraction 1/3 of the constraints. This is because the set of accepting permutations of the vertices involved in a constraint is 6 (size of S_3) while there are 2 accepted permutations. This is a fairly general phenomenon: a problem is approximation resistant if improving on the the trivial algorithm (in this case a random ordering) is already hard. We show that a fairly general class known as the ordering constraint satisfaction (OCSP) problems are all approximation resistant assuming the UGC.

An example of a OCSP would be all instances that contain 75% of constraints of the form "*i before j*" and 25% of constraints of the form "*i between j* and *k*". Hence the definition not only fixes the set of predicates but also the proportion of each predicate that appears in an instance.

We will use the notation $P \sim \Lambda$ to denote a payoff sampled from the distribution Λ. Notice that every payoff $P \sim \Lambda$ is assumed to be on the set of permutations of exactly k elements. However, there is no loss of generality since for every $q \leq k$, a payoff on set Π_q of permutations on q elements can be expressed as a payoff on Π_k by including dummy variables.

Let $\Pi_{k\to\mathbb{N}}$ denote the set of one to one maps from $[k] \to \mathbb{N}$. The domain of a payoff function P can be extended naturally from the set of permutations Π_k to $\Pi_{k\to\mathbb{N}}$. In particular, an injective map $f \in \Pi_{k\to\mathbb{N}}$, along with the standard ordering on the range $\mathbb{N}$ induces a permutation π_f on $[k]$. To extend the payoff, just define $P(f) = P(\pi_f)$ for all $f \in \Pi_{k\to\mathbb{N}}$.

Definition 6.1.3 (Λ-OrderingConstraintSatisfactionProblem (OCSP))**.** *An instance I of Ordering Constraint Satisfaction Problem Λ is given by $I = (\mathcal{V}, \mathcal{P})$ where*

- *$\mathcal{V} = \{y_1, \ldots, y_m\}$ is the set of variables that need to be ordered. Thus an ordering O is a one to one map from $\mathcal{V}$ to natural numbers $\mathbb{N}$.*
- *$\mathcal{P}$ is a probability distribution over constraints/payoffs applied to subsets of at most k variables from $\mathcal{V}$. More precisely, a sample $P \sim \mathcal{P}$ would be a payoff function from Λ, applied on a sequence of variables $y_S = (y_{s_1}, \ldots, y_{s_k})$. If $O_{|S}$ denotes the injective map from $y_S \to \mathbb{N}$ obtained by restricting O to y_S, then the payoff returned is $P(O_{|S})$.*

  Moreover, the type of a payoff $P \sim \mathcal{P}$ sampled from $\mathcal{P}$, is identical to the distribution associated with the OCSP Λ.

For a payoff $P \in \mathcal{P}$, we define $\mathcal{V}(P) \in \mathcal{V}$ to denote the set of variables on which P is applied. The objective is to find an ordering O of the variables that maximizes the total weighted payoff/expected payoff, i.e.,

$$\mathop{\mathbf{E}}_{P\sim\mathcal{P}}\left[P(O_{|P})\right]$$

Here $O_{|P}$ denotes the ordering O restricted to the variables in $\mathcal{V}(P)$. We define the value $\mathrm{OPT}(\mathcal{P})$ *as*

$$\mathrm{OPT}(I) \stackrel{\mathrm{def}}{=} \max_{O:\Pi_{\mathcal{V}\to\mathbb{N}}} \mathop{\mathbf{E}}_{P\sim\mathcal{P}} P(O_{|P})\,.$$

Observe that if the payoff functions P are predicates, then maximizing the payoff amounts to maximizing the number of constraints satisfied. The notions "payoff function" and "constraint" will be used interchangeably.

Definition 6.1.4. *Given an OCSP Λ, let*

$$\Lambda_{max} = \max_{\sigma \in \Pi_k} \mathop{\mathbf{E}}_{P \sim \Lambda} [P(\sigma)] \qquad \Lambda_{random} = \mathop{\mathbf{E}}_{P \sim \Lambda} \mathop{\mathbf{E}}_{\sigma \in \Pi_k} [P(\sigma)]$$

With these definitions, we can state the following general UG-hardness for OCSPs.

Theorem 6.1.5 (General UG-hardness)**.** *For every $\eta > 0$ and every OCSP of bounded arity k, the following holds: Given an instance of the OCSP Λ that admits an ordering with payoff at least $\Lambda_{max} - \eta$, it is Unique Games-hard to find an ordering of the instance that achieves a payoff of at least $\Lambda_{random} + \eta$.*

Notice that theorem 6.1.5 corresponds to the special case where the probability distribution Λ consists of a single payoff function. For the sake of exposition, we will present the proof of theorem 6.1.5 here. The proof of the more general theorem 6.4.3 is syntactically the same.

We define our notion of a ordering CSP and state our results. An ordering constraint satisfaction problem (OCSP) Λ of arity k is specified by a constraint payoff function $P : \Pi_k \to [-1, 1]$ where Π_k is the set of permutations of $\{1, 2, \ldots, k\}$. An instance of such an ordering CSP consists of a set of variables $\mathcal{V}$ and a collection of constraint tuples each of which is an ordered k-tuple of $\mathcal{V}$. The objective is to find a global ordering σ of $\mathcal{V}$ that maximizes the expected payoff $\mathbf{E}[P(\sigma_{|T})]$ for a random $T \in \mathcal{T}$ where $\sigma_{|T} \in \Pi_k$ is the ordering of the k elements of T induced by the global ordering σ. This is just the natural extension of CSPs to the world of ordering problems. (For generality, we allow payoff functions with range $[-1, 1]$ instead of $\{0, 1\}$ which would correspond to True/False constraints.) As with CSPs, we say that an ordering CSP of arity k and payoff function P is *approximation resistant* if its approximation threshold equals $\mathbf{E}_{\alpha \in \Pi_k}[P(\alpha)]$, which is the expected payoff obtained by a random permutation of the variables.

Note that in this language, Max. Acyclic Subgraph corresponds to the simplest ordering CSP: the arity 2 ordering CSP with payoff function that gives value 1 to the identity permutation and 0 to its reverse.

Our main result is that ***every*** ordering CSP, of arity bounded by a fixed k, is approximation resistant. Specifically, for every such OCSP, outperforming the trivial approximation ratio achieved by random ordering is Unique Games-hard.

Theorem 6.1.6 (Main)**.** *Let k be a positive integer and let Λ be a* Ordering CSP *associated with a payoff function $P : \Pi_k \to [-1, 1]$ on the set of k-permutations, Π_k. Let $\Lambda_{max} = \max_{\alpha \in \Pi_k} P(\alpha)$ be the maximum payoff of P, and $\Lambda_{random} = \mathbf{E}_{\alpha \in \Pi_k} P(\alpha)$ the average payoff of P (expected value achieved by a uniform random ordering). Then for every $\varepsilon > 0$, the following hardness result holds. Given an instance of the OCSP specified by payoff function P that admits an ordering with payoff at least $\Lambda_{max} - \varepsilon$, it is Unique Games-hard to find an ordering of the instance that achieves a payoff of at least $\Lambda_{random} + \varepsilon$ with respect to the payoff function P.*

Combining the unique game integrality gap instance of Khot-Vishnoi [40] along with the UG reduction, we obtain SDP integrality gaps for Max. Acyclic Subgraph problem. Our integrality gap instances also apply to a related SDP relaxation studied by Newman [53]. This SDP relaxation was shown to obtain an approximation better than half on random graphs which were previously used to obtain integrality gaps for a natural linear program [52].

6.1.1 Organization

We begin with an outline of the key ideas of the proof in section 6.2. The groundwork for the reduction is laid in section 6.3, where we define and construct multiscale gap instances. We present the dictatorship test in section 6.5, and convert it to a UG-hardness result in section 6.6. Using this UG-hardness result, we present SDP integrality gaps for Max. Acyclic Subgraph in section 6.7.

Towards generalizing these hardness results, we begin with formal definition of ORDERING CSPs and the natural semidefinite program for ORDERING CSPs in section 6.4. The construction of dictatorship tests for an ORDERING CSP starting from an object termed as *multi-scale gap instance* is presented in section 6.5. In section 6.3.1, we exhibit an explicit construction of *multi-scale gap* instances for every ORDERING CSP. Finally, in section 6.6, we sketch the component of the soundness analysis for MAX. ACYCLIC SUBGRAPH and ORDERING CSP hardness results, that is mostly borrowed from [55].

6.2 Overview of Reduction

The outline of the reduction to ordering problems is similar to the reduction in the previous chapters. We model the ordering problem as a CSP in order to use the techniques developed in [55]. An ordering $\sigma : \mathcal{V} \to [n]$ is really an assignment of $[n]$ to $\mathcal{V}$. Since the goal is to maximize satisfying assignments, it does not pay to set the "position" of two distinct vertices, v_1, v_2 to the same value. However, all the reduction used in earlier chapters needs the range of the assignment (the set $[n]$ here) to be of a fixed size independent of the size of the instance. This is because the reduction has a size doubly exponential in this alphabet size. In order to circumvent this issue, we show a *alphabet reduction* method.

To explain this method, consider the following *bucket version* of an ordering problem. Given an instance $\mathcal{J}$ of ORDERING CSP, and a parameter t, the t-bucket version of $\mathcal{J}$ asks for an assignment $\sigma_t : \mathcal{V} \to [t]$. The payoff of a labeling σ_t is the expected payoff of a random ordering that is consistent with σ_t in the following sense: for every v_1 and v_2 such that $\sigma_t(v_1) > \sigma_t(v_2)$, v_1 is placed after v_2 in the ordering. In other words, the random ordering is picked by randomly permuting the set of vertices assigned the same value in the bucket version, while preserving the ordering between such sets. Let $\mathrm{OPT}_t(\mathcal{J})$ denote the optimum of the t-bucket version. At the one end, setting $t = \infty$ gives us the original problem; while setting $t = 1$ gives a trivial instance where the optimum is exactly the payoff of a purely random ordering.

We then use the work of [55] to reduce UG instances to a t-ordering instance for a fixed t (thus making the problem amenable to the techniques in [55].) However, we treat the output instance as an instance of the original ordering problem. The catch is that the reduction does not guarantee anything about $\mathrm{OPT}(\mathcal{I})$ and only about $\mathrm{OPT}_t(\mathcal{I})$. We fix this by using the alphabet reduction developed in section 3.2.5, proving the intermediate theorem.

Theorem 6.2.1. *For every δ, and every* Ordering CSP *instance $\mathcal{J}$, there is are t, η and a polynomial time reduction from* Unique Games *to* Ordering CSP *such that:*

- *If Υ is at least $1-\eta$, then the optimum of the output instance, $\mathcal{I}$ is at least $\mathrm{OPT}(\mathcal{J})-\delta$;*
- *while if Υ is at most η satisfiable, then $\mathrm{OPT}(\mathcal{I}) \leq \mathrm{OPT}_t(\mathcal{J}) + \delta$.*
- *The output instances uses the same payoff functions as $\mathcal{J}$.*

The intermediate theorem is roughly in the scheme of our previous reductions. We compose a gap instance with an instance of unique games and argue about the two cases above. However, unlike the previous chapters, or even unlike Raghavendra's work [56] which we use in our analysis, our gap instance does not arise from a relaxation.

Our gap instances have a gap between the true optimum and the optimum of the bucketed version (for const. t) described above. We use techniques developed in section 3.2.5 to show that the alphabet size can always be a constant independent of the size of the instance. This is only true when the payoff respects permutations in a way orderings do (see section 3.2.5 for a precise formulation of this).

Finally, to show approx. resistance, we construct a *multiscale gap instance*: an instance $\mathcal{J}$ such that $\mathrm{OPT}(\mathcal{J}) \geq \Lambda_{max} - \delta$ while $\mathrm{OPT}_t(\mathcal{J}) \leq \Lambda_{random} + \delta$ where Λ is the class $\mathcal{J}$ belongs to (refer theorem 6.1.6). Section 6.3 shows the construction of such instance for Max. Acyclic Subgraph and for a general Ordering CSP. The two components together prove the main result (see theorem 6.1.6).

6.3 Multiscale Gap Instances

In this section, we construct the multiscale gap instances for Max. Acyclic Subgraph and Ordering CSP, that are used in the inapproximability reduction. The construction follows from the work by Charikar et al on designing algorithms for Max. Acyclic Subgraph [12].

Definition 6.3.1. *For $\eta > 0$ and a positive integer q, a (η, q)-*Multiscale Gap *instance is a weighted directed graph $G = (V, E)$ with the following properties:*

- $\mathrm{OPT}(G) = 1$ *and* $\mathrm{OPT}_q(G) \leq \frac{1}{2} + \eta$
- *There exists a solution $\{\boldsymbol{b}_{u,i} \mid u \in V, i \in [|V|]\}$ to* LC *relaxation with objective value at least $1 - \eta$ such that for all $u \in V$ and $1 \leq i \leq |V|$, we have $\|\boldsymbol{b}_{u,i}\|_2^2 = \frac{1}{|V|}$.*

Theorem 6.3.2 (Theorem 3.1, [12])**.** *If a directed graph G on n vertices has a maximum acyclic subgraph with at least a $\frac{1}{2} + \delta$ fraction of the edges, then, $||G||_C \geq \Omega\left(\frac{\delta}{\log n}\right)$.*

The following lemma and its corollary construct Multiscale Gap instances starting from graphs that are the "tight cases" of the above theorem.

Lemma 6.3.3. *Given $\eta > 0$ and a positive integer q, for every sufficiently large n, there exists a directed graph $G = (V, E)$ on n vertices such that* $\mathrm{OPT}(G) = 1$, $\mathrm{OPT}_q(G) \leq \frac{1}{2} + \eta$ *.*

Proof. Charikar et al (Section 4, [12]) construct a directed graph, $G = (V, E)$, on n vertices whose cut norm is bounded by $O(1/\log n)$. The graph is represented by the skew-symmetric matrix W, where $w_{ij} = \sum_{k=1}^{n} \sin \frac{\pi(j-i)k}{n+1}$. It is easy to verify that for every $0 < q < n$, $\sum_{k=1}^{n} \sin\left(\frac{\pi q k}{n+1}\right) \geq 0$. Thus, $w_{ij} \geq 0$ whenever $i < j$, implying that the graph is acyclic (in other words, $\mathrm{OPT}(G) = 1$).

We bound $\mathrm{OPT}_q(G)$ as follows. Let $\mathrm{OPT}_q(G) = \frac{1}{2} + \delta$ and let $O : V \to [q]$ be the optimal q-ordering. Construct a graph H on q vertices with a directed edge from $O(u)$ to $O(v)$ for every edge $(u, v) \in E$ with $O(u) \neq O(v)$. Now, using theorem 6.3.2, the cut norm of H is bounded from below by $\Omega\left(\frac{\delta}{\log q}\right)$. Moreover, since O is a partition of V, the cut norm of G is at least the cut norm of H. Thus, $\Omega\left(\frac{\delta}{\log q}\right) \leq ||H||_C \leq ||G||_C \leq O\left(1/\log n\right)$Thus, $\delta \leq O\left(\frac{\log q}{\log n}\right)$ implying that $\mathrm{OPT}_q(G) \leq \frac{1}{2} + O\left(\frac{\log q}{\log n}\right)$. Choosing n sufficiently large gives the required result. □

Corollary 6.3.4. *For every $\eta > 0$ and positive integer q, there exists a (η, q)-Multiscale Gap instance with a corresponding SDP solution $\{\boldsymbol{b}_{u,i} | u \in V, i \in [|V|]\}$ and $\boldsymbol{\mu} = \{\mu_e | e \in E\}$ satisfying $\|\boldsymbol{b}_{u,i}\|_2^2 = 1/|V|$ for all $u \in V, i \in [|V|]$.*

Proof. Let $G = (V, E)$ be the graph obtained by taking $\lceil 1/\eta \rceil$ disjoint copies of the graph guaranteed by lemma 6.3.3 and let $m = |V|$. Note that the graph still satisfies the required properties: $\mathrm{OPT}(G) = 1$, $\mathrm{OPT}_q(G) \leq \frac{1}{2} + \eta$. Let O be the ordering of $[m]$ that satisfies every edge of G. Let D denote the distribution over labellings obtained by shifting O by a random offset cyclically. For every $u \in V, i \in [m]$, $\mathbf{P}[D(u) = i] = 1/m$. Further, every directed edge is satisfied with probability at least $1 - \eta$. Being a distribution over integral labellings, D gives raise to a set of vectors satisfying the constraints in definition 6.3.1. G along with these vectors form the required (η, q)-multiscale gap instance. □

6.3.1 Gap Instances for general Ordering Problems

Theorem 6.3.5. *For every positive integer q, k, for every OCSP Λ of arity k and $\eta > 0$ there exists a $(q, \Lambda_{max}, \Lambda_{random} + \eta)$-multiscale gap instance I of Λ.*

Theorem 6.3.6. *For every $\eta > 0$ and positive integers q, k, there is a $m = m(k, q, \eta)$ and a distribution, D, over k-tuples of $[m]$ such that:*

- *The support of D is only over strictly increasing k-tuples of $[m]$.*

– *For any $f : [m] \to [q]$, let $\mathcal{D}_f$ denote the distribution over permutations of $[m]$ obtained by extending f as in section 6.4.2. For any $\sigma = (\sigma_1, \sigma_2, \ldots \sigma_k) \in \Pi_k$,*

$$\left| \mathop{\mathbf{P}}_{y \in \mathcal{D}_f; (d_1, d_2, \ldots d_k) \in D} [y(d_{\sigma_1}) < y(d_{\sigma_2}) < \ldots y(d_{\sigma_k})] - \tfrac{1}{k!} \right| \leq \eta$$

Before delving into the proof of the above theorem, let us see why it implies a proof of theorem 6.3.5.

Proof. (Proof of theorem 6.3.5) Let P be the payoff associated with OCSP Λ and let $\sigma \in \Pi_k$ maximize $P(\sigma)$. Let $m = m(k, q, \eta/k!)$ and D be the distribution as promised by theorem 6.3.6. Let $I = ([m], \sigma(D))$ be an OCSP instance with payoff P where $\sigma(D)$ denotes a distribution with the output permuted by σ. It is easy to check that the value obtained by the trivial ordering of $[m]$ is $P(\sigma) = \max_{\pi \in \Pi_k} P(\pi)$.

Now, viewing I as an instance of the OCSP Λ, define f to be an optimal q-ordering of I. Then, $\mathrm{OPT}_q(I)$ can be bounded as follows:

$$\mathop{\mathrm{OPT}}_q(I) = \mathsf{val}(f) = E_D E_{\pi \in \mathcal{D}_f}[P(\pi)] \leq \sum_{\sigma} \frac{P(\sigma)}{k!} + \eta \leq \Lambda_{random} + \eta\,.$$

$\square$

We will be interested in the set $[m]$ for $m = k^s$ for positive integers k and s. We will think of $[m]$ as the s-tuples of $[k]$, ordered by the lexicographic ordering of the tuples. A k-adic interval of order j, is a interval of $[m]$ specified by an element $r \in [k]^j$ and denotes the subset of $[k]^s$ whose first j coordinates match those in r. It is easy to check that for every j and r, such a set is, in fact, an interval of $[m]$ (due to the lexicographic ordering) and is of length $k^{(s-j)}$. A random k-adic interval of order j is a k-adic interval of order j where r is chosen uniformly at random from $[k]^j$. Unless otherwise specified, the intervals we will talk about in this section will be a k-adic interval.

Every k-adic interval, I, of order j strictly smaller than s naturally contains k disjoint intervals of order $j+1$, denoted by $I_1, I_2, \ldots I_k$ in the order of they appear in I. A random sub-interval J is obtained by chosen by picking one of these k sub-intervals uniformly at random.

The distribution we will analyse is parametrized by k, the arity of the CSP and a parameter s (that will be chosen depending on η and t in theorem 6.3.6.

Definition 6.3.7. *For positive integers k and s, the distribution $D(k, s)$ is a distribution over k-tuples (or equivalently a k-MO instance on $[m]$ for $m = k^s$ as follows.*

1. *Pick a random j uniformly in $1 \leq j \leq s-1$*
2. *Pick a random k-adic interval of order j from $[m]$*
3. *Pick x_j uniformly at random from the sub-interval I_j*
4. *Output $(x_1, x_2, \ldots x_k)$*

The first claim of theorem 6.3.6 follows immediately from the definition since the elements chosen are always in increasing order in the obvious ordering of $[m]$.

In the rest of this section, we will prove that in fact, no function $f : [m] \to [q]$ obtains more than negligible (for large enough s) advantage over random with respect to any permutation π for the distribution $D = D(k, s)$.

Fix a particular function $f : [m] \to [q]$. For $p \in [q]$ and an interval I, let $\mu_p(I)$ denote the fraction of I set to p in f.

Lemma 6.3.8. *For a random interval I chosen as in definition 6.3.7 and a random sub-interval, J, of I, we have*

$$\sum_{p=1}^{q} \mathbf{E}[\mu_p(I) - \mu_p(J)] \leq \sqrt{q/s}$$

Proof. Let $\beta_{j,p}$ be $E[\mu_p(I)^2]$ when I is a random k-adic interval of order j. For any $p \in [q]$,

$$E_{I,J}[|\mu_p(I) - \mu_p(J)|] \leq \left(E[(\mu_p(J) - \mu_p(I))^2]\right)^{1/2} = \left(E_I[\frac{1}{k}\sum_{i=0}^{k-1}(\mu_p(I_i) - \mu_p(I))^2]\right)^{1/2}$$
$$\leq \left(E[\frac{1}{k}\sum_{i=0}^{k-1}(\mu_p(I_i)^2 - \mu_p(I)^2]\right)^{1/2} = (\beta_{j-1,r} - \beta_{j,r})^{1/2}$$

Thus,

$$\sum_{p=1}^{q}\mathbf{E}[\mu_p(I) - \mu_p(J)] \leq \frac{1}{s}\sum_{r=0}^{q-1}\sum_{j=1}^{s}(\beta_{j-1,r} - \beta_{j,r})^{1/2} \leq \frac{1}{s}\sum_{r=0}^{q-1}\left(\sum_{j=1}^{s}1\right)^{1/2}\left(\sum_{j=1}^{s}\beta_{j-1,r} - \beta_{j,r}\right)^{1/2}$$
$$\leq \frac{1}{\sqrt{s}}\sum_{r=0}^{q-1}\beta_{0,r}^{1/2} \leq \sqrt{q/s}\left(\sum_{r=0}^{t-1}\beta_{0,r}\right)^{1/2} = \sqrt{q/s}$$

□

Lemma 6.3.9. *Given positive numbers $a_i^{(j)}$, $b_i^{(j)}$, $i \in [q]$, $j \in [k]$ such that for every j, $\sum_i a_i^{(j)} = \sum_i b_i^{(j)} = 1$,*

$$\sum_{\sigma\in[q]^k}\left|\prod_{j=0}^{k-1}a^j_{\sigma(j)} - \prod_{j=0}^{k-1}b^j_{\sigma(j)}\right| \leq \sum_{j=0}^{k-1}\sum_{i=0}^{q-1}\left|\ a_i^{(j)} - b_i^{(j)}\ \right|.$$

Proof. The proof follows by a simple induction over k. The two sides of the expression are equal for $k = 1$. For $k > 1$,

$$\sum_{\sigma\in[q]^k}\left|\prod_{j=0}^{k-1}a^{(j)}_{\sigma(j)} - \prod_{j=0}^{k-1}b^{(j)}_{\sigma(j)}\right| = \sum_{\sigma\in[q]^k}\left|a^{(0)}_{\sigma(0)}\prod_{j=1}^{k-1}a^{(j)}_{\sigma(j)} - a^{(0)}_{\sigma(0)}\prod_{j=1}^{k-1}b^{(j)}_{\sigma(j)}\right| + \left|a^{(0)}_{\sigma(0)}\prod_{j=1}^{k-1}b^{(j)}_{\sigma(j)} - b^{(0)}_{\sigma(0)}\prod_{j=1}^{k-1}b^{(j)}_{\sigma(j)}\right|$$
$$\leq \sum_{\sigma\in[q]^{k-1}}\left|\prod_{j=1}^{k-1}a^{(j)}_{\sigma(j-1)} - \prod_{j=1}^{k-1}b^{(j)}_{\sigma(j-1)}\right| + \sum_{i=0}^{q-1}\left|a_i^{(0)} - b_i^{(0)}\right|$$
$$\leq \sum_{j=0}^{k-1}\sum_{i=0}^{q-1}\left|\ a_i^{(j)} - b_i^{(j)}\ \right|$$

□

We can now finish the proof of theorem 6.3.6 using the above two lemmas. For any $\sigma \in [q]^k$, let $P(\sigma)$ denote the probability of the event $f(x_i) = \sigma_i$ when $x = (x_1, x_2, \ldots x_k)$ is chosen according to the distribution $D = D(k, s)$. For an interval I, let $P(\sigma, I)$ denote the above probability conditioned on $D(k, s)$ choosing I in step 2.

$$\begin{aligned}
\sum_\sigma \left| P(\sigma) - \mathop{\mathbf{E}}_I \prod_j \mu_{\sigma(j)}(I) \right| &\le \mathop{\mathbf{E}}_I \sum_\sigma \left| P(\sigma, I) - \prod_j \mu_{\sigma_j}(I) \right| \\
&= \mathop{\mathbf{E}}_I \sum_\sigma \left| \prod_j \mu_{\sigma_j}(I_j) - \prod_j \mu_{\sigma_j}(I) \right| \\
&\le \mathop{\mathbf{E}}_I \sum_j \sum_{p \in [q]} |\mu_p(I_j) - \mu_i(I)| && \text{(By lemma 6.3.9)} \\
&\le k\sqrt{q/s} && \text{(By lemma 6.3.8)}
\end{aligned}$$

For any permutation $\pi \in S_k$, the value of the q-ordering f with respect to π, $\mathsf{val}_t^\pi(D, f) = \sum_\sigma P(\sigma)\text{Payoff}^\pi(\sigma)$. Since Payoff takes values in $[0, 1]$, from the above argument,

$$\left| \mathsf{val}_t^\pi(D, f) - \sum_\sigma \text{Payoff}^\pi(\sigma) \mathop{\mathbf{E}}_I \left[\Pi_j \mu_{\sigma_j}(I)\right] \right| \le k\sqrt{q/s}.$$

Further, since the value of the second term on the left hand side is invariant of the permutation π, we get that

$$\left|\mathsf{val}_t^\pi(D, f) - \tfrac{1}{k!}\right| \le k\sqrt{q/s}.$$

Chosing s large enough depending on q and η, we immediately obtained the statement of theorem 6.3.6.

6.4 Reduction to Ordering CSP

In this section, we outline the ideas of the proof of theorem 6.1.5. To this end, we begin by formally defining a class of ordering constraint satisfaction problems.

6.4.1 Formal Definitions

Definition 6.4.1. *An Ordering Constraint Satisfaction Problem (OCSP) Λ is specified by a probability distribution over the family of* payoff functions $P : \Pi_k \to [-1, 1]$ *on the set* Π_k *of permutations on k elements. The integer k is referred to as the arity of the OCSP Λ.*

An example of a OCSP would be all instances that contain 75% of constraints of the form "*i before j*" and 25% of constraints of the form "*i between j* and *k*". Hence the definition not only fixes the set of predicates but also the proportion of each predicate that appears in an instance.

We will use the notation $P \sim \Lambda$ to denote a payoff sampled from the distribution Λ. Notice that every payoff $P \sim \Lambda$ is assumed to be on the set of permutations of exactly k elements. However, there is no loss of generality since for every $q \leq k$, a payoff on set Π_q of permutations on q elements can be expressed as a payoff on Π_k by including dummy variables.

Let $\Pi_{k \to \mathbb{N}}$ denote the set of one to one maps from $[k] \to \mathbb{N}$. The domain of a payoff function P can be extended naturally from the set of permutations Π_k to $\Pi_{k \to \mathbb{N}}$. In particular, an injective map $f \in \Pi_{k \to \mathbb{N}}$, along with the standard ordering on the range $\mathbb{N}$ induces a permutation π_f on $[k]$. To extend the payoff, just define $P(f) = P(\pi_f)$ for all $f \in \Pi_{k \to \mathbb{N}}$.

Observe that if the payoff functions P are predicates, then maximizing the payoff amounts to maximizing the number of constraints satisfied. The notions "payoff function" and "constraint" will be used interchangeably.

Definition 6.4.2. *Given an OCSP Λ, let*

$$\Lambda_{max} = \mathop{\mathbf{E}}_{P\sim\Lambda}\left[\max_{\sigma\in\Pi_k} P(\sigma)\right] \qquad\qquad \Lambda_{random} = \mathop{\mathbf{E}}_{P\sim\Lambda}\mathop{\mathbf{E}}_{\sigma\in\Pi_k}[P(\sigma)]$$

With these definitions, we can state the following general UG-hardness for OCSPs.

Theorem 6.4.3 (General UG-hardness)**.** *For every $\eta > 0$ and every OCSP of bounded arity k, the following holds: Given an instance of the OCSP Λ that admits an ordering with payoff at least $\Lambda_{max} - \eta$, it is Unique Games-hard to find an ordering of the instance that achieves a payoff of at least $\Lambda_{random} + \eta$.*

Notice that theorem 6.1.5 corresponds to the special case where the probability distribution Λ consists of a single payoff function. For the sake of exposition, we will present the proof of theorem 6.1.5 here. The proof of the more general theorem 6.4.3 is syntactically the same.

6.4.2 Relation to CSPs

An ordering O can be thought of as an assignment of values from $\{1,\ldots,m\}$ to each variable y_i such that $y_i \neq y_j$ for all $i \neq j$. By suitably extending the payoff functions $P \in \Lambda$, it is possible to eliminate the "one to one" condition ($y_i \neq y_j$ whenever $i \neq j$). More precisely, we shall extend the domain of payoff functions $P \in \Lambda$ from $\Pi_{k\to[m]}$ to $\mathbb{N}^{[k]}$ - the set of all maps from $[k]$ to $\mathbb{N}$.

Given an arbitrary function $f : [k] \to \mathbb{N}$, define a probability distribution $\mathcal{D}_f$ on the set of permutations Π_k by the following random procedure: 1) For each $j \in \mathbb{N}$ with $f^{-1}(j) \neq \phi$, pick a uniform random permutation π_j of elements in $f^{-1}(j)$. 2) Concatenate the permutations π_j in the natural ordering on $j \in \mathbb{N}$ to obtain the permutation $\pi \in \Pi_k$. For a payoff $P \in \Lambda$, define

$$P(f) = \mathop{\mathbf{E}}_{\pi\sim\mathcal{D}_f}[P(\pi)] \tag{6.1}$$

With this extension of payoff functions, the following lemma shows that optimizing over all orderings is equivalent to optimizing over all assignments of values in $[m]$ to variables $\{y_1, \ldots, y_m\}$.

Lemma 6.4.4. *For an instance $I = (\mathcal{V}, \mathcal{P})$ of a Λ-OCSP with $|\mathcal{V}| = m$, we have*

$$\max_{O \in \Pi_{\mathcal{V} \to \mathbb{N}}} \mathop{\mathbf{E}}_{P \in \mathcal{P}} P(O_{|P}) = \max_{f \in [m]^{\mathcal{V}}} \mathop{\mathbf{E}}_{P \in \mathcal{P}} P(f_{|P})$$

Here $[m]^{\mathcal{V}}$ denotes the set of all functions from $\mathcal{V}$ to $[m]$.

Proof. For every injective map $O : \mathcal{V} \to \mathbb{N}$, there is an injective map $O' : \mathcal{V} \to [m]$ corresponding to the permutation induced by O. Clearly, the objective value of O is the same as O'. Since $O' \in [m]^{\mathcal{V}}$, we have

$$\max_{O \in \Pi_{\mathcal{V} \to \mathbb{N}}} \mathop{\mathbf{E}}_{P \in \mathcal{P}} P(O_{|P}) \leq \max_{f \in [m]^{\mathcal{V}}} \mathop{\mathbf{E}}_{P \in \mathcal{P}} P(f_{|P})$$

Given an arbitrary function $f : \mathcal{V} \to [m]$, define a probability distribution D_f on the orderings $O \in \Pi_{\mathcal{V} \to [m]}$ by the following random procedure: 1) For each $j \in [m]$ with $f^{-1}(j) \neq \phi$, pick a uniform random permutation π_j of elements in $f^{-1}(j)$. 2) Concatenate the permutations π_j in the natural ordering on $j \in \mathbb{N}$ to obtain the ordering $O \in \Pi_{\mathcal{V} \to [m]}$. By our definition of extended payoffs P, it easily follows that,

$$\mathop{\mathbf{E}}_{P \in \mathcal{P}} \left[P(f_{|P}) \right] = \mathop{\mathbf{E}}_{P \in \mathcal{P}} \left[\mathop{\mathbf{E}}_{O \in D_f} P(O_{|P}) \right] = \mathop{\mathbf{E}}_{O \in D_f} \left[\mathop{\mathbf{E}}_{P \in \mathcal{P}} P(O_{|P}) \right].$$

In turn, this implies that

$$\max_{O \in \Pi_{\mathcal{V} \to \mathbb{N}}} \mathop{\mathbf{E}}_{P \in \mathcal{P}} P(O_{|P}) \geq \max_{f \in [m]^{\mathcal{V}}} \mathop{\mathbf{E}}_{P \in \mathcal{P}} P(f_{|P}),$$

thus finishing the proof. $\square$

By virtue of Lemma lemma 6.4.4, the Λ-OCSP instance $I = (\mathcal{V}, \mathcal{P})$ is transformed into a constraint satisfaction problem over variables $\mathcal{V}$, albeit over a domain $[m]$ whose size is not fixed. Specifically, the problem of finding an optimal ordering O for the Λ-OCSP instance can be creformulated as computing

$$\mathrm{OPT}(I) = \max_{y \in [m]^{\mathcal{V}}} \mathop{\mathbf{E}}_{P \in \mathcal{P}} \left[P(y_{\mathcal{V}(P)}) \right] \tag{6.2}$$

Here P refers to the extended payoff function as defined in eq. (6.1). For the sake of convenience, we will use y_P to denote $y_{\mathcal{V}(P)}$.

Taking the analogy with CSPs a step further, one can define a CSP Λ_q for every positive integer $q > 0$. Given an instance $I = (\mathcal{V}, \mathcal{P})$ of Λ-OCSP, the corresponding Λ_q problem is to find a q-ordering that maximizes the expected payoff. Formally, the goal of the Λ_q-CSP instance I is to compute an assignment $y \in [q]^m$ that is the maximizes the following:

$$\mathop{\mathrm{OPT}}_q(I) = \max_{y \in [q]^m} \mathop{\mathbf{E}}_{P \in \mathcal{P}} \left[P(y_P) \right] \tag{6.3}$$

The following claim is an easy consequence of the above definitions:

Claim 6.4.5. *For every Λ-OCSP instance $I = (\mathcal{V}, \mathcal{P})$, and integers $q \leq q'$*

$$\mathop{\mathrm{OPT}}_q(I) \leq \mathop{\mathrm{OPT}}_{q'}(I) \leq \mathrm{OPT}(I)\,,$$

Further, if $|\mathcal{V}| = m$ then $\mathrm{OPT}_m(I) = \mathrm{OPT}(I)$.

6.4.3 SDP Relaxation

Inspired by the interpretation of a Λ-OCSP as a CSP over a large domain, one can formulate a generic semidefinite program along the lines of [55]. The details of the generic semidefinite program are described here.

Given a Λ-OCSP instance $I = (\mathcal{V}, \mathcal{P})$, the goal is to find a collection of vectors $\{\boldsymbol{v}_{i,a}\}_{i\in\mathcal{V},a\in[m]}$ in a sufficiently high dimensional space and a collection $\{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ of distributions over local assignments. For each payoff $P \in \mathcal{P}$, the distribution μ_P is a distribution over $[m]^{\mathcal{V}(P)}$ corresponding to assignments for the variables $\mathcal{V}(P)$. We will write $\mathbf{P}_{x\in\mu_P}\{E\}$ to denote the probability of an event E with under the distribution μ_P.

LC Relaxation

$$
\begin{aligned}
&\text{maximize} \quad && \mathop{\mathbf{E}}_{P\sim\mathcal{P}} \mathop{\mathbf{E}}_{x\sim\mu_P} P(x) && \text{(LC)}\\
&\text{subject to} \quad && \langle \boldsymbol{b}_{s,i}, \boldsymbol{b}_{s',j}\rangle = \mathop{\mathbf{P}}_{x\sim\mu_P}\Big\{x_s = i, x_{s'} = j\Big\} \quad && (P\in\mathrm{supp}(\mathcal{P}),\ \ s,s'\in\mathcal{V}(P),\ \ i,j\in[m])\,. \\
& && && (6.4)\\
& && \mu_P \in \blacktriangle([m]^{\mathcal{V}(P)}) && \forall P\in\mathrm{supp}(\mathcal{P})
\end{aligned}
$$

Figure 6.1: Local Relaxation: LC for a Λ-OCSP

We claim that the above optimization problem can be solved in polynomial time. To show this claim, let us introduce additional real-valued variables $\mu_{P,x}$ for $P \in \mathrm{supp}(\mathcal{P})$ and $x \in [m]^{V(P)}$. We add the constraints $\mu_{P,x} \geq 0$ and $\sum_{x\in[m]^{V(P)}} \mu_{P,x} = 1$. We can now make the following substitutions to eliminate the distributions μ_P,

$$
\mathop{\mathbf{E}}_{x\sim\mu_P} P(x) = \sum_{x\in[m]^{V(P)}} P(x)\mu_{P,x}\,, \qquad \mathop{\mathbf{P}}_{x\sim\mu_P}\Big\{x_i = a\Big\} = \sum_{\substack{x\in[m]^{V(P)}\\ x_i=a}} \mu_{P,x}\,,
$$

$$
\mathop{\mathbf{P}}_{x\sim\mu_P}\Big\{x_i = a, x_j = b\Big\} = \sum_{\substack{x\in[m]^{V(P)}\\ x_i=a,x_j=b}} \mu_{P,x}\,.
$$

After substituting the distributions μ_P by the scalar variables $\mu_{P,x}$, it is clear that an optimal solution to the relaxation of $\mathcal{P}$ can be computed in time $\mathrm{poly}(m^k, |\mathrm{supp}(\mathcal{P})|)$.

The LC relaxation succinctly encodes several constraints. In the following claim, we present some of the additional properties that a feasible solution to LC can be assumed to satisfy.

Claim 6.4.6. *Given a feasible solution* $\{\boldsymbol{b}_{s,i} | s \in \mathcal{V}, i \in [m]\}$, $\boldsymbol{\mu} = \{\mu_e | e \in E\}$ *to the fig. 6.1 relaxation, the vectors can be transformed to another SDP solution* $\{\boldsymbol{b}^*_{s,i}\}$ *with the same objective value such that for some unit vector* $\mathbf{I}$ *the following hold:*

$$\langle \boldsymbol{b}^*_{s,i}, \boldsymbol{b}^*_{s,j} \rangle = 0 \qquad \forall\ i, j \in [m], i \neq j \tag{6.5}$$

$$\sum_{i \in [m]} \langle \boldsymbol{b}^*_{s,i}, \boldsymbol{b}^*_{s,i} \rangle = 1 \tag{6.6}$$

$$\sum_{i \in [m]} \boldsymbol{b}^*_{s,i} = \mathbf{I} \qquad \forall s \in \mathcal{V}\,, \tag{6.7}$$

$$\langle \boldsymbol{b}^*_{s,i}, \mathbf{I} \rangle = \|\boldsymbol{b}^*_{s,i}\|_2^2 \qquad \forall s \tag{6.8}$$

$$\in \mathcal{V}, i \in [m]\,, \tag{6.9}$$

$$\|\mathbf{I}\|_2^2 = 1 \tag{6.10}$$

$$\tag{6.11}$$

Note that while an integrality gap instance to the above relaxation would be an Λ-OCSP instance, I such that $\mathrm{SDP}(I)$ is "large" while $\mathrm{OPT}(I)$ is "small". A *multiscale gap* instance on the other hand has much weaker properties — only requiring $\mathrm{OPT}_q(I)$ to be small — thus making it easier to construct.

Definition 6.4.7. *An instance* I *of a* Λ-*OCSP is a* (q, c, s)*-multiscale gap instance if* $\mathrm{SDP}(I) \geq c$ *and* $\mathrm{OPT}_q(I) \leq s$.

Smoothing Coarsening Gaps

Definition 6.4.8. *For* $\alpha > 0$, *a* (q, c, s)*-multiscale gap instance* $I = (\mathcal{V}, \mathcal{P})$ *over* m *variables is said to be* α*-smooth if for every* $P \in \mathcal{P}$ *and* $x \in [m]^k$, $\mu_{P,x} \geq \alpha$.

Here we will outline a transformation on multiscale gap instance I^*, to another multiscale gap instance I with certain special properties including α-smoothness. Note that the smoothness parameter of the resulting solutions is $\alpha = \frac{\eta}{10m^k}$.

Lemma 6.4.9. *For all $\eta > 0$ the following holds, given a (q, c, s)-multiscale gap instance $I^* = (\mathcal{V}^*, \mathcal{P}^*)$ of a Λ-OCSP, for large enough m, there exists a $(q, c-\eta/5, s+\eta/5)$-multiscale gap instance $I = (\mathcal{V}, \mathcal{P})$ on m variables, an SDP solution $\{\boldsymbol{b}_{s,i}\}_{s\in\mathcal{V},i\in[m]}, \{\mu_P\}_{P\in\mathrm{supp}(\mathcal{P})}$ and a vector $\mathbf{I}$ satisfying*

$$\langle \boldsymbol{b}_{v,i}, \boldsymbol{b}_{v,i}\rangle = \frac{1}{m} \qquad \forall v \in \mathcal{V}, i \in [m]\,, \tag{6.12}$$

$$\mu_{P,x} \geq \frac{\eta}{10m^k} \qquad \forall P \in \mathcal{P}, x \in [m]^k\,, \tag{6.13}$$

and

$$\mathop{\mathbf{E}}_{P\sim\mathcal{P}}\mathop{\mathbf{E}}_{x\sim\mu_P} P(x) \geq c - \frac{\eta}{5} \qquad \mathop{\mathrm{OPT}}_{q}(I) \leq s + \frac{\eta}{5}$$

Proof. Intuitively, the SDP solution corresponding to instance I assigns each of the variables $y_i \in \mathcal{V}$ each of the locations in $[m]$ with equal probability. I is constructed by taking many copies of I^* and joining them side by side such that cyclic shifts of orderings obtain around the same payoff.

More formally, let $L = \lceil \frac{20}{\eta} \rceil$ and set $\mathcal{V} = \mathcal{V}^* \times [L]$. The distribution $\mathcal{P}$ is obtained by simply the product distribution of $\mathcal{P}^*$ and the uniform distribution over $[L]$. That is, for every $p = (y_1, y_2, \ldots y_k)$ in the support of $\mathcal{P}^*$ and for every $l \in [L]$, $\mathbf{P}_{\mathcal{P}}((y_1, l), (y_2, l), \ldots, (y_k, l)) = \mathbf{P}_{\mathcal{P}^*}(p)/L$.

Let O be an optimal ordering for I. Let $m = |\mathcal{V}| = L|\mathcal{V}^*|$. For every $i \in [m]$, define ordering $O^*_{(i)} : \mathcal{V} \to [m]$ to be $O^*(v, k) = i + k|\mathcal{V}| + O(v)$ (addition modulo m). Since except for at most one copy of $\mathcal{P}^*$, every other constraint is ordered as in O, the payoff of $O^*_{(i)}$ is at least $c - \eta/20$.

Further, since the q-ordering value of $\mathcal{P}$ is simply the average of the q-ordering values of the individual pieces, $\mathsf{val}_q(\mathcal{P}) \leq s$.

To construct the vectors, we consider the distribution over assignments obtained by taking, with probability $1-\eta/10$, one of $O^*_{(i)}$ with equal probability and taking a completely random assignment with probability $\eta/10$. It is easy to see that the probability $y \in \mathcal{V}$ is assigned $a \in [m]$ is exactly $1/m$. Further, since we take a completely random assignment with probability $\eta/10$, for any constraints $p \in \mathcal{P}$ and $x \in [m]^k$, the distribution assigns x to p with probability at least $\frac{\eta}{10m^k}$. The payoff obtained by this distribution is at least $(1-\eta/10)(c-\eta/20) \geq c-\eta/5$. The distribution over assignments naturally gives vectors satisfying the required constraints. $\square$

6.5 Dictatorship Test for OCSP

In this section, we will construct a dictatorship test for an OCSP Λ starting with a multi-scale gap instance I for the problem. Formally, let $I^* = (\mathcal{V}^*, \mathcal{P}^*)$ be a (q, c, s) multiscale gap instance with $|\mathcal{V}| = m$. Let $I = (\mathcal{V}, \mathcal{P})$ denote the $(q, c-\frac{\eta}{5}, s+\frac{\eta}{5})$-multiscale gap instance, which is $\alpha = \eta/10m^k$-smooth, obtained from lemma 6.4.9. Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote the SDP solution associated with the instance I. Define a dictatorship test $D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ on orderings O of $[m]^u$ as follows:

6.5.1 Completeness analysis

It is fairly simple to check that the completeness of the dictatorship test $D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ is close to the SDP value of I. Specifically, we will now show,

Lemma 6.5.1.

$$\mathsf{Completeness}(D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}) \geq \mathsf{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\varepsilon k = c - \frac{\eta}{5} - 2\varepsilon k$$

$D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ Test
Let $I = (\mathcal{V}, \mathcal{P})$ denote a $(q, c-\frac{\eta}{5}, s+\frac{\eta}{5})$-multiscale gap instance, which is $\alpha = \eta/10m^k$-smooth. Let $(\boldsymbol{V}, \boldsymbol{\mu})$ denote the SDP solution associated with the instance I.

– Sample a payoff P from the distribution $\mathcal{P}$. Let $\mathcal{V}(P) = S = \{s_1, s_2, \ldots, s_k\}$.

– Sample $\mathbf{z}_S = \{\mathbf{z}_{s_1}, \ldots, \mathbf{z}_{s_k}\}$ from the product distribution μ_P^u, i.e. For each $1 \le j \le R$, $z_S^{(j)} = \{z_{s_1}^{(j)}, \ldots, z_{s_k}^{(j)}\}$ is sampled using the local distribution μ_P on $[m]^{\mathcal{V}(P)}$.

– For each $s_i \in S$ and each $1 \le j \le R$, sample $\tilde{z}^j_{s_i}$ as follows: With probability $(1-\varepsilon)$, $\tilde{z}^{(j)}_{s_i} = z^{(j)}_{s_i}$, and with the remaining probability $\tilde{z}^{(j)}_{s_i}$ is a uniform random element from $[m]$.

– Query the ordering values $O(\tilde{\mathbf{z}}_{s_1}), \ldots, O(\tilde{\mathbf{z}}_{s_q})$.

– Return the Pay-Off : $P\Big(O(\tilde{\mathbf{z}}_{s_1}), \ldots, O(\tilde{\mathbf{z}}_{s_k})\Big)$

Figure 6.2: Dictatorship Test for a general Λ-OCSP

Proof. A dictator "m-ordering" O is given by $O(\mathbf{z}) = z^{(j)}$. The expected payoff returned by the verifier $D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ on O is given by

$$\mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z_S}} \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_S} \left[P\Big(O(\tilde{\mathbf{z}}_{s_1}), \ldots, O(\tilde{\mathbf{z}}_{s_k})\Big)\right] = \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z_S}} \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_S} \left[P_S\Big(\tilde{z}^{(j)}_{s_1}, \ldots, \tilde{z}^{(j)}_{s_k}\Big)\right]$$

With probability $(1-\varepsilon)^k$, $\tilde{z}^{(j)}_{s_i} = z^{(j)}_{s_i}$ for each $s_i \in S$. Further the payoff functions $P \in \mathcal{P}$ take values in $[-1, 1]$. Hence a lower bound for the expected payoff is given by

$$\mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z_S}} \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_S} \left[P\Big(O(\tilde{z}_{s_1}), \ldots, O(\tilde{z}_{s_q})\Big)\right] \ge (1-\varepsilon)^k \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z_S}} \left[P\Big(z^{(j)}_{s_1}, \ldots, z^{(j)}_{s_q}\Big)\right] + (1-(1-\varepsilon)^k) \cdot (-1)$$

The j^{th} coordinates $\mathbf{z}^{(j)}_S = \{z^{(j)}_{s_1}, \ldots, z^{(j)}_{s_q}\}$ are generated from the local probability distribution μ_P. Thus we get,

$$\mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z_S}} \left[P\Big(z^{(j)}_{s_1}, \ldots, z^{(j)}_{s_q}\Big)\right] = \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{x \in \mu_P} \Big[P(x)\Big] = \mathsf{val}(\boldsymbol{V}, \boldsymbol{\mu}) \tag{6.14}$$

The expected payoff is at least $(1-\varepsilon)^k \cdot \mathsf{val}(\boldsymbol{V}, \boldsymbol{\mu}) - (1-(1-\varepsilon)^k) \geq \mathsf{val}(\boldsymbol{V}, \boldsymbol{\mu}) - 2\varepsilon k$. $\square$

6.5.2 Soundness of dictatorship test

The following soundness claim is an immediate consequence of lemma 6.5.4 and lemma 6.5.3.

Theorem 6.5.2. *(Soundness Analysis) For every $\varepsilon > 0$, for any τ-pseudorandom ordering O of $[m]^u$,*

$$\mathsf{val}(O) \leq \underset{q}{\mathrm{OPT}}(I) + O(q^{-\frac{\varepsilon}{2}}) + o_\tau(1)$$

where $o_\tau(1) \to 0$ as $\tau \to 0$ keeping all other parameters fixed.

Lemma 6.5.3. *For every $\varepsilon > 0$, for any τ-pseudorandom ordering O of $[m]^u$*

$$\mathsf{val}(O) \leq \mathsf{val}_q(O^*) + \binom{k}{2} q^{-\frac{\varepsilon}{2}} + o_\tau(1)$$

where O^* is the q-coarsening of O and k denotes the arity of the OCSP Λ.

Proof. Let $OF^{[s,t]} : [m]^u \to \{0,1\}$ denote the functions associated with the q-ordering O^*. For the sake of brevity, we shall write OF^i for $OF^{[i,i]}$.

Note that the loss due to coarsening arises because for some payoffs P the k variables in $\mathcal{V}(P)$ do not fall into distinct bins during coarsening. Let us upper bound the probability that some two of the variables queried $\tilde{\mathbf{z}}_{s_i}, \tilde{\mathbf{z}}_{s_j}$ fall into same block during coarsening, i.e. $O^*(\tilde{\mathbf{z}}_{s_i}) = O^*(\tilde{\mathbf{z}}_{s_j})$. Observe that,

$$\begin{aligned}
\mathbf{P}\left(O^*(\tilde{\mathbf{z}}_{s_i}) = O^*(\tilde{\mathbf{z}}_{s_j})\right) &= \sum_{i \in [q]} \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_{s_i}, \mathbf{z}_{s_j}} \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_{s_i}, \tilde{\mathbf{z}}_{s_j}} \left[OF^i(\tilde{\mathbf{z}}_{s_i}) \cdot OF^i(\tilde{\mathbf{z}}_{s_j})\right] \\
&= \sum_{i \in [q]} \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_{s_i}, \mathbf{z}_{s_j}} \left[T_{1-2\varepsilon} OF^i(\mathbf{z}_{s_i}) \cdot T_{1-2\varepsilon} OF^i(\mathbf{z}_{s_j})\right]
\end{aligned}$$

As O is a q-coarsening of O, for each value $i \in [q]$, there are exactly $\frac{1}{q}$ fraction of $\mathbf{z}$ for which $O^*(\mathbf{z}) = i$. Hence for each $i \in [q]$, $\mathbf{E}_{\mathbf{z}}[OF^i(\mathbf{z}) = \frac{1}{q}]$. Further, since the ordering O^* is τ-pseudorandom, for every $j \in [u]$ and $i \in [q]$, $\mathrm{Inf}_j(T_{1-\varepsilon}OF^i) \leq \tau$. Hence using lemma 2.3.6, for sufficiently large q, the above probability is bounded by $q \cdot q^{-1-\frac{\varepsilon}{2}} + q \cdot o_\tau(1)$. By a simple union bound, the probability that two of the queried values fall in the same bin is at most $\binom{k}{2}\left(q \cdot q^{-1-\frac{\varepsilon}{2}} + q \cdot o_\tau(1)\right)$ As all the payoffs are bounded by 1 in absolute value, we can write

$$\begin{aligned}
\mathsf{val}(O) &\leq \mathsf{val}_q(O^*) + \mathbf{P}\left(\exists i, j \in [k] \text{ such that } O^*(\tilde{\mathbf{z}}_{s_i}) = O^*(\tilde{\mathbf{z}}_{s_j})\right) \\
&\leq \mathsf{val}_q(O^*) + \binom{k}{2} q^{-\frac{\varepsilon}{2}} + o_\tau(1)
\end{aligned}$$

□

Lemma 6.5.4. *For every choice of m, q, ε, and any τ-pseudorandom q-ordering O^* of $[m]^u$,* $\mathsf{val}_q(O^*) \leq \mathrm{OPT}_q(I) + o_\tau(1)$.

Proof. Let $OF^{[s,t]} : [m]^u \to \{0,1\}$ denote the functions associated with the q-ordering O^*. For the sake of brevity, we shall write OF^i for $OF^{[i,i]}$, and $\boldsymbol{\mathcal{F}} = (OF^{(1)}, \ldots, OF^{(q)})$. The expected payoff returned by the verifier in the dictatorship test $D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}$ is given by,

$$\mathsf{val}_q(O^*) = \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_S} \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_S} \left[P\Big(OF(\tilde{\mathbf{z}}_{s_1}), \ldots, OF(\tilde{\mathbf{z}}_{s_k})\Big)\right].$$

Further, since the ordering O^* is τ-pseudorandom, for every $j \in [u]$ we have $\mathrm{Inf}_j(T_{1-\varepsilon}OF^i) \leq \tau$. The proof follows from Lemma lemma 6.6.5 which we prove in the next section. □

6.6 Soundness Analysis for q-Orderings

In this section, we will sketch the proof of lemma 6.5.4. The proof of lemma 6.5.4 closely resembles the soundness analysis of dictatorship tests for the case of Generalized CSPs in [55]. However, in [55], the dictatorship test is analyzed for functions with domain $[q]^u$ and range $\blacktriangle_q$. In our application, we are interested in functions whose domain is $[m]^u$ while the output is in $\blacktriangle_q$ for some q. Hence lemma 6.5.4 is not an a formal consequence of the lemmas in [55].

For the sake of completeness we include a sketch of the proof here. By the preceding argument, to prove Lemma lemma 6.5.4 all that remained was to prove Lemma lemma 6.6.5. We will accomplish this in Section section 6.6.4 after developing some of the necessary preliminaries and tools.

6.6.1 Invariance Principle

The following invariance principle is an immediate consequence of the of Theorem 3.6 in the work Isaksson-Mossel [31].

Theorem 6.6.1. *(Invariance Principle [31]) Let Ω be a finite probability space with the least non-zero probability of an atom at least $\alpha \leq 1/2$. Let $\mathcal{L} = \{\ell_1, \ell_1, \ldots, \ell_m\}$ be an ensemble of random variables over Ω. Let $\mathcal{G} = \{g_1, \ldots, g_m\}$ be an ensemble of Gaussian random variables satisfying the following conditions:*

$$\mathbf{E}[\ell_i] = \mathbf{E}[g_i] \qquad \mathbf{E}[\ell_i^2] = \mathbf{E}[g_i^2] \qquad \mathbf{E}[\ell_i \ell_j] = \mathbf{E}[g_i g_j] \qquad \forall i, j \in [m]$$

Let $K = \log(1/\alpha)$. Let $\mathbf{F} = (F_1, \ldots, F_d)$ denote a vector valued multilinear polynomial and let $H_i = (T_{1-\varepsilon} F_i)$ and $\boldsymbol{H} = (H_1, \ldots, H_d)$. Further let $\mathrm{Inf}_i(\boldsymbol{H}) \leq \tau$ and $\mathbb{V}[H_i] \leq 1$ for all i.

If $\Psi : R^d \to R$ is a Lipschitz-continous function with Lipschitz constant C_0 (with respect to the L_2 norm). Then,

$$\left| \mathbf{E}\left[\Psi(\boldsymbol{H}(\mathcal{L}^u))\right] - \mathbf{E}\left[\Psi(\boldsymbol{H}(\mathcal{G}^u))\right] \right| \leq C_d \cdot C_0 \cdot \tau^{\varepsilon/18K} = o_\tau(1)$$

for some constant C_d depending on d.

6.6.2 Payoff Functions

For the sake of the proof, we will extend the payoff functions P corresponding to the CSP Λ_q to a multilinear polynomial on $\blacktriangle_q^k$. Specifically, the payoff functions $P \in \Lambda_q$ are defined over the set $[q]^k$ where k is the arity of Λ. Given a payoff function $P : [q]^k \to [-1, 1]$, we shall define a function $P' : R^{tk} \to R$ in two steps as follows:

– Define the function P' on Δ_q^k as follows:

$$P'(\mathbf{e}_{\beta_1}, \dots, \mathbf{e}_{\beta_k}) = P(\beta) \qquad\qquad \forall \beta \in [q]^k$$

– Extend the function P' from Δ_q^k to $\blacktriangle_q^k$ as a multilinear polynomial.

$$P'(\mathbf{x}_1, \dots, \mathbf{x}_k) = \sum_{\beta \in [q]^k} P(\beta) \prod_{i=1}^{k} x_{(i,\beta_i)} \qquad\qquad \forall \{\mathbf{x}_1, \dots, \mathbf{x}_k\} \in \blacktriangle_q^k$$

Abusing notation, we shall use $P \in \Lambda_q$ to denote both the payoff function over $[q]^k$ and the corresponding multilinear function over $\blacktriangle_q^k$. The domain of the input to P will be clear from the context.

6.6.3 Local and Global Distributions

Now, we shall describe two ensembles of random variables, namely the local integral ensembles $\mathcal{L}_P$ for each payoff P, and a global Gaussian ensemble $\mathcal{G}$.

Definition 6.6.2. *For every payoff $P \in \mathcal{P}$ of size at most k, the Local Distribution μ_P is a distribution over $[m]^{\mathcal{V}(P)}$. In other words, the distribution μ_P is a distribution over assignments to the CSP variables in set $\mathcal{V}(P)$. The corresponding Local Integral Ensemble is a set of random variables $\mathcal{L}_P = \{\ell_{s_1}, \dots, \ell_{s_k}\}$ each taking values in Δ_m.*

Definition 6.6.3. *The Global Ensemble* $\mathcal{G} = \{\boldsymbol{g}_s | s \in \mathcal{V}, j \in [m]\}$ *are generated by setting* $\boldsymbol{g}_s = \{g_{s,1}, \ldots, g_{s,m}\}$ *where*

$$g_{s,j} = \langle \mathbf{I}, \boldsymbol{b}_{s,j} \rangle + \langle (\boldsymbol{b}_{s,j} - (\langle \mathbf{I}, \boldsymbol{b}_{s,j} \rangle)\mathbf{I}), \zeta \rangle$$

and ζ *is a normal Gaussian random vector of appropriate dimension.*

It is easy to see that the local and global integral ensembles have matching moments up to degree two.

Observation 6.6.4. *For any set* $P \in \mathcal{P}$*, the global ensemble* $\mathcal{G}$ *matches the following moments of the local integral ensemble* $\boldsymbol{\mathcal{L}}_P$

$$\mathbf{E}[g_{s,j}] = \mathbf{E}[\ell_{s,j}] = \langle \mathbf{I}, \boldsymbol{b}_{s,j} \rangle \qquad \mathbf{E}[g_{s,j}^2] = \mathbf{E}[\ell_{s,j}^2] = \langle \mathbf{I}, \boldsymbol{b}_{s,j} \rangle$$
$$\mathbf{E}[g_{s,j} g_{s,j'}] = \mathbf{E}[\ell_{s,j} \ell_{s,j'}] = 0 \qquad \forall j \neq j', s \in \mathcal{V}(P)$$

6.6.4 Putting It All Together

Finally, we will now show the following lemma which forms the core of the soundness argument in lemma 6.5.4.

Lemma 6.6.5. *For a function* $\boldsymbol{\mathcal{F}} : [m]^u \to \blacktriangle_q$ *satisfying* $\mathrm{Inf}_j(T_{1-\varepsilon}\boldsymbol{\mathcal{F}}) \leq \tau$ *for all* $j \in [u]$*,*

$$\mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_S} \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_S} \left[P\Big(\boldsymbol{\mathcal{F}}(\tilde{\mathbf{z}}_{s_1}), \ldots, \boldsymbol{\mathcal{F}}(\tilde{\mathbf{z}}_{s_k}) \Big) \right] \leq \mathop{\mathrm{OPT}}_q(I) + o_\tau(1)$$

Here $o_\tau(1) \to 0$ *as* $\tau \to 0$ *while all other parameters are fixed.*

Proof. Let us denote $\boldsymbol{\mathcal{H}} = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}$. Let $\mathbf{F}(\mathbf{x}), \boldsymbol{H}(\mathbf{x})$ denote the multilinear polynomials corresponding to functions $\boldsymbol{\mathcal{F}}, \boldsymbol{\mathcal{H}}$ respectively. Let us denote,

$$D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) = \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_S} \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_S} \left[P\Big(\boldsymbol{\mathcal{F}}(\tilde{\mathbf{z}}_{s_1}), \ldots, \boldsymbol{\mathcal{F}}(\tilde{\mathbf{z}}_{s_k}) \Big) \right]$$

Each vector $\mathbf{z}_{s_i}$ is independently perturbed to obtain $\tilde{\mathbf{z}}_{s_i}$. The payoff functions P are multilinear when restricted to the domain $\blacktriangle_q$. Consequently, we can write

$$\begin{aligned} D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) &= \mathop{\mathbf{E}}_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_S} \left[P\Big(\mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_{s_1}} [\boldsymbol{\mathcal{F}}(\tilde{\mathbf{z}}_{s_1}) | \mathbf{z}_{s_1}], \ldots, \mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_{s_1}} [\boldsymbol{\mathcal{F}}(\tilde{\mathbf{z}}_{s_q} | \mathbf{z}_{s_k}]) \Big) \right] \\ &= \mathop{\mathbf{E}}_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_S} \left[P\Big(\boldsymbol{\mathcal{H}}(\mathbf{z}_{s_1}), \ldots, \boldsymbol{\mathcal{H}}(\mathbf{z}_{s_k}) \Big) \right] \end{aligned}$$

The last equality is due to the fact $\mathbf{E}_{\tilde{\mathbf{z}}_{s_i}}[\boldsymbol{\mathcal{F}}_{s_i}(\tilde{\mathbf{z}}_{s_i}) | \mathbf{z}_{s_i}] = T_{1-\varepsilon} \boldsymbol{\mathcal{F}}_{s_i}(\mathbf{z}_{s_i}) = \boldsymbol{\mathcal{H}}_{s_i}(z_{s_i})$. For each $s \in S$, the coordinates of $\mathbf{z}_s$ are generated by the distribution μ_P. Thecrefore the above expectation can be written in terms of the polynomial $\boldsymbol{H}$ applied integral ensemble $\mathcal{L}_P$. Specifically, we can write

$$D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) = \mathop{\mathbf{E}}_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{\mathbf{z}_S} \left[P\Big(\boldsymbol{\mathcal{H}}(\mathbf{z}_{s_1}), \ldots, \boldsymbol{\mathcal{H}}(\mathbf{z}_{s_k}) \Big) \right] = \mathop{\mathbf{E}}_{P\in\mathcal{P}} \mathop{\mathbf{E}}_{\mathcal{L}_P^u} \left[P\Big(\boldsymbol{H}(\ell^u_{s_1}), \ldots, \boldsymbol{H}(\ell^u_{s_k}) \Big) \right] \tag{6.15}$$

The following procedure $\mathsf{round}_{\boldsymbol{\mathcal{F}}}$ returns an ordering for the original Λ-OCSP instance I.

Let $\mathsf{round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu})$ denote the expected payoff of the ordering returned by the rounding scheme $\mathsf{round}_{\boldsymbol{\mathcal{F}}}$ on the SDP solution $(\boldsymbol{V}, \boldsymbol{\mu})$ for the Λ-OCSP instance I. By definition, we have:

$$\mathsf{round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) \leq \mathop{\mathrm{OPT}}_{q}(I) \tag{6.16}$$

In the remainder of the proof, we will show the following inequality:

$$\mathsf{round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) \geq D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) - o_\tau(1)$$

Along with Equation (eq. (6.16)), this would imply that $D^{\varepsilon}_{\boldsymbol{V},\boldsymbol{\mu}}(\boldsymbol{\mathcal{F}})$ is less than $\mathrm{OPT}_q(I) + o_\tau(1)$, thus showing the required claim. To this end, we will arithmetize the value of $\mathsf{round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu})$. Notice that the $\boldsymbol{g}_i$ are nothing but samples of the Global Ensemble $\mathcal{G}$ asso-

$\mathsf{round}_{\mathcal{F}}$ Scheme

Input: A Λ-OCSP instance $I = (\mathcal{V}, \mathcal{P})$ with m variables and an SDP solution $\{\boldsymbol{b}_{v,i}\}, \{\mu_P\}$.

Truncation Function Let $f_\blacktriangle : R^q \to \blacktriangle_q$ be a Lipschitz-continous function such that for all $\mathbf{x} \in \blacktriangle_q$, $f_\blacktriangle(\mathbf{x}) = \mathbf{x}$. Clearly, a function $f_\blacktriangle$ of this nature can be constructed with a Lipschitz constant C_q depending on q.

Scheme Sample R vectors $\zeta^{(1)}, \ldots, \zeta^{(R)}$ with each coordinate being i.i.d normal random variable.
For each $y_s \in \mathcal{V}$ do

– For all $1 \leq j \leq R$ and $c \in [m]$, compute the projection $g_{s,c}^{(j)}$ of the vector $\boldsymbol{b}_{s,c}$ as follows:

$$g_{s,c}^{(j)} = \langle \mathbf{I}, \boldsymbol{b}_{s,c} \rangle + \Big[\langle (\boldsymbol{b}_{s,c} - (\langle \mathbf{I}, \boldsymbol{b}_{s,c} \rangle) \mathbf{I}), \zeta^{(j)} \rangle \Big]$$

– Evaluate the function $\boldsymbol{\mathcal{H}} = T_{1-\varepsilon}\boldsymbol{\mathcal{F}}$ with $g_{s,c}^{(j)}$ as inputs. In other words, compute $\mathbf{p}_s = (p_{s,1}, \ldots, p_{s,q})$ as follows:

$$\mathbf{p}_s = \boldsymbol{H}(\mathbf{g_s})$$

– Round $\mathbf{p}_s$ to $\mathbf{p}_s^* \in \blacktriangle_q$ by using a Lipschitz-continous truncation function $f_\blacktriangle : R^q \to \blacktriangle_q$.

$$\mathbf{p}_s^* = f_\blacktriangle(\mathbf{p}_s)\,.$$

– Assign the Λ-OCSP variable $y_s \in \mathcal{V}$ the value $j \in [q]$ with probability $p_{s,j}^*$.

Figure 6.3: Rounding Scheme for a Λ-OCSP

ciated with I. By definition, the expected payoff is given by

$$\mathsf{round}_{\mathcal{F}}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathcal{G}_P^u} \Big[P\Big(f_\blacktriangle(\boldsymbol{H}(\mathbf{g}_{s_1}^u)), \ldots, f_\blacktriangle(\boldsymbol{H}(\mathbf{g}_{s_k}^u)) \Big) \Big] \tag{6.17}$$

We will show that the quantities in equation (eq. (6.15)) and equation (eq. (6.17)) are roughly equal. Fix a payoff $P \in \mathcal{P}$. Let $\Psi_P : R^{qk} \to R$ be a Lipschitz continous function defined as follows:

$$\Psi_P(\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_k) = P\Big(f_\blacktriangle(\mathbf{p}_1), \ldots, f_\blacktriangle(\mathbf{p}_k) \Big) \qquad \forall \mathbf{p}_1, \ldots \mathbf{p}_k \in \blacktriangle_q\,.$$

Applying the invariance principle (theorem 6.6.1) with the ensembles $\mathcal{L}_P$, $\mathcal{G}_P$, Lipschitz continous functional Ψ and the vector of kq multilinear polynomials given by $(\boldsymbol{H}, \boldsymbol{H}, \ldots, \boldsymbol{H})$ where $\boldsymbol{H} = (H_1, \ldots, H_q)$, we get the required result:

$$\mathsf{round}_{\boldsymbol{\mathcal{F}}}(\boldsymbol{V}, \boldsymbol{\mu}) = \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathcal{G}_P^u} \left[\Psi_P \Big(\boldsymbol{H}(\mathbf{g}_{s_1}^u), \ldots, \boldsymbol{H}(\mathbf{g}_{s_k}^u) \Big) \right]$$

$$\geq \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathcal{L}_P^u} \left[\Psi_P \Big(\boldsymbol{H}(\ell_{s_1}^u), \ldots, \boldsymbol{H}(\ell_{s_k}^u) \Big) \right] - o_\tau(1) \quad (\because InvariancePrinciple(theorem\ 6.6.1))$$

$$\geq \mathop{\mathbf{E}}_{P \in \mathcal{P}} \mathop{\mathbf{E}}_{\mathcal{L}_P^u} \left[P \Big(\boldsymbol{H}(\ell_{s_1}^u), \ldots, \boldsymbol{H}(\ell_{s_k}^u) \Big) \right] - o_\tau(1) \quad (\because \Psi_P(\mathbf{p}_1, \ldots, \mathbf{p}_k) = P(\mathbf{p}_1, \ldots, \mathbf{p}_k)\ if\ \forall i, \mathbf{p}_i \in \blacktriangle_q)$$

$$= D^{\varepsilon}_{\boldsymbol{V}, \boldsymbol{\mu}}(\boldsymbol{\mathcal{F}}) - o_\tau(1) \qquad (\because eq.\ (6.15))$$

□

6.7 SDP Integrality Gap

In this section, we construct integrality gaps for the fig. 6.1 relaxation using the unique games hardness reduction. We show the following result.

Theorem 6.7.1. *For any $\gamma > 0$, there exists a directed graph G such that the value of semi definite program fig. 6.1 is at least $1 - \gamma$, while* $\mathrm{OPT}(G) \leq \frac{1}{2} + \gamma$.

The proof uses a bipartite variant of the Khot-Vishnoi [40] Unique Games integrality gap instance as in [55, 48]. Specifically, the following is a direct consequence of [40].

The integrality gap instance $U = (\mathcal{A}_U \cup \mathcal{B}_U, E, \Pi = \{\pi_e : [u] \to [u] \mid e \in E\}, [u])$ presented in [40] is not bipartite. To obtain a bipartite unique games instance U', duplicate the vertices by setting $\mathcal{A}_U = \{(b, 0) | b \in V\}$ and $\mathcal{B}_U = \{(b, 1) | b \in V\}$. Further for each edge $(a, b) \in E$, introduce two edges $((a, 0), (b, 1))$ and $((a, 1), (b, 0))$ in U'. The SDP solution for the bipartite instance U' is obtained by assigning the vector corresponding to $b \in V$ to both vertices $(b, 0)$ and $(b, 1)$. Except for these minor modifications, the following theorem is a direct consequence of [40]

Theorem 6.7.2. *[40] For every $\eta > 0$, there exists a UG instance, $U = (\mathcal{A}_U \cup \mathcal{B}_U, E, \Pi = \{\pi_e : [u] \to [u] \mid e \in E\}, [u])$ and vectors $\{\boldsymbol{v}_{b,\ell}\}$ for every $b \in \mathcal{B}_U$, $\ell \in [u]$ and a unit vector $\mathbf{I}$ such that the following conditions hold:*

– *No assignment satisfies more than η fraction of constraints in Π.*

– *For all $b, b' \in \mathcal{B}_U, \ell, \ell' \in [u]$, $\langle \boldsymbol{v}_{b,\ell}, \boldsymbol{v}_{b',\ell'} \rangle \geq 0$ and $\langle \boldsymbol{v}_{b,\ell}, \boldsymbol{v}_{b,\ell'} \rangle = 0$.*

– *For all $b \in \mathcal{B}_U$, $\sum_{\ell \in [u]} \boldsymbol{v}_{b,\ell} = \mathbf{I}$ and $\langle \boldsymbol{v}_{b,\ell}, \mathbf{I} \rangle = \|\boldsymbol{v}_{b,\ell}\|_2^2$.*

– *The SDP value is at least $1-\eta$: $\mathbf{E}_{a \in \mathcal{A}_U, b, b' \in \mathcal{B}_U} \left[\sum_{\ell \in [u]} \langle \boldsymbol{v}_{b,\pi_{a\to b}(\ell)}, \boldsymbol{v}_{b',\pi'_{a\to b'}(\ell)} \rangle \right] \geq 1 - \eta$*

Proof of theorem 6.7.1. Let G be a (η, t)-multiscale gap instance with m vertices. Apply theorem 6.7.2, with a sufficiently small η to obtain a UGC instance U and SDP vectors $\{\boldsymbol{v}_{b,\ell} | b \in \mathcal{B}_U, \ell \in [u]\}$. Consider the instance Ψ constructed by running the UG-hardness reduction in section 6.4 on the UG instance U. The set of vertices of Ψ is given by $\mathcal{B}_U \times [m]^u$. Set $M = |\mathcal{B}_U| \times m^u$ and $N = |\mathcal{B}_U|$.

The program fig. 6.1 on the instance Ψ contains M vectors $\{\mathbf{W}_i^{(b,\mathbf{z})} | i \in [M]\}$ for each vertex $(b, \mathbf{z}) \in \mathcal{B}_U \times [m]^u$

Define a solution to fig. 6.1 as follows: Set the vector $\mathbf{I}$ to be the corresponding vector in the instance U. For each vertex $(b, \mathbf{z})$ of the graph Ψ define SDP vectors $\{\mathbf{W}_i^{(b,\mathbf{z})} | i \in [M]\}$ as follows:

$$\mathbf{W}_i^{(b,\mathbf{z})} = \begin{cases} \sum_{z_\ell = i} \boldsymbol{v}_{b,\ell} & \forall i \in [m], (b, \mathbf{z}) \in \mathcal{B}_U \times [m]^u \\ 0 & \forall i \notin [m] \end{cases}$$

Now we will check that the SDP vectors $\{\mathbf{W}_i^{(b,\mathbf{z})}\}$ satisfy conditions eq. (6.10)–eq. (6.7) of the fig. 6.1 relaxation.

– (Constraint eq. (6.8)) Since the vectors $\{\boldsymbol{v}_{b,\ell}\}$ have non-negative dot-product, the vectors $\{\mathbf{W}_i^{(b,\mathbf{z})}\}$ have non-negative inner-products too.

– (Constraint eq. (6.10)) For a fixed b and $\mathbf{z}$, the vectors $\{\mathbf{W}_i^{(b,\mathbf{z})}\}$ are constructed by *partitioning* the vectors $\{\boldsymbol{v}_{b,\ell}\}$ and assigning the vector sum over the partitions. Hence, for any i, j, the vectors $\mathbf{W}_i^{(b,\mathbf{z})}$ and $\mathbf{W}_j^{(b,\mathbf{z})}$ sum over disjoint set of ℓ. Thus,

$$\langle \mathbf{W}_i^{(b,\mathbf{z})}, \mathbf{W}_j^{(b,\mathbf{z})} \rangle = \langle \sum_{z_\ell = i} \boldsymbol{v}_{b,\ell}, \sum_{z_{\ell'} = j} \boldsymbol{v}_{b,\ell'} \rangle = 0 \qquad .$$

– (Constraint eq. (6.5)) For every vertex $(b, \mathbf{z})$ we have,

$$\sum_{i,j \in [M]} \langle \mathbf{W}_i^{(b,\mathbf{z})}, \mathbf{W}_j^{(b,\mathbf{z})} \rangle = \sum_{\ell, \ell' \in [u]} \langle \boldsymbol{v}_{b,\ell}, \boldsymbol{v}_{b,\ell'} \rangle = \sum_{\ell \in [u]} \langle \boldsymbol{v}_{b,\ell}, \boldsymbol{v}_{b,\ell} \rangle = 1 \,.$$

– (Constraint eq. (6.6)) For $i \notin m$, we have $\mathbf{W}_i^{(b,\mathbf{z})} = 0$, thereby trivially satisfying constraint eq. (6.6). For $i \in [m]$, we can write:

$$\langle \mathbf{I}, \mathbf{W}_i^{(b,\mathbf{z})} \rangle = \sum_{z_\ell = i} \langle \mathbf{I}, \boldsymbol{v}_{b,\ell} \rangle = \sum_{z_\ell = i} \|\boldsymbol{v}_{b,\ell}\|_2^2 \,.$$

Due to orthogonality of the vectors $\{\boldsymbol{v}_{b,i}\}$ for every vertex $b \in \mathcal{B}_U$, we get

$$\langle \mathbf{W}_i^{(b,\mathbf{z})}, \mathbf{W}_i^{(b,\mathbf{z})} \rangle = \langle \sum_{z_\ell = i} \boldsymbol{v}_{b,\ell}, \sum_{z_\ell = i} \boldsymbol{v}_{b,\ell} \rangle = \sum_{z_\ell = i} \|\boldsymbol{v}_{b,\ell}\|_2^2 = \langle \mathbf{I}, \mathbf{W}_i^{(b,\mathbf{z})} \rangle$$

– (Constraint eq. (6.7)) is satisfied by choice of $\mathbf{I}$.

To prove that the SDP value is close to 1, we first fix a particular choice of $a \in \mathcal{A}_U$, $b, b' \in \mathcal{B}_U$. Set $\pi = \pi_{a \to b}, \pi' = \pi_{a \to b'}$. The SDP value of edges from $(b, *)$ to $(b', *)$ is:

$$\mathop{\mathbf{E}}_{e\in G}\mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_u,\tilde{\mathbf{z}}_v}\sum_{i<j}\langle \mathbf{W}_i^{(b,\pi(\mathbf{z}_u))},\mathbf{W}_j^{(b',\pi'(\mathbf{z}_v))}\rangle = \mathop{\mathbf{E}}_{e\in G}\mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_u,\tilde{\mathbf{z}}_v}\sum_{i<j}\langle(\sum_{\tilde{\mathbf{z}}_u^{\ell}=i}\boldsymbol{v}_{b,\ell}),(\sum_{\tilde{\mathbf{z}}_v^{\ell'}=j}\boldsymbol{v}_{b',\ell'})\rangle$$
$$\geq \sum_{\ell}(\langle \boldsymbol{v}_{b,\pi(\ell)},\boldsymbol{v}_{b',\pi'(\ell)}\rangle)\mathop{\mathbf{E}}_{e\in G}\mathop{\mathbf{P}}_{\tilde{\mathbf{z}}_u,\tilde{\mathbf{z}}_v}[\tilde{\mathbf{z}}_u^{\ell}<\tilde{\mathbf{z}}_v^{\ell}]$$

With probability at least $(1-2\varepsilon)^2$, $\tilde{\mathbf{z}}_u=\mathbf{z}_u$, $\tilde{\mathbf{z}}_v=\mathbf{z}_v$. Further, since the coordinates of $\mathbf{z}_u,\mathbf{z}_v$ are generated from the multiscale gap instance, G, $\mathbf{P}[\mathbf{z}_u^{\ell}<\mathbf{z}_v^{\ell}]\geq 1-\eta$. Hence,

$$\mathop{\mathbf{E}}_{e\in G}\mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_u,\tilde{\mathbf{z}}_v}\sum_{i<j}\langle \mathbf{W}_i^{(b,\pi(\mathbf{z}_u))},\mathbf{W}_j^{(b',\pi'(\mathbf{z}_v))}\rangle \geq (1-2\varepsilon)^2(1-\eta)\sum_{\ell}\langle \boldsymbol{v}_{b,\pi(\ell)},\boldsymbol{v}_{b',\pi'(\ell)}\rangle$$

Thus, the expected payoff over the whole instance is:

$$\mathop{\mathbf{E}}_{a,b,b'}\mathop{\mathbf{E}}_{e\in G}\mathop{\mathbf{E}}_{\tilde{\mathbf{z}}_u,\tilde{\mathbf{z}}_v}\sum_{i<j}\langle \mathbf{W}_i^{(b,\pi(\mathbf{z}_u))},\mathbf{W}_j^{(b',\pi'(\mathbf{z}_v))}\rangle \geq (1-2\varepsilon)^2(1-\eta)\mathop{\mathbf{E}}_{a\in\mathcal{A}_U,b,b'\in\mathcal{B}_U}\sum_{\ell}\langle(\boldsymbol{v}_{b,\pi(\ell)}\boldsymbol{v}_{b',\pi'(\ell)})\rangle$$
$$\geq (1-2\varepsilon)^2(1-\eta)(1-\eta)\,.$$

Hence for sufficiently small choice of parameters ε,η and η, the SDP value for Ψ is greater than $1-\gamma$. On the other hand, the soundness analysis in section 6.4 (theorem 6.1.6) implies that the integral optimum for Ψ is at most $\frac{1}{2}+\gamma$ with sufficiently small choice of ε,η and η. $\square$

Bibliography

[1] Aaron Archer, Jittat Fakcharoenphol, Chris Harrelson, Robert Krauthgamer, Kunal Talwar, and Éva Tardos. Approximate classification via earthmover metrics. In *SODA '04: Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1079–1087, 2004. 37

[2] 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. 5

[3] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of np. *J. ACM*, 45(1):70–122, 1998. 5

[4] Per Austrin. Balanced max 2-sat might not be the hardest. In *STOC '07: Proceedings of the 39th Annual ACM Symposium on Theory of Computing*, pages 189–197, 2007. 7

[5] Per Austrin. Towards sharp inapproximability for any 2-csp. In *FOCS '07: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 307–317, 2007. 7

[6] Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory of Computing*, 7(1):27–43, 2011. 51

[7] Nikhil Bansal and Subhash Khot. Inapproximability of hypergraph vertex cover and applications to scheduling problems. In *Manuscript*, 2009. 48

[8] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *STOC '98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 161–168, 1998. 37

[9] Bonnie Berger and Peter W. Shor. Tight bounds for the Maximum Acyclic Subgraph problem. *J. Algorithms*, 25(1):1–18, 1997. 57

[10] Gruia Calinescu, Howard Karloff, and Yuval Rabani. Approximation algorithms for the 0-extension problem. In *SODA '01: Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 8–16, 2001. 2, 4, 36

[11] Moses Charikar, Venkatesan Guruswami, and Rajsekar Manokaran. Every permutation CSP of arity 3 is approximation resistant. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity*, 2009. 56

[12] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. On the advantage over random for maximum acyclic subgraph. In *FOCS*, pages 625–633. IEEE Computer Society, 2007. 57, 58, 65

[13] Chandra Chekuri, Sanjeev Khanna, Joseph (Seffi) Naor, and Leonid Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. *SIAM J. Disc. Math.*, 18:608–625, 2005. 37

[14] Benny Chor and Madhu Sudan. A geometric approach to betweenness. *SIAM Journal on Discrete Mathematics*, 11(4):511–523, November 1998. 59

[15] Julia Chuzhoy and Joseph (Seffi) Naor. The hardness of metric labeling. *SIAM J. Comput.*, 36(5):1376–1386, 2006. 36, 37, 41

[16] Gruia Cvalinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. In *STOC '98: Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 48–52, 1998. 4, 35

[17] Elias Dahlhaus, David Johnson, Christos Papadimitriou, Paul Seymour, and Mihalis Yannakakis. The complexity of multiway cuts (extended abstract). In *STOC '92: Proceedings of the 24th Annual ACM Symposium on Theory of Computing*, pages 241–251, 1992. 35, 36

[18] Irit Dinur and Shmuel Safra. On the hardness of approximating minimum vertex cover. *Annals of Mathematics*, 162(1):439–485, 2005. 5

[19] Guy Even, Joseph Naor, Baruch Schieber, and Madhu Sudan. Approximating minimum feedback sets and multicuts in directed graphs. *Algorithmica*, 20(2):151–174, 1998. 58

[20] Jittat Fakcharoenphol, Chris Harrelson, Satish Rao, and Kunal Talwar. An improved approximation algorithm for the 0-extension problem. In *SODA '03: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 257–265, 2003. 36

[21] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. Approximating metrics by tree metrics. *SIGACT News*, 35(2):60–70, 2004. 37

[22] 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. 5

[23] William Feller. *An Introduction to Probability Theory and Its Applications, Vol. 1, 3rd Edition*. Wiley, 3rd edition, January 1968. 23

[24] M. Grtschel, L. Lovsz, and A. Schrijver. *Geometric Algorithms And Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second ed. edition, 1993. 11

[25] Anupam Gupta and Éva Tardos. A constant factor approximation algorithm for a class of classification problems. In *STOC '00: Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 652–658, 2000. 37

[26] Venkatesan Guruswami, Johan Håstad, Rajsekar Manokaran, Prasad Raghavendra, and Moses Charikar. Beating the random ordering is hard: Every ordering csp is approximation resistant. *SIAM J. Comput.*, 40(3):878–914, 2011. 56

[27] Venkatesan Guruswami, Rajsekar Manokaran, and Prasad Raghavendra. Beating the random ordering is hard : Inapproximability of maximum acyclic subgraph. *In FOCS '08: Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, 2008. 7, 56

[28] Venkatesan Guruswami and Rishi Saket. On the inapproximability of vertex cover on k-partite k-uniform hypergraphs. In *Manuscript*, 2010. 48, 51

[29] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, 2001. 5, 8

[30] Johan Håstad. Every 2-csp allows nontrivial approximation. In Harold N. Gabow and Ronald Fagin, editors, *STOC*, pages 740–746. ACM, 2005. 20

[31] Marcus Isaksson and Elchanan Mossel. Maximally stable gaussian partitions with discrete applications, August 03 2009. 9, 10, 82

[32] A. Karazanov. On the minimal number of arcs of a digraph meeting all its directed cutsets. *Graph Theory Newsletters*, 8, 1979. 57

[33] David R. Karger, Philip Klein, Cliff Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. In *STOC '99: Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 668–678, 1999. 2, 4, 5, 35

[34] Howard Karloff, Subhash Khot, Aranyak Mehta, and Yuval Rabani. On earthmover distance, metric labeling, and 0-extension. In *STOC '06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 547–556, 2006. 2, 4, 36

[35] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations. New York: Plenum*, pages 85–103, 1972. 57

[36] Alexander Karzanov. Minimum 0-extension of graph metrics. *Europ. J. Combinat.*, 19:71–101, 1998. 36

[37] Alexander Karzanov. A combinatorial algorithm for the minimum $(2, r)$-metric problem and some generalizations. *Combinatorica*, 18:549–569, 1999. 36

[38] Subhash Khot. On the power of unique 2-prover 1-round games. In *IEEE Conference on Computational Complexity*, page 25, 2002. 2, 5, 7

[39] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. 5, 7, 18, 50, 51

[40] 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. IEEE Computer Society, 2005. 48, 62, 87, 88

[41] Jon Kleinberg and Éva Tardos. Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields. *J. ACM*, 49(5):616–639, 2002. 4, 5, 35, 36, 37, 41

[42] Robert Krauthgamer, James R. Lee, Manor Mendel, and Assaf Naor. Measured descent: A new embedding method for finite metrics. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 434–443, 2004. 36

[43] Amit Kumar, Rajsekar Manokaran, Madhur Tulsiani, and Nisheeth K. Vishnoi. On lp-based approximability for strict csps. In *SODA*, pages 1560–1573, 2011. 46

[44] Jean B. Lasserre. An explicit equivalent positive semidefinite program for nonlinear 0-1 programs. *SIAM Journal on Optimization*, 12(3):756–769, 2002. 12

[45] James R. Lee and Assaf Naor. Extending lipschitz functions via random metric partitions. *Inventiones Mathematicae*, 160(1):59–95, 2005. 36

[46] L. Lovsz and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM JOURNAL ON OPTIMIZATION*, 1:166–190, 1991. 12

[47] Cláudio Lucchesi and Daniel H. Younger. A minimax theorem for directed graphs. *Journal London Math. Society*, 17:369–374, 1978. 57

[48] Rajsekar Manokaran, Joseph (Seffi) Naor, Prasad Raghavendra, and Roy Schwartz. Sdp gaps and ugc hardness for multiway cut, 0-extension and metric labelling. *In STOC '08: Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008. 7, 34, 41, 87

[49] Monaldo Mastrolilli, Maurice Queyranne, Andreas S. Schulz, Ola Svensson, and Nelson A. Uhan. Minimizing the sum of weighted completion times in a concurrent open shop. In *Manuscript*, 2009. 48

[50] Elchanan Mossel. Gaussian bounds for noise correlation of functions and tight analysis of long codes. In *FOCS*, pages 156–165. IEEE Computer Society, 2008. 9, 10, 23

[51] Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: invariance and optimality. In *FOCS*, pages 21–30. IEEE Computer Society, 2005. 9, 10, 16, 17, 20, 23

[52] Alantha Newman. Approximating the maximum acyclic subgraph. Master's thesis, MIT, June 2000. 57, 62

[53] Alantha Newman. Cuts and orderings: On semidefinite relaxations for the linear ordering problem. In *APPROX-RANDOM*, pages 195–206, 2004. 62

[54] Christos H. Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. Syst. Sci.*, 43(3):425–440, 1991. 57

[55] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? *In STOC '08: Proceedings of the 40th ACM Symposium on Theory of Computing*, 2008. 7, 45, 47, 48, 63, 64, 74, 81, 82, 87

[56] Prasad Raghavendra. *Limits to Approximability : Understanding the power of Semidenite Programming.* PhD thesis, University of Washington, Seattle, June 2009. 30, 31, 64

[57] Prasad Raghavendra and David Steurer. How to round any csp. In *FOCS*, pages 586–594, 2009. 48

[58] Prasad Raghavendra and David Steurer. Towards computing the grothendieck constant. *In SODA '09: Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2009. 7

[59] V. Ramachandran. Finding a minimum feedback arc set in reducible flow graphs. *J. Algorithms*, 9:299–313, 1988. 57

[60] Ran Raz. A parallel repetition theorem. *SIAM J. Comput.*, 27(3):763–803, 1998. 5

[61] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. Tight integrality gaps for lovasz-schrijver lp relaxations of vertex cover and max cut. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 302–310. ACM, 2007. 55

[62] Alexander Schrijver. *Theory of linear and integer programming.* Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1999. 11

[63] Paul Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995. 58

[64] H. D. Sherali and W. P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM J. Discrete Math.*, 3(3):411–430, 1990. 12, 49