Tel Aviv University
Raymond and Beverly Sackler
Faculty of Exact Sciences
School of Computer Sciences

Expansion
and
Approximability

Thesis submitted for the degree of Doctor of Philosophy by

Oded Schwartz

Under the supervision of Prof. **Shmuel Safra** and Prof. **Amnon Ta-Shma**

Submitted to the Senate of Tel-Aviv University
October 2007

To all my parents, sisters and brothers and to Anat.

Acknowledgment

To my advisor Muli Safra, whose prolific ideas and lucid views on math, the universe, and everything guided me through this research. His enthusiastic, optimistic, easygoing approach to settling fundamental hard problems, and his constant patience, were more than I could hope for.

To my advisor Amnon Ta-Shma, whose ability to know the right thing to say and do, when tackling theoretical problems and when dealing with earthly dilemmas, have been so enlightening and valuable to me.

To Noga Alon for his encouragement, support and sound advice, that made the difference.

To Hanoch Levy, for putting the fun in fundamental course teaching.

To Joseph S. B. Mitchell for inspiring discussions, encouragement and support.

To Amitai Armon, Adi Avidor, Avi Ben-Aroya, Elad Hazan, Asaf Shapira - thank you for fruitful collaboration and joyful joint work.

To Vera Asodi, Guy Kindler, Yoad Lustig, Eran Rom and Shakhar Smorodinsky, thank you for making difficult times easier and for insightful and far-reaching advice.

To all my parents, sisters and brothers, thank you for everything.
And to Anat, for all the love and support.

Abstract

Expanders are graphs that are sparse, yet highly connected. In this thesis, we consider an elementary algorithm for constructing expanders, how to use them for proving hardness of approximation, how to utilize expanders for obtaining better approximations and their relation to the parallel-repetition technique.

In Chapter 2 we presents two variants of the Parallel-Repetition. One that preserves uniqueness but works only for good expanders and union of disjoint expanders, and the other that works for any instance but does not preserve uniqueness. We show that the two variants of the Parallel-Repetition technique perform "optimally", i.e, the success probability decays exponentially fast with k, regardless of the alphabet size and with no power on ε, albeit only down to some constant error probability of the generated instance.

Such analysis also has algorithmic consequences: it allows converting an approximation algorithm of expander instances for one set of parameters (error, size of alphabet and approximation ratio) into another, so that an optimal algorithm for one such set of parameters suffices to obtain optimal approximations for all the others. This chapter is based on the paper [SS07].

In Chapter 3 we describe a short and easy to analyze construction of constant-degree expanders. Expanders are some of the most widely used objects in theoretical computer science. Many algorithm were suggested for constructing such graphs (see Chapter 3 for further discussion).

Our construction relies on the replacement product, applied by [RVW02] to give an iterative construction of bounded-degree expanders. Here we give a simpler construction, which applies the replacement product, only a constant number of times, to turn the Cayley expanders of [AR94], whose degree is polylog n, into constant degree expanders. This allows us to prove the required expansion using a new simple combinatorial analysis of the replacement product (instead of the spectral analysis used in [RVW02]). This chapter is based on the paper [ASS07].

In Chapter 4 we study the complexity of bounded packing problems, mainly the problem of k-SetPacking. We prove that k-SetPacking cannot be efficiently approximated to within a factor of $O(\frac{k}{\ln k})$ unless $P = NP$. This improves the previous factor of $\frac{k}{2^{\Omega(\sqrt{\ln k})}}$ by Trevisan [Tre01].

This result extends to the problem of k-DimensionalMatching and the problem of IndependentSet in $(k+1)$-claw-free graphs. To this end we introduce and studey the notion of hyper-disperser. This chapter is based on the paper [HSS06].

Contents

Chapter 1

Introduction

A Short History

Distinguishing between tractable problems —those computable in polynomially bounded time— and the intractable ones —those that need more time resources— remains one of the main goals of computational complexity. Cook, Karp and Levin [Coo71, Kar72, Lev73] demonstrated in the early 70's that a large class of natural combinatorial problems, whose tractability has not been settled, are all equivalent in this respect: either all are tractable or none of them is. This class is known as NP-Complete.

The tractability of solving those problems is the question whether $P = NP$. This question is considered the most fundamental open question of computer science. Therefore, when approaching an optimization problem that is known to be NP-Complete, one does not expect to find an optimal solution, as this would resolve the P vs. NP open question.

Much research effort is therefore invested in obtaining polynomial time approximation algorithms that guarantee a "good-enough" solution for *every* input.

However, for many problems, an approximate solution is as hard to obtain as an optimal one. This fact is interpreted as inapproximability for those problems. Many inapproximability proofs rely on the seminal PCP Theorem [FGL$^+$96, AS92, ALM$^+$98, Din07].

For some problems, this fundamental theorem strengthen the inapproximability factors, allowing, in some cases, a sharp threshold. That is, there is a factor so that efficiently approximating the problem to within this factor is tractable, but approximating it even slightly better is NP-hard.

Nevertheless, the complexity of approximating several problems was not thus resolved, and a gap still exists between the known efficient approximation guarantee and the best NP-hardness factor demonstrated.

1.1 On Parallel-Repetition, Unique-Game and Max-Cut

Between P and NP-hard. Consider a problem that is neither known to be in P nor is it known to be NP-hard. An alternative option to consider is that it may be neither in P nor in NP-hard. Unless $P = NP$, there are such problems. That is, let $NPI = NP \setminus (P \cup NP - hard)$; then if $P \neq NP$ then $NPI \neq \emptyset$ [Lad75].

No natural problem has yet been proven to be in NPI, even assuming NPI exists, i.e, $P \neq NP$. There are, however, a few promising, well studied candidates. For example, approximating ClosestVector and ShortedtVector problems to within factors of some polynomial range (see [Sch87, Ban93, GG00, AR05]), GraphIsomorphism [BHZ87] and Factoring (the decision version). These problems are not known to be in P, while proving any of them to be NP-hard would imply the collapse of the polynomial time hierarchy.

For any NPI prospective problem A, define the class A-hard to be the class of all problems having a polynomial time reduction from A, and the equivalence class A-Complete to be the class of all problems having a polynomial time reduction to and from A. Showing reductions among NPI candidates would help sort the complexity of these problems.

Such reductions have been shown between approximation problems that are neither known to be in P, nor known to be in NP-hard. Such recently popular NPI candidate to reduce from, is the *Gap-UniqueGame*- $[\varepsilon, 1-\varepsilon]$ (for an arbitrarily small $\varepsilon > 0$). Denote its hardness class by UG-hard*. Khot [Kho02] conjectured this problem to be NP-hard, however, no evidence for this problem being either in P or NP-hard has been shown. When there is a gap between the best NP-hardness of approximation factor and the known approximation guarantee, a tight threshold may sometimes exist between P and UG-hard.

Following Feige, Kindler, and O'Donnell (see [FKO07] , section "Strong Parallel-Repetition

*Indeed *Gap-UniqueGame*- $[\varepsilon, 1-\varepsilon]$ is a different problem for every $\varepsilon > 0$. Saying that A is UG-hard denotes the fact that there exists an $\varepsilon > 0$ so that *Gap-UniqueGame*- $[\varepsilon, 1-\varepsilon] \leq_P A$

Problem"), we would like to consider MaxCut as a possible substitute for UniqueGame. By the reduction of Khot, Kindler, Mossel and O'Donnell [KKMO04], it is already known that *Gap-MaxCut*-$[1 - c\sqrt{\varepsilon}, 1 - \varepsilon]$, (where c is any constant smaller than $\frac{2}{\pi}$) is UG-hard. Showing a reduction in the other direction would imply the complexity of UniqueGame and all problems shown hard for it relies on the hardness of MaxCut.

Assuming such a reduction, showing an efficient algorithm for UniqueGame requires "only" improving on Goemans-Williamson polynomial time approximation algorithm [GW95] for MaxCut. On the other hand, proving all these problems NP-hard must imply a meaningful improvement on the technique of Trevisan *et al.* [TSSW00] and Håstad's NP-hardness factor for MaxCut [Hås01]. Both of these tasks seem clearer for the MaxCut problem than for the UniqueGame problem, and well beyond current techniques.

A possible technique for proving this conjecture is to apply Parallel-Repetition to Gap-MaxCut and obtain Gap-UniqueGame instance. As is later shown, the current parameters known for Parallel-Repetition (by [Raz98, Hol07]) are not strong enough to be applied here. Such an analysis of Parallel-Repetition may in fact exist, which would imply the previous conjecture, i.e, a MaxCut-hardness for all UG-hard problems.

Parallel-Repetition

The kth Parallel-Repetition of a ConstraintGraph instance U is an instance $U^{\otimes k}$ of ConstraintGraph problem, where the new vertices are k-tuples of the original vertices, two new vertices are connected if the k corresponding edges exist in U and the constraints are naturally defined as the disjunction of the k corresponding original constraints. Note that the uniqueness is preserved, namely, the Parallel-Repetition of a UniqueGame instance is a UniqueGame instance.

If U is $1 - \varepsilon$ satisfiable, then $U^{\otimes k}$ is at least $(1 - \varepsilon)^k$ satisfiable, as one can take an assignment to $U^{\otimes k}$ that is entirely consistent with the optimal assignment to U. This assignment is not necessarily the best one [FRS90].

There is, however, a qualitatively similar upper-bound on the satisfiability of $U^{\otimes k}$. Raz [Raz98] and Holenstein [Hol07] showed that $U^{\otimes k}$ is at most $\left(1 - \frac{\varepsilon^3}{6000}\right)^{\frac{k}{2\lg|\Sigma|}}$ satisfiable, where $|\Sigma|$ is the alphabet size of U. However, the cubic power of ε in this upper-bound prohibits using this analysis for showing MaxCut-hardness for UniqueGame.

Therefore, for this goal (and for other uses, as we later discuss) we are interested in improving the parameters of the upper-bound analysis of the Parallel-Repetition (the power on ε, the dependency on $|\Sigma|$), or alternatively, suggesting other uniqueness preserving amplification techniques which would allow such improvements.

1.1.1 Our Contribution

This chapter is based on the paper [SS07]. It presents two variants of the Parallel-Repetition: the Noisy-Parallel-Repetition and the Expanding-Parallel-Repetition. In the first variant we add self-loops (with equality constraints) to the original instance before performing the original Parallel-Repetition. In the second variant, in addition to the self-loops, we add a graph with large spectral-gap, with trivial constraints (i.e, constraints that are always satisfied), before performing the original Parallel-Repetition.

The Noisy-Parallel-Repetition variant preserves the uniqueness property, but works only for good expanders (or union of disjoint expanders). The Expanding-Parallel-Repetition variant works well for any instance, but does not preserve the uniqueness property.

We show that the two variants above perform “optimally” (i.e, the success probability decays exponentially fast with k, regardless of the alphabet size and with no power on ε) albeit only down to some constant error probability of the generated instance.[†]

Both variants are not sufficient for the goal of proving MaxCut-hardness for UniqueGame. For this goal we need a uniqueness preserving amplification that works for any graph, can be utilized to obtain any (arbitrarily small) soundness, and has an “optimal” amplification rate.

Note that, in contrast to [Raz98, Hol07], our proof does not necessarily work for the 2-prover model, but only for the purpose where Parallel-Repetition is most often applied: amplifying hardness of approximation. We also show some algorithmic application to these variants.

[†]Recently Raz [Raz08] has shown that applying the Parallel-Repetition technique to MaxCut instances with arbitrarily large k does not yield ”optimal” parameters, namely it cannot be used for proving MaxCut-hardness for UniqueGame. See Chapter 2 for details.

1.2 An Elementary Construction of Constant-Degree Expanders

Expanders are graphs, that are simultaneously sparse, yet highly connected, in the sense that every cut contains (relatively) many edges. A d-regular graph $G = (V, E)$ is a δ-expander if for every set $S \subseteq V$ of size at most $\frac{1}{2}|V|$ there are at least $\delta d|S|$ edges connecting S and $\overline{S} = V \setminus S$.

Another widely used notion of expansion is based on algebraic properties of a matrix representation of the graph. Let $G = (V, E)$ be an n-vertex d-regular graph, and let A be the adjacency matrix of G, that is, the $n \times n$ matrix, with $A_{i,j}$ being the number of edges between i and j. It is easy to see that 1^n (the uniform vector) is an eigenvector of A with the largest eigenvalue d, and that this is the only eigenvector with this eigenvalue iff G is connected. We denote by $\lambda_2(G)$ the second largest eigenvalue of A. It is easy to see that $\lambda_2(G) = \max_{0 \neq x \perp 1^n} \langle Ax, x\rangle / \langle x, x\rangle$. Let $\gamma = 1 - \frac{\lambda_2}{d}$ be the (normalized) spectral-gap of G. By [Alo86, AM85, Dod84] we know a quantitative relation between the edge expansion and the spectral-gap:

$$\frac{\gamma}{2} \leq \delta \leq \sqrt{2\gamma}$$

1.2.0.1 Usefulness

Expanders are some of the most widely used objects in theoretical computer science, and have also found many applications in other areas of computer-science and mathematics. See the survey of Hoory et. al. [HLW06] for a discussion of several applications and references.

Expanders of Constant Degree. The most useful expanders are those with constant degree. A priori, it is not clear that constant-degree expanders even exist. Pinsker [Pin73] established their existence, using a probabilistic argument. In most applications, however, one needs to be able to efficiently construct constant degree expanders explicitly.

Explicit Expanders. There are two notions of constructibility of d-regular expanders. The first (weaker) notion requires the n-vertex graph to be constructible in polynomial time in its size. The second (stronger) notion requires that given a vertex v and $i \in [d]$ it would be possible to generate the i^{th} neighbor of v in time polynomial in the representation of v, namely, $\mathrm{Poly}(|v|) = \mathrm{Poly}(\log n)$. Such an expander is said to be *fully explicit*.

In applications where one needs to use the entire graph, it is often enough to use the weaker notion. However, in such cases (e.g. in certain reductions) one frequently needs to be able to construct a graph of a given size n.

In other cases, where one needs only part of the expander (e.g., when performing a random walk on a large expander) one usually needs the stronger notion of fully explicitness. However, in these cases it is usually enough to be able to construct an expander of size $\mathrm{Poly}(n)$, as what we are interested in is actually the logarithm of the size of the graph.

Explicit Expanders and Spectral Analysis. Margulis [Mar73] and Gabber and Galil [GG81] were the first to efficiently construct constant degree expanders. Following was a sequence of works that culminated in the construction of Lubotzky, Phillips and Sarnak [LPS88] and Margulis [Mar88] of Ramanujan Graphs. These constructions rely (directly or indirectly) on estimations of the second largest eigenvalue of the graphs, and some of them, rely on deep mathematical results.

A simpler, iterative construction was given by Reingold, Vadhan and Wigderson [RVW02]. This construction relies on proving the expansion of the graphs by estimating their eigenvalues, and is the first construction of constant degree expanders with relatively elementary analysis.

1.2.1 Our Contribution

This chapter is based on the paper [ASS07]. We describe a short and easy to analyze construction of constant-degree expanders. The construction yields an explicit constant-degree expander of any desired size. A slight variation of the construction gives a fully-explicit constant-degree expander of size that is at most polynomially larger than the input parameter n (which, as mentioned above, suffices for most cases where fully-explicitness is required).

The construction relies on the replacement product, and on the poly-logarithmic degree expander construction of [AR94]. As our construction uses the replacement product only a constant number of times, this enables us to prove the required expansion using a simple combinatorial analysis of the replacement product (instead of the spectral analysis used in

[RVW02]).

1.3 k-Set Packing and Related Problems

In Chapter 4 we consider the SetPacking problem. The input to SetPacking is a set of elements and a family of subsets of the elements. The objective is to find a maximal number of disjoint subsets. We examine the change in the complexity of this problem when bounds are applied to it. In particular, we try to illustrate the connection between the bounded parameters (e.g, sets size, occurrences of elements) and the complexity of the bounded problem.

It is already known that bounded variants of optimization problems are often easier to approximate than the general, unbounded problems. The IndependentSet problem illustrates this well: it cannot be approximated to within $O(N^{1-\varepsilon})$ unless $P = NP$ [Hås99, Zuc07]. Nevertheless, once the input graph has a bounded degree d, much better approximations exist (e.g, a $\frac{d \log\log d}{\log d}$ approximation by [Vis96]).

The general problem of SetPacking has been extensively studied (for example [Wig83, BYM84, BH92, Hås99, Zuc07]). Quite tight approximation algorithms and inapproximability factors are known for this problem. Håstad [Hås99] proved that SetPacking cannot be approximated to within $O(N^{1-\varepsilon})$ unless $NP \subseteq ZPP$ (for every $\varepsilon > 0$, where N is the number of sets). Recently Zuckerman [Zuc07] showed the same inapproximability factor under $P \neq NP$ assumption. The best approximation algorithm achieves an approximation ratio of $O(\frac{N}{\log^2 N})$ [BH92]. In contrast, the case of bounded variants of this problem seems to be of a different nature.

Bounds on SetPacking. k-SetPacking is the problem of SetPacking where the size of each subset is bounded by k. Another natural bound is the colorability of the input. That is, the minimal number of colors needed for coloring the elements, so that no two elements of the same color participate in a joint subset. We denote this problem by k-DimensionalMatching.

These bounded variants of SetPacking are known to admit approximation algorithms better than their general versions, the quality of the approximation being a function of the bounds (see Chapter 4 for details). With some abuse of notations, one can say that hardness of approximation factor of SetPacking is a monotonous increasing function in each of the bounded parameters: the sets size, the number of occurrences of each element and the colorability. For example, inapproximability factor for instances where each set is of size at most 3 holds for instances where the bound is 4.

For large k values, we are usually interested in the asymptotic dependence of the approximation ratio (and inapproximability factor) on k. Currently, the best polynomial time approximation algorithm for k-SetPacking achieves an approximation ratio of $\frac{k}{2}$ [HS89]. This is, to date, the best approximation algorithm for k-DimensionalMatching as well. Alon et al. [AFWZ95] proved that k-SetPacking is NP-hard to approximate to within $k^c - \varepsilon$ (for some $c > 0$ and for suitably large k). This was later improved [Tre01] to a factor of $\frac{k}{2^{\Omega(\sqrt{\ln k})}}$.

1.3.1 Our Contribution

This chapter is based on the paper [HSS06]. We improve the known inapproximability factor for the variant k-SetPacking, and show that it is NP-hard to approximate k-SetPacking to within $O\left(\frac{k}{\ln k}\right)$.

This result is then extended to hold for k-DimensionalMatching (and shown to hold for IndependentSet in $(k+1)$-claw-free graphs).

For proving this inapproximability, we introduce and study the notion of Hyper Disperser which is a the natural generalization of a disperser graph, applied to hyper graphs. This object is used to enforce consistency. The optimality of the hyper-dispersers obtained, allows utilizing it in a way that performs better than using many dispersers in parallel.

1.4 Collaborators

Chapter 2 is based on the paper [SS07] written in collaboration with Muli Safra. Chapter 3 is based on the paper [ASS07] written in collaboration with Noga Alon and Asaf Shapira. Chapter 4 is based on the paper [HSS06] written in collaboration with Elad Hazan and Muli Safra. The following papers are not included in this thesis: the paper [BASTS07] written with Avi Ben-Aroya and Amnon Ta-Shma, the paper [AAS06] written in collaboration with Adi Avidor and Amitai Armon, and the paper [SS05] written in collaboration with Muli Safra.

1.5 General Preliminaries

1.5.1 Expanders

Expanders are graphs that are simultaneously sparse, yet highly connected, in the sense that every cut contains (relatively) many edges.

The relevance of graph eigenvalues to its expansion is well studied. The relation between the expansion of the graph and its second largest eigenvalue is often used to bound the mixing time of random walks. Utilizing the expansion of a graph or its mixing time proves to be useful in the context of NP-hardness as well (e.g, [PY88, Din07]).

Let $G = (V, E)$ be a d-regular graph with adjacency matrix M. The normalized spectral-gap γ of G is $\gamma(G) = 1 - \frac{\lambda_2(A)}{d}$ where $\lambda_2(A)$ is the second-largest eigenvalue of A. We say that G as γ-expander.

The relative edge expansion h of G is $h(G) = \min_{S \subseteq V, |S| \leq \frac{1}{2}|V|} \frac{|E(S, V \setminus S)|}{d|S|}$.

Theorem 1.1 (Expander theorem). *[Alo86, AM85, Dod84] Let $G = (V, E)$ be a γ-expander, with relative edge expansion h. Then,*

$$\frac{1}{2}\gamma \leq h \leq \sqrt{2\gamma}$$

1.5.2 Approximations

We usually use the convention of approximation ratio larger than 1 for maximization problems, and smaller than 1 for minimization problems. Formally,

Let O be an optimization problem, let ALG be an approximation algorithm for this problem, and let I be an instance of O. Denote by $OPT(I)$ the optimal solution for I, and by $ALG(I)$ the solution found by ALG applied to I. Denote by r the ratio of these two: $r = \max\{\frac{|ALG(I)|}{|OPT(I)|}, \frac{|OPT(I)|}{|ALG(I)|}\}$. Then we say that ALG is a c-approximation for O if for every input I, $r \leq c$. For maximization problems, we sometimes refer to the approximation ratio (and inapproximability factor) as $c' = \frac{1}{c}$ (thus it is smaller than 1).

1.5.3 Gap-Problems

In order to prove inapproximability of an optimization problem, one usually defines a corresponding gap problem. Recall the definition of gap-problems:

Definition 1.2 (Gap Problem - Minimization). *Let O be a minimization problem. Gap-O-$[\alpha, \beta]$ is the following decision problem:*
Given an input instance, decide whether

- *there exists a solution of size at most α, or*
- *every solution of the given instance is of size larger than β.*

If the size of the solution resides between these values, then any output suffices.

Similarly,

Definition 1.3 (Gap problem - Maximization). *Let O be a maximization problem. Gap-O-$[\alpha, \beta]$ is the following decision problem:*
Given an input instance, decide whether

- *there exists a solution of fractional size at least β, or*
- *every solution of the given instance is of fractional size smaller than α.*

If the size of the solution resides between these values, then any output suffices.

Clearly, for any optimization problem, if *Gap*-O-$[\alpha, \beta]$ is NP-hard, then it is NP-hard to approximate O to within any factor smaller than $\frac{\beta}{\alpha}$.

Chapter 2

On Parallel-Repetition, Unique-Game and Max-Cut

2.1 Introduction

Proving UG-hardness of approximation where NP-hardness of approximation is not known, has recently been proven to be a fruitful technique. In some cases, the UG-hard factors match the known approximation guarantee. For example, the best NP-hardness of approximating VERTEXCOVER is 1.3606 [DS02], where the best UG-hardness is $2 - \varepsilon$ [KR03], matching the known 2-approximation for this problem. Another example is the MAXCUT problem. The best NP-hardness of approximation factor for this problem is $\frac{16}{17}$ [TSSW00, Hås01], where the UG-hardness of approximation [KKMO04] matches the Goemans-Williamson approximation constant (≈ 0.87856) [GW95]. For more UG-hard problems and further discussion of UG-hardness see overview in [Kho05].

Approximability of MAXCUT. For a MAXCUT instance whose optimal cut contains almost all edges, one can efficiently find a cut only slightly smaller than the optimal [GW95]. Therefore, the approximation ratio is close to 1 and it may make more sense to measure the approximability in terms of the unsatisfied fraction, rather than the satisfied fraction. In other words, to consider the dual problem - MINUNCUT.

In most cases, when considering approximation ratios, they are either constant or stated as a function of the input size. However, for MINUNCUT the (in)approximation factor c_ε

is a function of ε - the fractional size of the optimal solution.

For MinUnCut there is a sharp threshold between P and UG-hardness around $\frac{2}{\pi} \cdot \frac{1}{\sqrt{\varepsilon}}$ [GW95, KKMO04], but NP-hardness only for factors smaller than $\frac{16}{15}$ (the NP-hardness is by a simple reduction from MaxCut inapproximability of Håstad and Trevisan *et al.* [Hås01, TSSW00] , see Appendix 2.6.1)*.

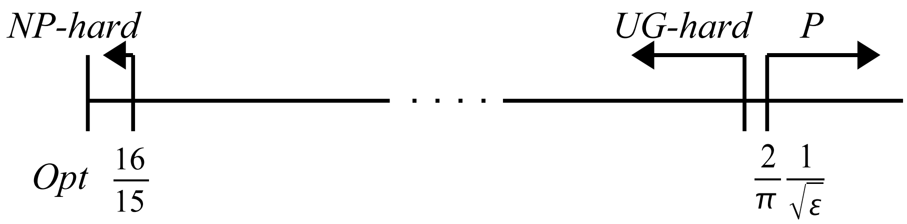

Figure 2.1: The Complexity of Approximating MinUnCut for various ratios. ε is the fractional size of the optimum.

MaxCut-hardness vs. UG-hardness

Denote by MaxCut-hard and MaxCut-complete the hardness and completeness classes of *Gap-MaxCut*-$[1 - c\sqrt{\varepsilon}, 1 - \varepsilon]$, where c is some constant smaller than $\frac{2}{\pi}$. By the reduction from UniqueGame to MaxCut of Khot *et al.* [KKMO04], it is known that any MaxCut-hard problem is UG-hard as well. However, a reduction from MaxCut to UniqueGame is not known (in fact, UniqueGame is not known to be hard for any other problem). Note that, technically speaking, MaxCut is a special case of UniqueGame. However, for UniqueGame, we are interested in soundness arbitrarily close to 0, where in MaxCut we are intersected in soundness close to 1. This seemingly makes MaxCut harder than UniqueGame. Therefore, proving MaxCut-hardness for a problem appears to be a stronger result than showing it is UG-hard.

It would therefore be of great interest to show such a reduction. This would show that proving UG-hardness is equivalent to proving MaxCut-hardness. Let us consider the possibility of such a reduction, namely:

*A few algorithms guarantee an approximation ratio for MinUnCut that depend on the input size (e.g, $O\left(\sqrt[3]{\frac{\lg n}{\varepsilon^2}}\right)$ of Trevisan [Tre05] and $O\left(\sqrt{\frac{\lg n}{\varepsilon}}\right)$ of Gupta and Talwar [GT06]). Note however, that such algorithms are not relevant for constant ε, but only for sub-constant ε.

Conjecture 2.1 (MaxCut **conjecture).** *There exists a polynomial-time reduction from* MaxCut *to* UniqueGame *such that for every constant* $\varepsilon' > 0$ *there exist constants* $c_{\varepsilon'}$ *and* $\varepsilon > 0$ *so that*

$$\textit{Gap-MaxCut-}[1 - c_{\varepsilon'} \cdot \varepsilon, 1 - \varepsilon] \leq_p \textit{Gap-UniqueGame-}[\varepsilon', 1 - \varepsilon']$$

Stated otherwise, assuming this conjecture, one only needs to consider the approximability of MinUnCut in order to understand the complexity of UniqueGame. If it can be approximated better than Goemans-Williamson approximation [GW95] then the UniqueGame-conjecture is false. If the NP-hardness of MinUnCut [TSSW00, Hås01], can be significantly improved then the UniqueGame-conjecture is right. And if MinUnCut can be shown to be in NPI (which would make it the first natural problem in NPI) then the UniqueGame-conjecture is false, but still UG-hard problems are not in P, unless $P = NP$.

Proving the MaxCut conjecture is the grand objective. Here we only manage to prove special cases of it. One way of showing a reduction as stated in the MaxCut conjecture may be to apply Parallel-Repetition [Raz98] assuming optimal parameters. We next consider the Parallel-Repetition, its parameters and some of its variations.

Hardness Reduction via Parallel-Repetition

The Parallel-Repetition is a generic technique that is used for amplifying two-provers interactive proofs and hardness of approximation problems. It is often used as a first step of NP-hardness of approximation reductions, in order to reduce the soundness of a given gap-problem to an arbitrarily small constant (e.g, [Hås01, DS02, DGKR05]). The parameters of this technique are of independent interest and have impact on the its usability for various applications.

Applying the kth Parallel-Repetition to a MaxCut instance U yields a UniqueGame instance $U^{\otimes k}$ where each vertex is k vertices of U, each edge corresponds to k edges of U, and the constraints are defined naturally. Assume that the optimal solution for U satisfies a $(1-\varepsilon)$ fraction of the constraints. Then, the optimal solution for $U^{\otimes k}$ satisfies at least $(1-\varepsilon)^k$ fraction of the constraints (as one can take an assignment that is anywhere consistent with the optimal solution of U). This assignment was thought to be optimal for some time, but in fact better assignments —that are not product assignments— sometimes exist [FRS90]. However, qualitatively speaking, the tightest known upper-bound on the successes probability does decay in an exponential rate (in k), as shown by Raz [Raz98] and by Holenstein [Hol07]. For the general ConstraintGraph problem it is $\left(1-\frac{\varepsilon^3}{6000}\right)^{\frac{k}{2\lg|\Sigma|}}$.

Alas, due to the cubic power of ε in the upper-bound, this guarantee is not sufficient for some applications, in particular for the gap problem *Gap-MaxCut-* $[1-c\sqrt{\varepsilon}, 1-\varepsilon]$: the upper bound on the soundness of $U^{\otimes k}$ might be larger than the lower bound on its completeness (regardless of k) †.

2.1.0.1 The Parameters of Parallel-Repetition.

When considering an error-probability amplification technique, such as the Parallel-Repetition, a few attributes are of interest. We next consider these attributes of Parallel-Repetition and related techniques.

Amplification Rate. Consider an upper-bound on the amplification rate of the kth Parallel-Repetition of $1-\varepsilon$ satisfiable UniqueGame. If it is of the form $(1-\varepsilon^\alpha)^{ck}$ then, by the lower bound of $(1-\varepsilon)^k$ we have $\alpha \geq 1, c \leq 1$ and we are interested in as small as possible α and as large as possible c.

The Rate Depends on the Alphabet Size. In the upper-bound on Parallel-Repetition of Raz [Raz98], c is a constant that depends on $|\Sigma|$ - the alphabet size of the input instance.

Raz [Raz95] also proves that for the general ConstraintGraph instances, which are far from being completely satisfiable, the exponent of the upper-bound has to depend on $|\Sigma|$ (it is multiplied by a factor of $O(\frac{\lg\lg|\Sigma|}{\lg|\Sigma|})$). That is, the $\frac{1}{\lg|\Sigma|}$ factor in the exponent of the upper-bound of Raz [Raz98] cannot be significantly improved. This bound on the exponent, however, does not necessarily hold for almost completely satisfiable instances for UniqueGame instances and for other variants of the Parallel-Repetition.

Indeed, Feige and Kilian [FK00] consider a variant of the Parallel-Repetition, which they call the "miss-match" form. They show that the amplification rate of their variant does not depend on $|\Sigma|$ (though their amplification rate is not tight, i.e, not of the form $(1-\varepsilon^\alpha)^{ck}$).

†In fact, for the special case of MaxCut, the loss can be shown to be only square (see [FL92, FKO07]). However, this square loss is still too large for allowing the reduction from MaxCut to UniqueGame.

The Size of the Output Instance and its Alphabet. The kth Parallel-Repetition increases both the size of the instance and its alphabet by a power of k. For some applications (see for example Section 2.4) we would like the increase of the instance size and the alphabet size to be as small as possible for a given error probability amplification.

Error probability: Source and Target. An amplification technique may have limitations on the error probability it can handle in the input instance and the error probability it can guarantee in the generated instance. The Parallel-Repetition handles any arbitrarily small constant error probability δ in the input instance, and generates an instance with an arbitrarily small constant success probability. However, if we are interested in error probability δ of the input which is polynomially small and success probability of the output which is polynomially small, then the generated instance $U^{\otimes k}$ is of exponential size, and therefore the procedure is no longer polynomial.

The PCP theorem can be viewed as an amplification technique that handles well input instances with polynomially small error probability. Note that in Dinur's proof of the PCP theorem, the success probability can be reduced to some constant (larger than $\frac{1}{2}$), but not to an arbitrarily small constant [Din07, Bog05].

Uniqueness Preservation. For the purpose of showing a reduction from MaxCut to UniqueGame, the amplification applied has to preserve the uniqueness property of the constraints. The uniqueness is indeed preserved in the original Parallel-Repetition. In the variant of Feige and Kilian, the output instance is not UniqueGame, even if the input instance is.

2.1.0.2 "Optimal" Parallel-Repetition.

Ideally we would like an "Optimal" Parallel-Repetition. That is, a uniqueness preserving amplification technique that guarantees a tight upper-bound, is independent of the alphabet size and of the spectral-gap and works for an arbitrarily small constant error probability of the input and an arbitrarily small constant success probability of the output.

After the initial submission of this thesis, Raz [Raz08] has given a counter example for an "optimal" performance of Parallel-Repetition. He has demonstrated that the Parallel-Repetition (and the Noisy-Parallel-Repetition) does not perform "optimally" for arbitrarily large k. In particular, when applied to the odd-cycles game (already considered in [FKO07]), a $1-\varepsilon$ satisfiable instance yields a $(1-\varepsilon^2)^{\Theta(k)}$ satisfiable unique-game instance.

This contradicts a conjecture regarding the optimality of Noisy-Parallel-Repetition which appeared in a previous version of this thesis and in [SS07], and means that neither Parallel-Repetition nor Noisy-Parallel-Repetition provide an immediate reduction for proving MaxCut-hardness for UniqueGame. There may still, however, be some hope that conjecture 2.1 is right; that there is some other uniqueness preserving amplification technique, so that the upper-bound on the success probability is at most $(1-\delta^\alpha)^{ck}$ for some constant $\alpha < 2$, and thus it suffices as a reduction from MaxCut to UniqueGame.

2.1.1 Variants of the Parallel-Repetition

This chapter presents two variants of the Parallel-Repetition: the Noisy-Parallel-Repetition and the Expanding-Parallel-Repetition.

Noisy-Parallel-Repetition. In the first variant we add self-loops with equality constraints to the original graph G, before performing the original Parallel-Repetition. Their relative weight is P_{loop} (say, $P_{loop} = \frac{1}{2}$). Note that adding self-loops changes the success probability from $1-\varepsilon$ to $P_{loop} + (1-P_{loop}) \cdot (1-\varepsilon)$. This means that a completeness close to perfect remains as such, and a soundness close to 0 becomes close to P_{loop}. Clearly, Noisy-Parallel-Repetition preserves uniqueness.

One can think of the kth-Noisy-Parallel-Repetition as follows. The vertices are k-tuples of the original vertices (the same as in Parallel-Repetition). Given a k-vertex $x' = (x_1, ..., x_k)$ its neighbor $y' = (y_1, ..., y_k)$ is generated according to the following distribution. y_i is set to be x_i with probability P_{loop} and to be a random neighbor of x_i in G with probability $1-P_{loop}$. The constraints are naturally defined to be the disjunction of the constraints of the k corresponding edges.

Expanding-Parallel-Repetition. In the second variant, in addition to the self-loops, we add to G a graph H with a large spectral-gap, with trivial constraints (always satisfied), and relative weight P_H, before performing the original Parallel-Repetition. The resulting instance is not a UniqueGame instance, even if the original instance is, due to the constraints of H.

One can think of the kth-Expanding-Parallel-Repetition as follows. The vertices are k-tuples of the original vertices (the same as in Parallel-Repetition). Given a k-vertex $x' = (x_1, ..., x_k)$ its neighbor $y' = (y_1, ..., y_k)$ is generated according to the following distribution. y_i is set to be x_i with probability P_{loop}, to be a random neighbor of x_i in H with probability P_H and to be a random neighbor of x_i in G with probability $1 - P_H - P_{loop}$. The constraints are naturally defined to be the disjunction of the constraints of the corresponding edges from G, and self-loops (the trivial constraints of H don't matter). This means that, unlike the Parallel-Repetition, a k-edge connects two k-vertices $x' = (x_1, ..., x_k)$ and $y' = (y_1, ..., y_k)$ even if only for some of the coordinates i the vertices x_i and y_i are neighbors in G.

2.1.2 Our Results

We show that the two variants of the Parallel-Repetition technique perform "optimally", albeit only down to some constant error probability of the generated instance. That is, for every $k \leq \frac{1}{\varepsilon}$, the success probability decays exponentially fast with k, regardless of the alphabet size and with no power on ε.

The Noisy-Parallel-Repetition variant preserves the uniqueness property, but works only for good expanders (or union of disjoint expanders). The Expanding-Parallel-Repetition variant works well for any instance, but does not preserve the uniqueness property. Unfortunately both variants are not sufficient for the goal of proving MaxCut-hardness for UniqueGame. For this goal we need an amplification that works for any graph, preserves the uniqueness and can be utilized to obtain any (arbitrarily small) soundness.

Theorem 2.2 (Main theorem). *Let U be a $1 - \varepsilon$ satisfiable* ConstraintGraph *instance. Let $\mathbf{U}$ be the kth Noisy-Parallel-Repetition of U, with $P_{loop} = \frac{1}{2}$, where $k \leq \frac{1}{\varepsilon}$.*

Then $\mathbf{U}$ is at most $(1 - \varepsilon)^{c_\gamma k}$ satisfiable where $\gamma = \gamma(U)$ and $c_\gamma = \frac{\gamma}{12800 \cdot \lg \frac{800}{\gamma}}$.

Moreover, if G is a union of expanders, each with eigenvalue-gap at least γ then the same theorem holds (with c_γ multiplied by some constant).

Note that we do not try to optimize c_γ. We later show that the Noisy-Parallel-Repetition is constructive, namely,

Corollary 2.3 (Constructiveness). *Given an assignment to the generated instance of the kth Noisy-Parallel-Repetition that satisfies $1 - \delta$ fraction of its constraint for $\delta < c_\gamma$,* *one can, in polynomial time, compute an assignment to the input instance that satisfies at least a* $1 - \frac{\delta}{c_\gamma \cdot k}$ *fraction of the constraints.*

We then show a similar bound for the Expanding-Parallel-Repetition, with no dependency on $\gamma(U)$.

Corollary 2.4 (Expanding-Parallel-Repetition). *Let* U *be a* $1-\varepsilon$ *satisfiable* ConstraintGraph *instance. Let* $\mathbf{U}$ *be the* k*th Expanding-Parallel-Repetition of* U*, with* $P_{loop} = \frac{1}{2}$ *and* $P_H = \frac{1}{4}$*, where* $k \leq \frac{1}{\varepsilon}$*.*

Then $\mathbf{U}$ *is at most* $(1-\varepsilon)^{ck}$ *satisfiable, where* c *is some universal constant (that does not depend on the spectral-gap or the alphabet size of the input instance).*

The Expanding-Parallel-Repetition is constructive, namely,

Corollary 2.5 (Constructiveness). *Given an assignment* A *to the generated instance of Expanding-Parallel-Repetition that satisfies* $1-\delta$ *fraction of its constraint for* $\delta < c$*, one can, in polynomial time, compute an assignment to the input instance that satisfies at least a* $1 - \frac{\delta}{c \cdot k}$ *fraction of the constraints.*

Note that, in contrast to [Raz98, Hol07], our proof does not necessarily work for the 2-prover model, but only for the purpose where Parallel-Repetition is most often applied: amplifying hardness of approximation. We also show some algorithmic application to these variants.

Algorithmic Applications. We show how one can use the "Optimal" Parallel-Repetition conjecture (or sometimes the "Optimal" Noisy-Parallel-Repetition theorem for expanders) for generalizing approximations for UniqueGame and for amplifying approximations for expanders instances.

Outline

In Section 2.2 we give some preliminaries, including some of the notations we later use. The main theorem, regarding Noisy-Parallel-Repetition, is proved in Section 2.3. We start with the case of expanders (Section 2.3.1), then consider the case of union of disjoint expanders (Section 2.3.2). The Expanding-Parallel-Repetition and a comparison to the "miss-match" form is then considered (Section 2.3.3). Section 2.4 presents some algorithmic applications. Open problems and concluding remarks are discussed in Section 2.5.

2.2 Preliminaries

2.2.1 Graphs.

All graphs considered here are undirected and regular (with possibly parallel edges). For any graph $G = (V, E)$ and two subsets $A, B \subseteq V$ the set $E_G(A, B)$ is all edges $(u, v) \in E$ connecting a vertex $u \in A$ to a vertex $v \in B$. Similarly, $E_G(A)$ denotes the set of all edges $(u, v) \in E$ connecting a vertex $u \in A$ to any vertex of G. We write $E(A, B)$ and $E(A)$ when G is obvious from the context.

When considering the adjacency matrix M_G of a d-regular graph G, we think of the normalized matrix. That is, the sum of each row (and column) is 1, and all eigenvalues are reals in the range $[-1, 1]$. Stated otherwise, M_G is the 0/1 adjacency matrix of G normalized by a factor of $\frac{1}{d}$.

2.2.2 Problems Definition.

Let us now formally define the ConstraintGraph and UniqueGame problems and the Parallel-Repetition theorem.

Definition 2.6. *The* ConstraintGraph *Problem (also known as* GraphLabelling *Problem) is:*
Input: $U = \langle G, \Sigma, C \rangle$ *where* $G = (V, E)$ *is an undirected graph,* Σ *a finite alphabet and* $C = \{c_e\}_{e \in E}$ *a set of constraints (one constraint for each edge) where* $c_e \subseteq \Sigma \times \Sigma$.
Objective: Find an assignment $A : V \to \Sigma$ *that maximizes the number of satisfied constraints.*

We say that a constraint c_e *of an edge* $e = (u, v)$ *is satisfied by* A *if* $(A(u), A(v)) \in c_e$.

We denote by $\delta(A(U))$ the fractional size of constraints of U that are unsatisfied by A, that is,

$$\delta(A(U)) = \frac{1}{|C|} \cdot |\{c_{(u,v)} \in C \mid (A(u), A(v)) \notin c_{(u,v)}\}|$$

We denote by $\delta(U)$ (or sometimes by $\varepsilon(U)$) the minimal error of U, i.e, the fraction of unsatisfied constraints in an optimal assignment to U:

$$\delta(U) = \min_{A:V \to \Sigma} \delta(A(U))$$

Let $\gamma(U)$ be the spectral gap of G, namely, $\gamma(G) = 1 - \lambda(G)$ where $\lambda(G)$ is the second-largest eigenvalue of the adjacency matrix M_G of G.

Definition 2.7 (UNIQUEGAME)**.** *The* UNIQUEGAME *problem* (UG) *is a special case of the* CONSTRAINTGRAPH *problem, where each constraint is a permutation on* Σ*, i.e, for every* $e \in E$ *there exists a permutation* $\pi_e : \Sigma \to \Sigma$ *such that* $c_e = \{(\sigma, \pi_e(\sigma))\}_{\sigma \in \Sigma}$

Namely, every assignment to a vertex u *uniquely determines the assignment to its neighbor* v *in order for the constraint* $c_{(u,v)}$ *to be satisfied.*

We denote by UG_Σ the UNIQUEGAME problem, over the specific alphabet Σ. For any integer $i \geq 2$, we denote by UG_i the problem UG_Σ over some Σ such that $|\Sigma| = i$.

Definition 2.8 (MAXCUT **and** MAX-q-CUT)**.** *The* MAXCUT *problem is:*
Input: an undirected graph $G = (V, E)$*.*
Objective: find a partition of V *into* $(S, V \setminus S)$ *so that* $|E(S, V \setminus S)|$ *is maximized.*

The MINUNCUT *problem is:*
Input: an undirected graph $G = (V, E)$*.*
Objective: find a partition of V *into* $(S, V \setminus S)$ *so that* $1 - |E(S, V \setminus S)|$ *is minimized.*

The MAX-q-CUT *problem is:*
Input: an undirected graph $G = (V, E)$*.*
Objective: find a partition of V *into* q *subsets* $S_1, ..., S_q$ *so that*

$$\sum_{1 \leq i < j \leq q} |E(S_i, S_j)|$$

is maximized.

2.2.3 Parallel Repetition.

Definition 2.9 (Parallel-Repetition). *Let* $U = \langle G, \Sigma, C \rangle$ *be an instance of* CONSTRAINT-GRAPH *problem. The* k*th Parallel-Repetition of* U*, denoted by* $U^{\otimes k}$ *is an instance* $\langle G^{\otimes k}, \Sigma^k, C^{\wedge k} \rangle$ *of* CONSTRAINTGRAPH *where:*

$G^{\otimes k} = (V^k, E')$*, i.e, a vertex of* $G^{\otimes k}$ *is a* k*-tuple of vertices of* G*.*

An edge $e' \in E'$ *connects two vertices* $\langle v_{i_1}, ..., v_{i_k} \rangle, \langle v_{j_1}, ..., v_{j_k} \rangle \in V^k$ *if and only if* $(v_{i_1}, v_{j_1}), ..., (v_{i_k}, v_{j_k})$ *are all edges of* G*.*

The constraint $c_{e'} : \Sigma^k \to \Sigma^k$ *of an edge* $e' \in E^k$ *where* $e' = (\langle v_{i_1}, ..., v_{i_k}\rangle, \langle v_{j_1}, ..., v_{j_k}\rangle)$ *is satisfied by an assignment* $A : V^k \to \Sigma^k$ *if and only if the constraint of the* k *corresponding edges of* G *are satisfied by* A *restricted to the vertices of* G*. Namely,* $C^{\wedge k} = \{c_{e'}\}_{e' \in E'}$ *where*

$$c_{e'}(A(\langle v_{i_1}, ..., v_{i_k}\rangle), A(\langle v_{j_1}, ..., v_{j_k}\rangle)) = \bigwedge_{l \in [k]} c_{(v_{i_l}, v_{j_l})}(A(\langle v_{i_1}, ..., v_{i_k}\rangle)_{|l}, A(\langle v_{j_1}, ..., v_{j_k}\rangle)_{|l})$$

Note that the adjacency matrix $M_{G^{\otimes k}}$ of $G^{\otimes k}$ is simply the kth tensor product of the adjacency matrix M_G of G. Observe that if U is a UniqueGame instance then so is $U^{\otimes k}$. Observe also that $\gamma(M_G) = \gamma(M_{G^{\otimes k}})$. This is true as the eigenvalues of $M_1 \otimes M_2$ are exactly

$$\{\lambda_1 \cdot \lambda_2 \mid \lambda_1 \text{ is an eigenvalue of } M_1 \text{ and } \lambda_2 \text{ is an eigenvalue of } M_2\}$$

and therefore the second-largest eigenvalue of $A_{G^{\otimes k}} = (M_G)^{\otimes k}$ is $\lambda(G) \cdot 1^{k-1} = \lambda(G)$[‡].

Theorem 2.10. *[Raz98] (The exact parameters are from [Hol07]). If* U *is an instance of* ConstraintGraph *problem with error probability* δ*, then for every* k*,* $U^{\otimes k}$ *has an error probability at least* $1 - \left(1 - \frac{\delta^3}{6000}\right)^{\frac{k}{2\lg|\Sigma|}}$

Sometimes it is convenient to consider the following generalization of Parallel-Repetition where we have k different games:

Definition 2.11 (Parallel Games). *Let* $U_i = \langle G_i = (V_i, E_i), \Sigma_i, C_i\rangle$ *be* Constraint-Graph *instances for* $i \in [k]$*. Assume all* G_i *have the same degree* d*. The parallel games of* $\{U_i\}_{i\in[k]}$*, denoted by* $\bigotimes_{i\in k} U_i$ *is an instance* $\langle G', \Sigma', C'\rangle$ *where,*

- $G' = \bigotimes_{i=1}^k G_i = (E', V')$*. of*

  $V' = V_1 \times ... \times V_k$*. We denote a vertex of* G' *by* k-vertex.

  An edge $e' \in E'$ *connects two vertices* $\langle v_{i_1}, ..., v_{i_k}\rangle, \langle v_{j_1}, ..., v_{j_k}\rangle \in V'$ *if and only if* $(v_{i_1}, v_{j_1}), ..., (v_{i_k}, v_{j_k})$ *are edges of* $G_1, ..., G_k$ *respectively. We denote an edge of* G' *by* k-edge.

- $\Sigma' = \Sigma_1 \times ... \times \Sigma_k$

[‡]Indeed MaxCut instance U for which $\gamma(U) = 1 - \lambda_2(U)$ reduces to UniqueGame instance $U^{\otimes k}$ of the same spectral gap. However, the other direction is not known: the reduction in [KKMO04] from UniqueGame to $1-\varepsilon$ satisfiable MaxCut yields an instance M with spectral gap $\gamma(M) = O(\varepsilon)$, as the ε noise long-code test has spectral gap of $O(\varepsilon)$.

- *The constraint* $c_{e'} \subseteq \Sigma^k \otimes \Sigma^k$ *of an edge* $e' \in E'$ *where* $e' = (\langle v_{i_1}, ..., v_{i_k}\rangle, \langle v_{j_1}, ..., v_{j_k}\rangle)$ *is satisfied by an assignment* $A : V' \rightarrow \Sigma'$ *if and only if each constraint of the* k *corresponding edges of* $G_1, ..., G_k$ *is satisfied by* A *restricted to the vertices of the relevant* G_i*. Namely,* $C^{\wedge k} = \{c_{e'}\}_{e' \in E'}$ *where*

$$c_{e'}(A(\langle v_{i_1}, ..., v_{i_k}\rangle), A(\langle v_{j_1}, ..., v_{j_k}\rangle)) = \bigwedge_{l \in [k]} c_{(v_{i_l}, v_{j_l})}(A(\langle v_{i_1}, ..., v_{i_k}\rangle)_{|l}, A(\langle v_{j_1}, ..., v_{j_k}\rangle)_{|l})$$

For a k-vertex $v' = \langle v_{i_1}, ..., v_{i_k}\rangle \in V'$ we denote by $v'[j]$ the j'th coordinate of v', namely v_{i_j}. We denote by $A(v')_{|l}$ the assignment $A(v')$ projected on $v'[l]$.

For $v \in V$, let $V'_{i\downarrow v} \subseteq V'$ be the set of all k-vertices in V' having v at their ith coordinate, namely,

$$V'_{i\downarrow v} = \{v' \in V' \mid v'[i] = v\}$$

For $v \in V$, $\sigma \in \Sigma$ and an assignment $A' : V' \rightarrow \Sigma'$ let $V'_{i\downarrow v=\sigma}$ be the subset of $V'_{i\downarrow v}$ which are assigned σ at their ith coordinate, namely,

$$V'_{i\downarrow v=\sigma} = \{v' \in V'_{i\downarrow v} \mid A'(v')_{|i=\sigma}\}$$

and for $(u, v) \in E$, let $E'_{i\downarrow(u,v)} \subseteq E'$ be

$$E'_{i\downarrow(u,v)} = \{(x', y') \in E' \mid x'[i] = u, y'[i] = v\}$$

Let A' be an assignment to $\bigotimes_{i \in k} U_i$. Then we denote by A_i^{Plur} the assignment to U_i that is the most plural assignment at coordinate i. Namely,

$$A_i^{Plur}(v) = \operatorname*{argmax}_{\sigma \in \Sigma} |\{v' \in V'_{i\downarrow v} \mid A(v')_{|i} = \sigma\}|$$

2.3 Variants of Parallel Repetition

In this section we prove the main theorem regarding Noisy-Parallel-Repetition:

Theorem 2.2 (Main theorem) *Let U be a $1-\varepsilon$ satisfiable* ConstraintGraph *instance. Let $\mathbf{U}$ be the kth Noisy-Parallel-Repetition of U, with $P_{loop} = \frac{1}{2}$, where $k \leq \frac{1}{\varepsilon}$. Then $\mathbf{U}$ is at most $(1-\varepsilon)^{c_\gamma k}$ satisfiable where $\gamma = \gamma(U)$ and $c_\gamma = \frac{\gamma}{12800 \cdot \lg \frac{800}{\gamma}}$. Moreover, if G is a union of expanders, each with eigenvalue-gap at least γ then the same theorem holds (with c_γ multiplied by some constant).*

2.3.1 Noisy-Parallel-Repetition for Expanders

Proof of the main theorem (Theorem 2.2). We prove the upper-bound for the case of expanders first (Lemma 2.12). The upper-bound for union of disjoint expanders is proved in Corollary 2.22. From here on, when analyzing Noisy-Parallel-Repetition, we assume the self-loops and consider only the Parallel-Repetition operator.

Lemma 2.12 (Parallel-Repetition - Constructive upper-bound). *Let $U = \langle G, \Sigma, C\rangle$ be a* ConstraintGraph *instance wih spectral-gap $\gamma = \gamma(U)$ and set $\mathbf{U} = U^{\otimes k}$.*

Let A be an assignment that satisfies $1-\delta$ fraction of $\mathbf{U}$ and let A_i^{Plur} be the most plural assignment of A at coordinate i. For every $i \in [k]$ denote by δ_i the unsatisfied fraction of G by A_i^{Plur}.

Then $1-\delta \leq 1 - c_\gamma \cdot \min(1, \sum_{i\in[k]} \delta_i)$, where $c_\gamma = \frac{\gamma}{12800 \cdot \lg \frac{800}{\gamma}}$. Stated otherwise, if $\delta < c_\gamma$ then there exists $i \in [k]$ so that A_i^{Plur} satisfies at least $1 - \frac{\delta}{kc_\gamma}$ fraction of U.

Lemma 2.13 (Parallel-Games - Constructive upper-bound). *Let $U_i = \langle G_i, \Sigma_i, C_i\rangle$ be* ConstraintGraph *instances for $i \in [k]$. Set $\mathbf{U} = \bigotimes_{i\in k} U_i = \langle G', \Sigma', C'\rangle$ and let $\gamma = \min\{\gamma(U_i)\}_{i\in[k]}$.*

Let A' be an assignment that satisfies $1-\delta$ fraction of $\mathbf{U}$ and let A_i^{Plur} be the most plural assignment of A at coordinate i. For every $i \in [k]$ denote by δ_i the unsatisfied fraction of G_i by A_i^{Plur}.

Then $1-\delta \leq 1 - c_\gamma \cdot \min(1, \sum_{i\in[k]} \delta_i)$, where $c_\gamma = \frac{\gamma}{12800 \cdot \lg \frac{800}{\gamma}}$.

Stated otherwise, if $\delta < c_\gamma$ then there exists $i \in [k]$ so that A_i^{Plur} satisfies at least $1 - \frac{\delta}{c_\gamma k}$ fraction of U_i.

Note that Lemma 2.12 is an immediate corollary of Lemma 2.13, by taking $U_i = U$ for every $i \in [k]$. The constructiveness follows, as one can easily compute A_i^{Plur} for every $i \in [k]$ given A, and check the satisfied fraction by each of those assignments. See Section 2.4.2 for further discussion on the efficiency of the constructiveness.

Corollary 2.14 (Parallel-Repetition - upper-bound). *If $U = \langle G, \Sigma, C\rangle$ is a* ConstraintGraph *instance and $\mathbf{U} = U^{\otimes k}$, then the maximal satisfiability of $\mathbf{U}$ is at most $1 - c_\gamma \cdot \min(1, \delta(U) \cdot k)$, where $\gamma = \gamma(U)$.*

Corollary 2.15 (Parallel-Games - upper-bound). *Let $U_i = \langle G_i = (V_i, E_i), \Sigma_i, C_i\rangle$ be* ConstraintGraph *instances for $i \in [k]$. Set $\mathbf{U} = \bigotimes_{i\in k} U_i = \langle G', \Sigma', C'\rangle$ and let $\gamma = \min\{\gamma(U_i)\}_{i\in[k]}$.*

Then $\mathbf{U}$ is at most $1 - c_\gamma \cdot \min(1, \sum_{i\in[k]} \delta(U_i))$ satisfiable.

Proof of Lemma 2.13. Let A be an assignment to $\mathbf{U}$. Recall that A_i^{Plur} is the assignment to U_i that is the most plural assignment at coordinate i. We show that the sum of unsatisfied fraction of U_i by A_i (over all $i \in [k]$) is, up to sum multiplicative factor, a lower-bound on the unsatisfied fraction of $\mathbf{U}$ by A.

We do this by considering two cases. Either A is very consistent with the plurality assignments, and the upper-bound comes up naturally, or A is far from being consistent with the plurality assignments, and the inconsistency incurs many unsatisfied k-edges.

We say that an edge of G_i is *red* if it is not satisfied by A_i^{Plur}. Recall that δ_i is the fraction of red edges of G_i.

We say that a vertex $v \in V_i$ is *red* if it has many occurrences that disagree with A_i^{Plur}, that is, if $\Pr_{v'\in V'_{i\downarrow v}}\left[A(v')_{|i} = A_i^{Plur}(v)\right] \le \frac{99}{100}$. Denote by α_i the fraction of red vertices of G_i.

Let $S \subseteq [k]$ be the set of coordinates i for which at least $\frac{\delta_i}{4}$ of the vertices are red. Clearly $\sum_{i\in[k]} \delta_i = \sum_{i\in S} \delta_i + \sum_{i\in\overline{S}} \delta_i$. Therefore, (at least) one of the two sums on the right hand side is at least half of the left hand side.

We therefore only need to consider two cases. Either

1. $\sum_{i\in S} \delta_i \ge \frac{1}{2}\sum_{i\in[k]} \delta_i$, (in which case we show a lot of inconsistency in the projected assignment to U_i, namely many unsatisfied k-edges due to self-loops of G) or

2. $\sum_{i\in\overline{S}} \delta_i \ge \frac{1}{2}\sum_{i\in[k]} \delta_i$ (in which case we show many unsatisfied k-edges due to red edges).

We show that indeed in both cases there are many unsatisfied k-edges. We start with (1):

Claim 2.16. *In case (1), A satisfies at most $1 - c_\gamma \cdot \min(1, \sum_{i\in[k]} \delta_i)$ fraction of $\mathbf{U}$, where $c_\gamma = \frac{\gamma}{12800\cdot \lg \frac{800}{\gamma}}$.*

Proof. We utilize in this proof the γ expansion of the graph, as well as the self-loops of weight at least $\frac{1}{2}$.

Let $i \in S$ and let $v \in V_i$ be a red vertex. A k-edge $e' = (x', y') \in E'_{i\downarrow(v,v)}$ is unsatisfied if $A(x')_{|i} \neq A(y')_{|i}$ (as $x'[i] = y'[i] = v$ and the constraint on the self-loop of v demands equality).

Let

$$P_{i,v} = \Pr_{e'=(x',y')\in E'_{i\downarrow(v,v)}} [e' \text{ is not satisfied by } A]$$

We next give a lower-bound on the probability $P_{i,v}$. We then find a lower-bound on the probability P_1 that a random k-edge (of G') is unsatisfied, by considering the number of red vertices, and taking into account that a k-edge may be unsatisfied due to more than one red vertex, so we do not over-count.

Claim 2.17.

$$P_{i,v} \geq \frac{\gamma}{400}$$

Proof. Let $G'_{i\downarrow v} = (V'_{i\downarrow v}, E'_{i\downarrow(v,v)})$, namely $G'_{i\downarrow v}$ is a subgraph of G' where we consider only k-edges that correspond to the self-loops on v in the ith coordinate. As G_i is d-regular with (at least) $\frac{d}{2}$ self-loops on each vertex, we have that $G'_{i\downarrow v}$ is D-regular for $D \geq \frac{d^k}{2}$.

Consider the spectral-gap of $G'_{i\downarrow v}$. $G'_{i\downarrow v}$ is the graph (tensor) product of the following two graphs: (1) a single vertex v and self-loops on it and (2) $\bigotimes_{j\in[k]\setminus\{i\}} G_j$. Therefore, we have that $\gamma(G'_{i\downarrow v}) = \min\{\gamma(G_j)\}_{j\in[k]\setminus\{i\}} \geq \min\{\gamma(G_j)\}_{j\in[k]} = \gamma$.

By the Expander theorem (Theorem 1.1) we know that for every $M \subseteq V'_{i\downarrow v}$ of fractional size at most $\frac{1}{2}$,

$$\frac{|E_{G_{i\downarrow v}}(M, \overline{M})|}{D \cdot |M|} \geq \frac{\gamma}{2}$$

Claim 2.18. *There exists a cut $(M, \overline{M})$ of $V'_{i\downarrow v}$ so that $\frac{1}{100} \leq \frac{|M|}{|V'_{i\downarrow v}|} \leq \frac{1}{2}$ and every k-edge connecting M to $\overline{M}$ is unsatisfied.*

Proof. Consider the partition of $V'_{i\downarrow v}$ according to the values of the assignment A projected to v: $V'_{i\downarrow v=a_1}, ..., V'_{i\downarrow v=a_{|\Sigma|}}$. Every k-edge $e' \in E(V'_{i\downarrow v=a}, V'_{i\downarrow v=b})$, where $a \neq b$, is unsatisfied.

Let $\sigma = A_i^{Plur}(v)$. As v is a red vertex, $\frac{|V'_{i\downarrow v=\sigma}|}{|V'_{i\downarrow v}|} \leq \frac{99}{100}$. If $\frac{1}{100} \leq \frac{|V'_{i\downarrow v=\sigma}|}{|V'_{i\downarrow v}|}$ then we are done - by setting M to be the smaller of the two sets $V'_{i\downarrow v=\sigma}$ and $V'_{i\downarrow v} \setminus V'_{i\downarrow v=\sigma}$.

Otherwise, as $V'_{i\downarrow v=\sigma}$ is the largest of the $|\Sigma|$ parts of $V'_{i\downarrow v}$, for every $a \neq \sigma$ the part $V'_{i\downarrow v=a}$ is of fractional size at most $\frac{1}{100}$. We set M to be a union of a few of these parts so that its fractional size is at least $\frac{1}{100}$ and at most $\frac{2}{100}$. ■

Therefore, by taking M as guaranteed in the above claim, a random k-edge $e' \in E_{G'}(V'_{i\downarrow v})$ is unsatisfied with probability

$$P_{i,v} \geq \frac{|E'_{i\downarrow(v,v)}|}{|E_{G'}(V'_{i\downarrow v})|} \cdot \frac{|M|}{|V'_{i\downarrow v}|} \cdot \frac{\gamma}{2} \geq \frac{1}{2} \cdot \frac{1}{100} \cdot \frac{\gamma}{2} = \frac{\gamma}{400}$$

where $\frac{|E'_{i\downarrow(v,v)}|}{|E_{G'}(V'_{i\downarrow v})|} \geq \frac{D \cdot |V'_{i\downarrow v}|}{d^k \cdot |V'_{i\downarrow v}|} \geq \frac{1}{2}$.

■

This already yields a lower-bound on the total number of pairs of k-edges e' and vertex v where e' is unsatisfied due to the red vertex v (by multiplying $\frac{\gamma}{400}$ with number of k-edges that contain a self-loop of a red vertices). However, this may be an over-counting for the number of unsatisfied k-edges, as a k-edge may be unsatisfied due to more than one red vertex. Let us consider a more accurate estimate.

We first give an upper-bound on the probability that a random k-edge (x', y') has more than r red vertices in coordinates S' of $\mathbf{U}$ (where the parameter r is set later). For this purpose it is more convenient to consider, from here on, only coordinates that do not have too many red vertices.

Recall that α_i is the fraction of red vertices in coordinate i (recall that $\forall i \in S, \frac{\delta_i}{4} \leq \alpha_i \leq \frac{1}{2}$). If $\sum_{i \in S} \alpha_i \leq 1$ then let $S' = S$. Otherwise set $S' \subseteq S$ to be such that $\frac{1}{2} \leq \sum_{i \in S'} \alpha_i \leq 1$.

Let R be a random variable corresponding to the number of red vertices of coordinates S' in a random k-vertex. $\mathbb{E}R \leq \sum_{i \in S'} \alpha_i \leq 1$. Therefore (by Chernoff's [Che52] bound, $\Pr[R > (1+\alpha)\mathbb{E}R] < \left(\frac{e^\alpha}{(1+\alpha)^{(1+\alpha)}}\right)^{\mathbb{E}R}$)

$$\Pr[R > r] < \left(\frac{e^{r-1}}{r^r}\right) < \left(\frac{e}{r}\right)^r$$

Consider a random k-edge $e' = (x', y') \in E'$. The probability that it is unsatisfied due to the red vertex v and that the number of red vertices is at most r (by choosing $r = \lg \frac{800}{\gamma}$) at least

$$P_{i,v} - \left(\frac{e}{r}\right)^r > \frac{\gamma}{400} - \frac{\gamma}{800} = \frac{\gamma}{800}$$

Therefore, the probability P_1 that a random k-edge $e' \in E'$ is unsatisfied is

$$P_1 \geq \frac{1}{2r} \sum_{i \in S'} \alpha_i \cdot \frac{\gamma}{800} \tag{2.1}$$

where the $2r$ in the denominator is due to the fact that we count every k-edge at most $2r$ times in the summation (at most r times for each of its two k-vertices). Therefore,

$$P_1 \geq \frac{\gamma}{1600 \cdot \lg \frac{800}{\gamma}} \cdot \sum_{i \in S'} \alpha_i \tag{2.2}$$

as $\sum_{i \in S'} \alpha_i \geq \min(\frac{1}{2}, \sum_{i \in S} \frac{\delta_i}{4}) \geq \min(\frac{1}{2}, \frac{1}{8} \cdot \sum_{i \in [k]} \delta_i)$, we have

$$P_1 \geq \frac{\gamma}{1600 \cdot \lg \frac{800}{\gamma}} \cdot \min(\frac{1}{2}, \frac{1}{8} \cdot \sum_{i \in [k]} \delta_i) \geq c_\gamma \cdot \min(1, \sum_{i \in [k]} \delta_i) \tag{2.3}$$

as Claim 2.16 asserts, and we are done for case (1). ∎

Claim 2.19. *In case (2), A satisfies at most $1 - \frac{1}{10} \min(1, \sum_{i \in [k]} \delta_i)$ fraction of* $\mathbf{U}$.

Proof. We show that the red edges of coordinates $\overline{S}$, together with the high consistency, yield many unsatisfied k-edges.

Claim 2.20. *For any $i \in \overline{S}$ there are at least $\frac{\delta_i nd}{4}$ red edges at coordinate i that do not touch red vertices of coordinate i.*

Proof. There are $\delta_i \frac{nd}{2}$ edges that are red in coordinate i. There are at most $\alpha_i nd \leq \frac{\delta_i n}{4} d$ edges of G touching the red vertices of coordinate i. Therefore, there are at least $\frac{\delta_i nd}{4}$ red edges at coordinate i that do not touch red vertices of coordinate i. ∎

We pick a subset of such edges of size $\frac{\delta_i nd}{4}$ and call them orange edges.

Claim 2.21. *If an edge (u, v) is orange (at coordinate i) then at least $\frac{98}{100}$ of the set $E_{i\downarrow(u,v)}$ —the k-edges that contain (u, v) at coordinate i— are not satisfied.*

Proof. At most $\frac{1}{100}$ of them disagree with A_i^{Plur} on either sides, as both u and v are not red. Therefore, at least $\frac{98}{100}$ of $E_{i\downarrow(u,v)}$ agree with A_i^{Plur} on both sides and therefore are unsatisfied. ■

Let us now give an upper-bound on the probability that a random k-edge (x', y') has more than r orange edges in coordinates $\overline{S}$ of $\mathbf{U}$ (where the parameter r is set later). For this purpose it is more convenient to assume that there are not too many red edges in coordinates $\overline{S}$. If $\sum_{i\in\overline{S}} \delta_i \le 1$ then let $S' = \overline{S}$. Otherwise let $S' \subseteq \overline{S}$ be such that $\frac{1}{2} \le \sum_{i\in S'} \delta_i \le 1$.

Consider a random k-edge $e' \in E_{i\downarrow(u,v)}$. Similarly to the previous claim, the probability that e' is both unsatisfied due to (u, v) at coordinate i and that e' contains at most r orange edges in coordinates S' is at least

$$\frac{98}{100} - \left(\frac{e}{r}\right)^r$$

Therefore, the probability that a random k-edge is unsatisfied is

$$P_2 \ge \sum_{i\in S'} \frac{\frac{\delta_i}{4} \cdot \left(\frac{98}{100} - \left(\frac{e}{r}\right)^r\right)}{r}$$

where the r in the denominator is from the fact that we count every k-edge at most r times in the summation.

Taking r so that $\frac{98}{100} \ge 2 \cdot \left(\frac{e}{r}\right)^r$ (e.g, $r = \lg \frac{200}{98}$) , and as $\sum_{i\in S'} \delta_i \ge \min(\frac{1}{2}, \frac{1}{2}\sum_{i\in[k]} \delta_i)$ we obtain

$$P_2 \ge \frac{98}{400} \cdot \frac{1}{\lg \frac{200}{98}} \min(\frac{1}{2}, \sum_{i\in[k]} \delta_i) \ge \frac{1}{10} \cdot \min(1, \sum_{i\in[k]} \delta_i)$$

as the claim asserts. ■

By Claim 2.16 (case (1)) and Claim 2.19 (case (2)), A satisfies at most $1 - c_\gamma \cdot \min(1, \sum_{i\in[k]} \delta_i)$ fraction of $\mathbf{U}$. Therefore, Lemma 2.13 follows. ■

2.3.2 Noisy-Parallel-Repetition for Union of Disjoint Expanders.

We now prove the upper-bound of the main theorem for the case of union of disjoint expanders.

Corollary 2.22. *Let* $U = \langle G, \Sigma, C\rangle$ *be a* ConstraintGraph *instance. Assume that* G *is a union of* t *disjoint expanders, each with eigenvalue gap at least* γ*, namely* $U = \bigcup_{i\in[t]}\{U_i\}$ *where* $U_i = \langle G_i, \Sigma_i, C_i\rangle$.

Set $U^{\otimes k} = \langle G^{\otimes k}, \Sigma^k, C^{\wedge k}\rangle$.

Then $U^{\otimes k}$ *maximal satisfiability is at most* $1 - c'_\gamma \cdot \min(1, k\cdot\delta(U))$*, where* $c'_\gamma = \frac{c_\gamma}{10^4}$.

If the upper-bound for expanders instances had worked for any (arbitrarily small) constant of success probability of the output instance, then this corollary would have been immediate, by a weighted averaging of the success probability of the connected components of $U^{\otimes k}$.

However, this is not the case. Therefore, the guarantee on the error probability of some of the connected components of $U^{\times k}$ is smaller than what it could have been otherwise.

Proof. Lemma 2.12 cannot be used here, as G may be unconnected, therefore $\gamma(\mathbf{U})$ may be zero. Note however, that any connected component G' of $G^{\otimes k}$, has $\gamma(G') \geq \gamma$. This is true as every connected component G' of $G^{\otimes k}$ is a tensor product of k connected components of G, each of them having eigenvalue gap $\gamma(G') \geq \gamma(G)$. Therefore, we can use Lemma 2.13 instead of Lemma 2.12.

Let $p_i = \frac{|G_i|}{|G|}$ be the probability that a random vertex (or a random edge) falls inside the component G_i. We have $\sum_{i\in[t]} p_i\delta(U_i) = \delta(U)$ and $\sum_{i\in[t]} p_i = 1$.

For $s = (s_1, ..., s_k) \in [t]^k$ denote by G_s the connected component $G_{s_1} \otimes ... \otimes G_{s_k}$. Let $p_s = \prod_{i\in s} p_i$ be its fractional size, and let $\delta_s = \sum_{i\in s}\delta_i$.

The probability P of a random k-edge of $G^{\otimes k}$ to be unsatisfied is the probability of a random k-edge to be in a certain connected component G_s times the probability that a random k-edge of G_s is unsatisfied, summing over all connected components $\{G_s\}_{s\in[t]^k}$. Namely,

$$P = \sum_{s\in[t]^k} p_s \cdot \Pr_{e'\in E(G_s)}[e' \text{ is unsatisfied by A}]$$

By Lemma 2.13, as each connected component G_s has $\gamma(G_s) \geq \gamma$,

$$P \geq c_\gamma \cdot \sum_{S\in[t]^k} p_s \cdot \min(1, \delta_s) \tag{2.4}$$

As $\mathbb{E}_s\delta_s = k\mathbb{E}_i\delta_i = k\sum_{i\in[t]} p_i\delta_i = k\delta(U)$, by Chernoff's bound we have that

$$\Pr\left[\delta_s \in R\right] \geq \frac{45}{100}$$

where $R = \left[\frac{1}{100} \cdot k\delta(U), 10 \cdot k\delta(U)\right]$

Therefore,

$$\mathbb{E}\left[\delta_s \mid \delta_s \in R\right] \geq \frac{k\delta(U)}{100} \cdot \frac{45}{100}$$

For $k \leq \frac{1}{\delta(U)}$, and for every $\delta_s \in R$, the $\min(1,\delta_s)$ expression in equation 2.4 reduces the contribution of δ_s to the sum by a factor of at most 10. Therefore,

$$\begin{aligned}
P &\geq c_\gamma \cdot \sum_{S\in[t]^k \ \mid \ \delta_s\in R} p_s \cdot \min\left(1,\delta_s\right) \\
&\geq \frac{1}{10} \cdot c_\gamma \cdot \sum_{S\in[t]^k \ \mid \ \delta_s\in R} p_s \cdot \delta_s \\
&= \frac{1}{10} \cdot c_\gamma \cdot \mathbb{E}\left[\delta_s \mid \delta_s \in R\right] \\
&\geq 10^{-4} \cdot c_\gamma \cdot k\delta(U)
\end{aligned}$$

For $k > \frac{1}{\delta(U)}$, the satisfiability of $U^{\otimes k}$ is at most the satisfiability of $U^{\otimes \frac{1}{\delta(U)}}$. We therefore have that for every k, $U^{\otimes k}$ maximal satisfiability is at most $1 - c'_\gamma \cdot \min(1, k \cdot \delta(U))$, where $c'_\gamma = \frac{c_\gamma}{10^4}$.

■

We therefore have the main theorem for the case of union of disjoint expanders as well.

■

2.3.3 Expanding-Parallel-Repetition for Non-Expanders Instances

We next consider the Expanding-Parallel-Repetition variation that yields an "optimal" amplification for every instance, up to some constant factor of success probability of the generated instance, even for instances that may lack good expansion properties. This variation, however, does not preserve the uniqueness property of the constraint. The proof is simple, given the analysis for the Noisy-Parallel-Repetition.

Corollary 2.4(Expanding-Parallel-Repetition) *Let U be a $1-\varepsilon$ satisfiable* ConstraintGraph *instance. Let $\mathbf{U}$ be the kth Expanding-Parallel-Repetition of U, with $P_{loop} = \frac{1}{2}$ and $P_H = \frac{1}{4}$, where $k \leq \frac{1}{\varepsilon}$.*

Then $\mathbf{U}$ is at most $(1-\varepsilon)^{ck}$ satisfiable, where c is some universal constant (that does not depend on the spectral-gap or the alphabet size of the input instance).

We show that by choosing H so that $\gamma(H)$ is large (say, a complete graph, or a constant degree expander) and setting P_H and P_{loop} to be constants this variant of Parallel-Repetition is "optimal". Note that by choosing H to be a constant degree expander (rather than a complete-graph) $\mathbf{U}$ is of constant degree (provided that U is).

The idea to use such "expanderizing" technique already appears as a first step in Dinur's new proof of the PCP [Din07], though it is critical there that the added expander is of constant degree.

Proof. Let $U = \langle G = (V, E), \Sigma, C\rangle$ be an instance of *Gap-ConstraintGraph-* $[1-\delta, 1-\varepsilon]$. Let $U_H = \langle H, \Sigma, C_H\rangle$ where $\gamma(H) = \gamma$ and the constraints C_H are always satisfied. Let $U_{loops} = \langle (V, E_{loops}), \Sigma, C_{loops}\rangle$ where E_{loops} are self-loop for every vertex, and the constraints C_{loops} are the equality constraints. Set U' to be an instance constructed by combining the instances U_H, U_{loops} and U, with relative weights P_H, P_{loop} and $1 - P_H - P_{loop}$ respectively.

We thus get an instance U' of *Gap-ConstraintGraph-* $\left[1-\frac{\delta}{4}, 1-\frac{\varepsilon}{4}\right]$ with $\gamma(U') = \frac{\gamma}{4}$. Applying now the kth Parallel-Repetition, for $k \leq \frac{4}{\delta}$, yields, by the main theorem, an instance $\mathbf{U} = (U')^{\otimes k}$ of *Gap-ConstraintGraph-* $\left[1 - \frac{c_\gamma}{4} \cdot \frac{k\delta}{4}, 1 - \frac{k\varepsilon}{4}\right]$.

■

The constructiveness (Corollary 2.5) follows, as any good assignment to $(U')^{\otimes k}$ can be efficiently converted (by the main theorem) into a good assignment to U'. This assignment is a good assignment to U as well, up to a $\frac{1}{1-P_{loop}-P_H}$ multiplicative factor on the error probability.

As the analysis of the Expanding-Parallel-Repetition is an immediate result of the analysis of the Noisy-Parallel-Repetition, any improvement of the latter (say, to hold for any k, not only for $k \leq \frac{1}{\delta}$) immediately translates to the same improvement for the Expanding-Parallel-Repetition for arbitrary ConstraintGraph instances (again, without preserving uniqueness property).

2.3.3.1 The Miss-Match Form

Feige and Kilian [FK00] introduce and discuss the miss-match form of the two-prover one round proof system, and analyze the application of Parallel-Repetition to it. In the miss-match form of proof-system, the verifier asks the first prover two questions, and either applies a test only to the two answers of this prover, or asks the second prover two questions —one of which is identical to one of the two questions already asked— and also tests that the identical questions are answered with identical answers. The decision whether to test the answers of the first prover only, or to test the consistency of the two provers as well, may rely on the answer of the first prover.

Feige and Kilian show that in order to reduce the success probability of miss-match form from $1 - \varepsilon$ to $1 - \delta$, one can apply the kth Parallel-Repetition with $k = \text{Poly}(\frac{1}{\varepsilon}, \frac{1}{\delta})$. Their analysis works for any constants $0 < \varepsilon < \delta < 1$, and is independent of the alphabet size.

They also show that any two-prover one round proof system can be reduced to the miss-match form, such that the success probability changes by a constant factor at most.

Comparing their amplification to our Expanding-Parallel-Repetition in the context of ConstraintGraph, both can be applied to arbitrary ConstraintGraph and both amplification rates are independent of the alphabet size. Our amplification exceeds theirs in the sense that in order to amplify $1 - \varepsilon$ error-probability to $1 - \delta$, one has to apply the kth Parallel-Repetition with k linear in $\frac{\varepsilon}{\delta}$, while theirs is polynomial in this factor. However, our analysis works only for δ that is lower-bounded by some universal constant, while theirs works for any (arbitrarily small) constant δ.

2.4 Algorithmic Applications

2.4.1 Approximating Algorithms via Optimal Parallel-Repetition

Let us now consider a simple algorithmic application of "optimal" analysis for Parallel-Repetition and the implication of constructiveness in such an analysis. The idea is to use the Parallel-Repetition to convert an algorithm for $1-\varepsilon_0$ satisfiable instances of $UniqueGame_{\Sigma_0}$ (for some specific $\varepsilon_0 > 0$ and Σ_0) into an algorithm that performs well on $1-\varepsilon$ satisfiable instances of $UniqueGame_{\Sigma}$ for any $\varepsilon \leq \varepsilon_0$ and corresponding Σ. Let us consider some applications for this technique.

By [GW95, CMM06]) on the one hand, and by [KKMO04] on the other hand we have a tight threshold of approximability for almost completely satisfiable instances of UniqueGame (and MaxCut). That is, for every $\varepsilon \leq \frac{1}{\lg|\Sigma|}$, given a $1-\varepsilon$ satisfiable instance $U = \langle G, \Sigma, C\rangle$ of $UniqueGame_{\Sigma}$, finding an assignment that satisfies $1 - c\sqrt{\varepsilon \lg|\Sigma|}$ of its constraints is in P for some constant c, but it is UG-hard for any factor slightly smaller than c .

Consider UniqueGame instances where the underlying graph is an expander. Observe that this is a special case of UniqueGame, and the optimal tradeoffs between error, alphabet-size and approximation factor may be better than in the general case. In particular one may come up with an approximation that does not depend on the alphabet size, but only on the spectral-gap and the satisfiability of the instance.

Recently, Arora *et al.*[AKK$^+$08], suggested techniques that may indeed achieve that goal. Their algorithm seems to find an assignment that satisfies a constant fraction of the constraints for any $1-\varepsilon$ satisfiable UniqueGame instance U, as long as $\varepsilon \leq \gamma^2$.

This exciting result implies that UniqueGame for expander graphs is seemingly easier. Which in turn implies that a general reduction from UniqueGame to UniqueGame that is a good expander, would result in UniqueGame being in P, thus refuting the UniqueGame-conjecture.

We next show how the main theorem (Theorem 2.2) can be utilized to amplify an algorithm A that finds an assignment satisfying a (large enough) constant fraction of the constraints into an algorithm A' that obtains an assignment that satisfies all but aa small fraction.

Let A be an algorithm that guarantees a solution satisfying at least $1-c_\gamma$ for $1-\gamma^2$

satisfiable instances. Then for every $1-\varepsilon$ satisfiable instance, we can find a $1-O(\frac{\varepsilon}{\gamma^2\cdot c_\gamma})$ satisfying solution, as long as $\varepsilon \le \gamma^2$ (where $c_\gamma = \frac{\gamma}{12800\cdot\lg\frac{800}{\gamma}}$).

2.4.2 A Note on the Efficiency of the Constructiveness.

As already mentioned, the constructiveness is obtained here simply by considering the k plurality assignments. The trivial algorithm for this runs in time $\mathrm{Poly}(|U^{\otimes k}|)$, similar to the running time of the (implicit [Raz07]) constructiveness of [Raz98, Hol07].

However, there is a simple Las-Vegas randomized algorithm that runs in expected time $O(k\cdot|\Sigma|\cdot|U|^2)$, assuming an oracle access to the assignment A for $U^{\otimes k}$ (which is probably not the case for the analysis of Raz and of Holenstein [Raz07]). This is done by sampling A (rather than reading it entirely) and computing the plural value with high probability for every coordinate $i\in[k]$. The resulting k assignments can then be checked to see whether any of them satisfy enough constraints of U.

Therefore, when considering expander instances, it may be preferable to use our constructiveness (rather than the one implicit in [Raz98, Hol07]) both because of time considerations and because of the approximation ratio guarantee.

2.5 Concluding Remarks and Open problems

We showed two variants of the Parallel-Repetition technique that perform "optimally", i.e, the success probability decays exponentially fast with k, regardless of the alphabet size, albeit only down to some constant.

The Noisy-Parallel-Repetition variant preserves the uniqueness property, but works only for good expanders (or union of disjoint expanders). The Expanding-Parallel-Repetition variant works well for any instance, but does not preserve the uniqueness property.

Considering the goal of proving MaxCut-hardness for UniqueGame, these two amplification techniques are not sufficient. To achieve this goal, we would like (1) to obtain both results simultaneously, i.e, to have no dependency on the spectral-gap while maintaining the uniqueness property; and (2) to generalize this so it holds for an arbitrarily small constant target success probability. As mentioned above, by the recent counterexample of Raz [Raz08] this is not true for the Parallel-Repetition (and its Noisy-Parallel-Repetition variant).

2.6 Appendix

2.6.1 The NP-hardness of Approximating MinUnCut.

We next consider the NP-hardness for ε-solvable MinUnCut. By Trevisan *et al.* and Håstad [TSSW00, Hås01] we know that for any arbitrarily small $\varepsilon > 0$,

$$\textit{Gap-MaxCut-}\left[\frac{16}{32}+\varepsilon, \frac{17}{32}+\varepsilon\right] \in NP - hard$$

therefore

$$\textit{Gap-MinUnCut-}\left[\frac{15}{32}-\varepsilon, \frac{16}{32}-\varepsilon\right] \in NP - hard$$

For every arbitrarily small $\varepsilon' > 0$, this can be reduced to

$$\textit{Gap-MinUnCut-}\left[\left(\frac{15}{32}-\varepsilon\right)\cdot\varepsilon', \left(\frac{16}{32}-\varepsilon\right)\cdot\varepsilon'\right]$$

Just add sufficient number of edges (on new vertices) that can always be satisfied. Therefore, for every (arbitrarily small) $\varepsilon'' > 0$, MinUnCut is NP-hard to approximate better than $\frac{16}{15}$, even for instances that are ε'' solvable.

Applying the exact same reduction to the NP-hardness of *Gap-E2Lin2-* $\left[\frac{11}{16}-\varepsilon, \frac{12}{16}-\varepsilon\right]$ [TSSW00, Hås01] gives NP-hardness for

$$\textit{Gap-UniqueGame-}\left[\left(\frac{4}{16}-\varepsilon\right)\cdot\varepsilon', \left(\frac{5}{16}-\varepsilon\right)\cdot\varepsilon'\right]$$

Therefore, for every (arbitrarily small) $\varepsilon'' > 0$, DualUniqueGame$_2$ (and DualUniqueGame in general) is NP-hard to approximate better than $\frac{5}{4}$, even for instances that are ε'' solvable.

2.6.2 Larger Spectral-Gap for MaxCut.

We next note that *Gap-MaxCut-* $[1-\delta, 1-\varepsilon]$ with arbitrary spectral-gap reduces to *Gap-MaxCut-* $[1-\delta(2-\varepsilon), 1-2\varepsilon]$ with spectral-gap $\gamma = \Omega(\varepsilon)$.

Let the input to the reduction be $U = \langle G, \{0,1\}, C\rangle$, a $1-\varepsilon$ solvable instances of MaxCut with arbitrary spectral gap $\gamma(U)$.

Let $U_{exp} = \langle G_{exp}, \{0,1\}, C_2\rangle$ be an instance on the same set of vertices, where the constrains are "$\neq$" and G_{exp} is a constant degree expander, namely, $\gamma(U_{exp}) \geq \frac{1}{10}$.

Let $\mathbf{U}$ be the output of the reduction where $\mathbf{U}$ is the natural weighted combination of U and U_{exp} with weights $1-\varepsilon$ and ε respectively. Then $\gamma(\mathbf{U}) \geq \frac{\varepsilon}{10}$. If U is at least $1-\varepsilon$ solvable then $\mathbf{U}$ is at least $(1-\varepsilon)^2 > 1-2\varepsilon$ solvable. And if U is at most $1-\delta$ solvable, then $\mathbf{U}$ is at most $(1-\delta)(1-\varepsilon)+\varepsilon = 1-\delta(1-\varepsilon)$ solvable.

Chapter 3

An Elementary Construction of Constant-Degree Expanders

3.1 Introduction

All graphs considered here are finite, undirected and may contain self-loops and parallel edges. Expanders are graphs, which are simultaneously sparse, yet highly connected, in the sense that every cut contains (relatively) many edges. In this chapter we mostly work with the notion of *edge-expansion*. A d-regular graph $G = (V, E)$ is a δ-edge-expander (δ-expander for short) if for every set $S \subseteq V$ of size at most $\frac{1}{2}|V|$ there are at least $\delta d|S|$ edges connecting S and $\overline{S} = V \setminus S$, that is, $e(S, \overline{S}) \geq \delta d|S|$. For brevity, we say that a graph is an $[n, d, \delta]$-expander if it is an n-vertex d-regular δ-expander. Expanders are some of the most widely used objects in theoretical computer science, and have also found many applications in other areas of computer-science and mathematics. See the survey of Hoory et. al. [HLW06] for a discussion of several applications and references. Another widely used notion of expansion is based on algebraic properties of a matrix representation of the graph. Let $G = (V, E)$ be an n-vertex d-regular graph, and let A be the adjacency matrix of G, that is, the $n \times n$ matrix, with $A_{i,j}$ being the number of edges between i and j. It is easy to see that 1^n is an eigenvector of A with eigenvalue d, and that this is the only eigenvector with this eigenvalue iff G is connected. We denote by $\lambda_2(G)$ the second largest eigenvalue of A. It is easy to see that $\lambda_2(G) = \max_{0 \neq x \perp 1^n} \langle Ax, x\rangle / \langle x, x\rangle$. The following is a well known relation between the expansion of G and $\lambda_2(G)$.

Theorem 3.1 ([Alo86], [AM85] ,[Dod84]). *Let G be a δ-expander with adjacency matrix A, let $\lambda_2 = \lambda_2(G)$ be the second largest eigenvalue of A, and let its (normalized) spectral gap be $\gamma = 1 - \frac{\lambda_2}{d}$. Then,*

$$\frac{\gamma}{2} \leq \delta \leq \sqrt{2\gamma}$$

Our construction uses only the left-hand simple inequality.

The most useful expanders are those with constant degree. A priori, it is not clear that constant-degree expanders even exist. Pinsker [Pin73] established their existence.

Theorem 3.2 ([Pin73]). *There exists a fixed $\delta > 0$, such that for any $d \geq 3$ and even integer n, there is an $[n, d, \delta]$-expander, which is d-edge-colorable* *.

One way to prove the above is to take a random d-regular bipartite graph. In most applications, however, one needs to efficiently construct constant degree expanders explicitly. There are two notions of constructibility of d-regular expanders. The first (weaker) notion requires the n-vertex graph to be constructible in polynomial time in its size. The second (stronger) notion requires that given a vertex v and $i \in [d]$ it would be possible to generate the i^{th} neighbor of v in time $\text{Poly}(\log n)$. Such an expander is said to be *fully explicit*. In applications, where one needs to use the entire graph, it is often enough to use the weaker notion. However, in such cases (e.g. in certain reductions) one frequently needs to be able to construct a graph of a given size n. It has been observed by many that to this end it is enough to be able to construct graphs of size $\Theta(n)$ (e.g., one can take a cn-vertex expander and join groups of c vertices to get an n-vertex expander with positive expansion). In other cases, where one needs only part of the expander (e.g., when performing a random walk on a large expander) one usually needs the stronger notion of fully explicitness. However, in these cases it is usually enough to be able to construct an expander of size $\text{Poly}(n)$, as what we are interested in is actually the logarithm of the size of the graph.

Margulis [Mar73] and Gabber and Galil [GG81] were the first to efficiently construct constant degree expanders. Following was a sequence of works that culminated in the construction of Lubotzky, Phillips and Sarnak [LPS88] and Margulis [Mar88] of Ramanujan Graphs. These constructions rely (directly or indirectly) on estimations of the second largest eigenvalue of the graphs, and some of them, rely on deep mathematical results. A

*That is, one can assign its edges d colors such that edges incident with the same vertex are assigned distinct colors.

simpler, iterative construction was given by Reingold, Vadhan and Wigderson [RVW02]. This construction also relies on proving the expansion of the graphs by estimating their eigenvalues, and is the first construction of constant degree expanders with relatively elementary analysis. Additional algorithms for constructing bounded-degree expanders appear in [Ajt94] and [BLar], but these algorithms are not fully explicit in the sense described above.

Our construction is based on the replacement product of two graphs G and H, which is one of the most natural ways of combining two graphs. We start by defining this basic operation.

Definition 3.3. *Let G be a D-regular D-edge-colorable graph on n vertices and let H be a d-regular graph on D vertices. Suppose G is already equipped with a proper D-edge-colorings. The* replacement product *$G \circ H$ is the following $2d$-regular graph on nD vertices: We first replace every vertex v_i of G with a cluster of D vertices, which we denote $C_i = \{v_1^i, \ldots, v_D^i\}$. For every $1 \le i \le n$ we put a copy of H on C_i by connecting v_p^i to v_q^i if and only if $(p, q) \in E(H)$. Finally, for every $(p, q) \in E(G)$, which is colored t, we put d parallel edges between v_t^p and v_t^q.*

Note that if H is d-edge-colorable then $G \circ H$ is $2d$-edge colorable: simply color the copies of H within each set C_i using colors $1, \ldots, d$. As the edges between the sets C_i form d parallel copies of a perfect matching on the vertices of $G \circ H$, we can color any set of d parallel edges using the colors $d+1, \ldots, 2d$. Already in the 80's, Gromov [Gro83] has analyzed the effect of (a slight variant of) this operation on the spectral properties of graphs. Reingold, Vadhan and Wigderson [RVW02] considered the above variant, and showed, via a reduction to their algebraic analysis of the zig-zag product, that if two graphs are expanders then so is their product. Their argument is based on analyzing $\lambda(G \circ H)$ as a function of $\lambda(G)$ and $\lambda(H)$. We analyze the replacement product *directly* via an elementary combinatorial argument.

Theorem 3.4. *Suppose E_1 is an $[n, D, \delta_1]$-expander and E_2 is a $[D, d, \delta_2]$-expander. Then, $E_1 \circ E_2$ is an $[nD, 2d, \frac{1}{80}\delta_1^2\delta_2]$-expander.*

The proof of Theorem 3.4 is very simple; we show that $e(X, \overline{X})$ has either many edges within the clusters C_i or between them. Our main result is a new construction of constant-degree expanders. The main idea can be summarized as follows: a simple special case of one of the results of [AR94] gives a construction of $[n, O(\log^2 n), \frac{1}{4}]$-expanders. To get expanders with constant degree we construct such an $[n, O(\log^2 n), \frac{1}{4}]$-expander and then apply the replacement product with another similar expander in order to reduce the degree to $O(\sqrt{\log n})$ (in fact the degree could easily be further reduced, but this suffices to our purpose). We now find a constant degree expander of size $O(\sqrt{\log n})$, using exhaustive search, and apply a final replacement product to get a constant degree. Note that here we do not care much about the fact that the replacement product decreases the edge-expansion as we only apply it twice. A suitable choice of parameters gives the following construction, whose analysis relies solely on the easy part of Theorem 3.1, a simple special case of the result of [AR94] and on the elementary analysis of the replacement product (Theorem 3.4).

The following theorem states the explicit constructiveness:

Theorem 3.5 (Main Result). *There exists a fixed $\delta > 0$ such that for any integer $q = 2^t$ and for any $q^4/100 \leq r \leq q^4/2$ there is a polynomial time constructible $[q^{4r+12}, 12, \delta]$-expander.*

For completeness we prove all the necessary ingredients, thus obtaining a short and self-contained construction of constant-degree expanders. It is easy to see that given n, Theorem 3.5 can be used to construct an m-vertex expander with $n \leq m = O(n \log n)$.

3.1.1 Outline

The construction and its analysis appear in Section 3.2. In Section 3.3 we observe that simple variants of Theorem 3.5 give a construction with $\Theta(n)$ vertices and a construction which is fully explicit (albeit of size within some polynomially range). Section 3.3 contains some remarks regarding the construction (e.g, how to improve the edge expansion).

3.2 The Construction

Let us start by describing the special case of [AR94] that suffices for our purposes. For any $q = 2^t$ and $r \in \mathbb{N}$, we define a graph $LD(q, r)$ as follows. The vertices are all elements of $\mathbb{F}_q^{r+1}$, which can be thought of as all strings $(a_0, \ldots, a_r) \in \mathbb{F}_q^{r+1}$. A neighbor of a vertex a is indexed by an element $(x, y) \in \mathbb{F}_q^2$. In this notation neighbor (x, y) of vertex $a = (a_0, \ldots, a_r)$ is $a + y \cdot (1, x, x^2, \ldots, x^r)$. $LD(q, r)$ is clearly a q^2-regular graph on q^{r+1} vertices. It is also q^2-edge-colorable as we can color the edges indexed (x, y) using the "color" (x, y) (note that this is well defined as addition and subtraction are identical in $\mathbb{F}_{2^t}$). The following result is a special case of the result of [AR94]:

Theorem 3.6 ([AR94]). *For any $q = 2^t$ and integer $r < q$ we have $\lambda_2(LD(q, r)) \leq rq$.*

Note that the above theorem, together with the left inequality of Theorem 3.1, imply that if $r \leq q/2$ then $LD(q, r)$ is a $[q^{r+1}, q^2, \frac{1}{4}]$-expander. We first prove our main result based on Theorems 3.4, 3.6 and the left inequality of Theorem 3.1. We then prove these three results.

We next show a combinatorial analysis of the replacement product. The analysis here is not tight and could be improved. Indeed, we do not try to present the strongest possible bound, but rather to give one with a simple proof. Note that it suffices for our purpose, as we apply it only a constant number of times.

Proof of Theorem 3.5: Given integers q and $\frac{q^4}{100} \leq r \leq \frac{q^4}{2}$, we start by enumerating all 3-regular graphs on q^2 vertices until we find one which is a δ-expander and 3-edge colorable (one exists by Theorem 3.2). This step can clearly be carried out in time $q^{O(q^2)}$. Denote by E_1 the expander we find and define $E_3 = LD(q^4, r)$, $E_2 = LD(q, 5)$ and set $E_4 = E_3 \circ (E_2 \circ E_1)$ to be our final graph. As E_1, E_2 and E_3 are $[q^2, 3, \delta]$, $[q^6, q^2, \frac{1}{4}]$ and $[q^{4r+4}, q^8, \frac{1}{4}]$ expanders respectively, E_4 is a $[q^{4r+12}, 12, \delta']$-expander for some absolute constant δ' (here we rely on Theorem 3.4). Moreover, given E_1 one can easily compute E_4 in time polynomial † in the size of E_4. As $r \geq q^4/100$, E_4 is of size at least $q^{q^4/10}$, thus the first step of finding E_1 also takes time polynomial in the size of E_4, as needed. ■

†Note that when constructing E_2 and E_3 we need representations of $\mathbb{F}_q$ and $\mathbb{F}_{q^4}$. These representations can be found using exhaustive search in time $\mathrm{Poly}(q^4)$ that is much smaller than the size of E_4 and thus negligible.

Proof of Theorem 3.4: Let $E_3 = E_1 \circ E_2$ and consider any set X of vertices in E_3 of size at most $\frac{1}{2}nD$. Note that we can view the vertex set of E_3 as composed of n clusters of vertices $C_1, \ldots, C_n$, each of size D. Our goal is to show that there are at least $\frac{1}{80}\delta_1^2\delta_2 \cdot 2d|X|$ edges leaving X. We consider two cases. Either many of the vertices of X are in clusters C_i which are sparsely populated by X, in which case many edges are leaving X within the clusters C_i due to the expansion properties of E_2. Or there are many of the vertices of X which reside in densely populated clusters C_i, in which case there are many edges leaving X between the clusters, due to the expansion properties of E_1.

Set $X_i = X \cap C_i$, let $I' \subseteq [n]$ be the set of indices of the sets X_i, whose size is at most $(1 - \frac{1}{4}\delta_1)D$ and let $I'' = \{1, \ldots, n\} \setminus I'$. We first consider the contribution of the sets X_i with $i \in I'$. As E_2 is a δ_2-expander, there are at least $\frac{1}{4}\delta_1\delta_2 d|X_i|$ edges connecting X_i and $C_i \setminus X_i$. Partition X into two sets X' and X'' according to I' and I'' as follows: $X' = \bigcup_{i \in I'} X_i$ and $X'' = \bigcup_{i \in I''} X_i$. By the above, the number of edges connecting X' and $\overline{X}$ is at least $\frac{1}{4}\delta_1\delta_2 d|X'|$. If $|X'| \geq \frac{1}{10}\delta_1|X|$ then we are done, as this means that there are at least $\frac{1}{80}\delta_1^2\delta_2 \cdot 2d|X|$ edges connecting X and its complement $\overline{X}$.

Suppose then that $|X'| \leq \frac{1}{10}\delta_1|X|$, implying that $|X''| \geq (1 - \frac{1}{10}\delta_1)|X|$. We now consider the contribution of the edges leaving the sets C_i. As the sets X_i with $i \in I''$ have size at least $(1 - \frac{1}{4}\delta_1)D$ we infer that $|X''|/D \leq |I''| \leq |X''|/(1 - \frac{1}{4}\delta_1)D$. In particular, as $|X''| \leq |X| \leq \frac{1}{2}nD$ we have $|I''| \leq \frac{2}{3}n$. Therefore, as E_1 is an $[n, D, \delta_1]$-expander, there is a set of edges M, where $|M| \geq \frac{1}{2}\delta_1 D|I''|$, connecting the vertices of I'' with the vertices of I'. Let us now consider the corresponding $d|M| \geq \frac{1}{2}\delta_1 dD|I''|$ edges in the graph E_3. These edges connect vertices from $\bigcup_{i \in I'} C_i$ with vertices from $\bigcup_{i \in I''} C_i$. As each X_i with $i \in I''$ is of size at least $(1 - \frac{1}{4}\delta_1)D$, we infer that at most $\frac{1}{4}\delta_1 dD|I''|$ of these $d|M|$ edges connect a vertex in $C_i \setminus X_i$ with a vertex of $\bigcup_{i \in I'} C_i$. Therefore, there are at least $\frac{1}{4}\delta_1 dD|I''|$ edges connecting $\bigcup_{i \in I''} X_i$ with the vertices of $\bigcup_{i \in I'} C_i$. The number of these $d|M|$ edges that connect vertices from $\bigcup_{i \in I''} C_i$ with vertices of X' is clearly at most $d|X'|$. As we have $|X'| \leq \frac{1}{10}\delta_1|X| \leq \frac{1}{6}\delta_1 D|I''|$ we infer that there are at most $\frac{1}{6}\delta_1 dD|I''|$ such edges. We conclude that at least $\frac{1}{12}\delta_1 dD|I''|$ edges connect vertices of $\bigcup_{i \in I''} X_i$ (that belong to X) with vertices of $\bigcup_{i \in I'} C_i \setminus X_i$ (that belong to $\overline{X}$). As $|I''| \geq |X''|/D$ and $|X''| \geq \frac{1}{2}|X|$ this means that there are at least $\frac{1}{48}\delta_1 2d|X|$ edges connecting X and $\overline{X}$, as needed. ∎

Proof of Theorem 3.6: The proof follows by considering a Caley graph and its eigenvectors, which are the characters. Set $\mathbb{F} = \mathbb{F}_{2^t}$, $n = 2^{t(r+1)}$ and let M be the $n \times n$ adjacency matrix of $LD(2^t, r)$. Let $L : \mathbb{F} \to \{0, 1\}$ be any surjective linear map [‡]. Let us describe the eigenvectors of M over $\mathbb{R}$. We will use elements of $\mathbb{F}^{r+1}$ in order to "name" these vectors as well as to "name" entries of these vectors. For every sequence $a = (a_0, \ldots, a_r) \in \mathbb{F}^{r+1}$, let v_a be the vector, whose b^{th} entry (where $b \in \mathbb{F}^{r+1}$) satisfies $v_a(b) = (-1)^{L(\sum_{i=0}^{r} a_i b_i)}$. It is easy to see that the vectors $\{v_a\}_{a\in\mathbb{F}^{r+1}}$ are orthogonal, therefore these are the only eigenvectors of M. Clearly, $v_a(b + c) = v_a(b)v_a(c)$ for any $b, c \in \mathbb{F}^{r+1}$. Let us show that v_a is indeed an eigenvector and en-route also compute its eigenvalue.

$$(Mv_a)(b) = \sum_{c\in\mathbb{F}^{r+1}} M_{bc} \cdot v_a(c) = \sum_{x,y\in\mathbb{F}} v_a(b+y(1, x, ..., x^r)) = \left(\sum_{x,y\in\mathbb{F}} v_a(y, yx, ..., yx^r)\right) \cdot v_a(b) \ .$$

Therefore $\lambda_a = \sum_{x,y\in\mathbb{F}} v_a(y, yx, ..., yx^r)$ is the eigenvalue of v_a. Set $p_a(x) = \sum_{i=0}^{r} a_i x^i$ and write

$$\lambda_a = \sum_{x,y\in\mathbb{F}} (-1)^{L(y\cdot p_a(x))} = \sum_{\{x,y\in\mathbb{F}\ :\ p_a(x)=0\}} (-1)^{L(y\cdot p_a(x))} + \sum_{\{x,y\in\mathbb{F}\ :\ p_a(x)\neq 0\}} (-1)^{L(y\cdot p_a(x))} \ .$$

If $p_a(x) = 0$, then $(-1)^{L(y\cdot p_a(x))} = 1$ for all y, thus such an x contributes q to λ_a. If $p_a(x) \neq 0$ then $y \cdot p_a(x)$ takes on all values in $\mathbb{F}$ as y varies, and hence $(-1)^{L(y\cdot p_a(x))}$ varies uniformly over $\{-1, 1\}$ implying that these x's contribute nothing to λ_a. Therefore, when $a = 0^n$ we have $\lambda_a = q^2$. Otherwise, when $a \neq 0^n$, p_a has at most r roots, and therefore $\lambda_a \leq rq$. ■

A proof of left inequality of Theorem 3.1: Let A be the adjacency matrix of G and note that as A is symmetric we have $\lambda_2 = \max_{0\neq x\perp 1^n} \langle xA, x\rangle/\langle x, x\rangle$. For a set $S \subseteq V(G)$ let x_S be the vector satisfying $x_i = 1$ when $i \in S$ and $x_i = 0$ otherwise, and note that $\langle x_S A, x_S\rangle = 2e(S)$ and $\langle x_S A, x_{\overline{S}}\rangle = e(S, \overline{S})$. Set $x = |\overline{S}| \cdot x_S - |S| \cdot x_{\overline{S}}$ and note that $x\perp 1^n$. Therefore,

$$\lambda_2(|S| + |\overline{S}|)|S||\overline{S}| = \lambda_2\langle x, x\rangle \geq \langle xA, x\rangle = 2|S|^2 e(\overline{S}) + 2|\overline{S}|^2 e(S) - 2|S||\overline{S}|e(S, \overline{S}). \quad (3.1)$$

As G is d-regular we have $e(S) = \frac{1}{2}(d|S| - e(S, \overline{S}))$ and $e(\overline{S}) = \frac{1}{2}(d|\overline{S}| - e(S, \overline{S}))$. Plugging this into (3.1), solving for $e(S, \overline{S})$ and using $|S| \leq n/2$, we complete the proof by

[‡] For example, if we view the elements of $\mathbb{F}$ as element of $\{0, 1\}^t$ then we can define $L(a_0, a_1, \ldots, a_{t-1}) = a_0$.

inferring that

$$e(S,\overline{S}) \geq (d-\lambda_2)|S||\overline{S}|/n \geq \frac{1}{2}(d-\lambda_2)|S| \ .$$

3.3 Concluding Remarks and Open Problems

Expanders on $\Theta(n)$ vertices: Let us first show how to apply Theorem 3.5 in order to construct for every large n, an expander on $\Theta(n)$ vertices. Let N_t be the set of integers of the form q^{4r+12} where $q = 2^t$ and $q^4/100 \le r \le q^4/2$. By Theorem 3.5 we can generate an expander of size n for every $n \in \bigcup_{t=1}^{\infty} N_t$ in time poly(n). Note that for every $t \ge 2$ we have

$$\max\{N_t\} = q^{4 \cdot \frac{q^4}{2} + 12} = 2^{\lg q (2q^4 + 12)} > 2^{(\lg q + 1)(\frac{64}{100} q^4 + 12)} = (2q)^{4 \cdot \frac{(2q)^4}{100} + 12} = \min\{N_{t+1}\} \ .$$

Therefore for every $n \ge 4^{4 \cdot \frac{4^4}{100} + 12}$ there exists a t such that $n \in [\min\{N_t\}, \max\{N_t\}]$. Hence, for every such n there exists a $q = 2^t$ and $\frac{q^4}{100} \le r_0 \le \frac{q^4}{2}$ such that $n/q^4 \le q^{4r_0+12} \le n$.

Now, given n let $q = 2^t$ and $q^4/100 \le r_0 \le q^4/2$ be such that $n/q^4 \le q^{4r_0+12} \le n$ (as guaranteed by the previous paragraph). We start by using Theorem 3.5 to construct a $[q^{4r_0+12}, 12, \delta]$-expander E satisfying $n/q^4 \le q^{4r_0+12} \le n$. If $n/32 \le q^{4r_0+12}$ we return E. Otherwise set $t = \lfloor n/16q^{4r_0+12} \rfloor < q^4$ and use exhaustive search to find a 6-regular expander E' on $12t$ vertices (which exists by Theorem 3.2). This step takes time $q^{O(q^4)}$, which is polynomial in the size of E because $|E| \ge q^{\frac{1}{25}q^4}$ as $r \ge q^4/100$. We now replace every edge of E with t parallel edges to get a $[q^{4r_0+12}, 12t, \delta]$-expander E''. We then define $E'' \circ E'$ to be the final 12-regular graph on m vertices with $n/2 \le m \le n$.

Fully explicit expanders: We now show that for every t we can construct a fully explicit $[2^{t\lfloor 2^t/t \rfloor}, d, \delta]$-expander for some constants $d, \delta > 0$. Thus, for every n we can construct such an expander of size $n \le m \le n^2$. The idea is to significantly reduce the degree (say, to $\sqrt{\lg \lg n}$), so the innermost graph is fully explicit as it is very small. Then the entire construction is fully explicit as well.

We use the previous argument to find an expander of size $2^{2t} \le m \le c2^{2t}$. As noted in Section 3.1 we can then turn it into a constant degree expander E_1 of size precisely 2^{2t}. This step takes time $2^{O(t)}$. It is useful to "name" the vertices of E_1 using pairs of elements of $\mathbb{F}_{2^t}$. Set $E_2 = LD(2^t, \lfloor 2^t/t \rfloor - 3)$ and define $E_3 = E_2 \circ E_1$ as the final constant degree expander on $2^{t\lfloor 2^t/t \rfloor}$ vertices. To see that E_3 is fully explicit, note that we can view a vertex of $LD(q, r)$ as composed of $r+1$ elements of $\mathbb{F}_q$. Therefore, a vertex of $E_3 = E_2 \circ E_1$ can be viewed as $r + 1 = \lfloor 2^t/t \rfloor - 2$ elements $(a_0, \ldots, a_r)$ of $\mathbb{F}_{2^t}$ (representing a vertex of E_2) and another pair of elements x, y of $\mathbb{F}_{2^t}$ (representing a vertex of E_1). Suppose the degree of E_1 is d' in which case the degree of E_3 is $2d'$. Given $r+3$ elements $(a_0, \ldots, a_r, x, y)$ of $\mathbb{F}_{2^t}$ and $i \in [2d']$ we do the following. If $1 \le i \le d'$ we return $(a_0, \ldots, a_r, x', y')$, where (x', y') is the i^{th} neighbor of vertex (x, y) in E_1. We can do so by generating E_1 from scratch in time $2^{O(t)}$. If $d'+1 \le i \le 2d'$, we return the vertex $(a'_0, \ldots, a'_r, x, y)$, where $a'_i = a_i + yx^i$. To do so we use a representation of $\mathbb{F}_{2^t}$ that we find using exhaustive search in time $2^{O(t)}$. We finally note that one can easily adopt our arguments to get *space efficient* variants of our constructions. We omit the details.

Edge expansion close to $\frac{1}{2}$: The expanders we constructed have a positive edge expansion. However, by applying Theorem 3.1 it is easy to see that for every ϵ we can raise the graphs we construct to an appropriate power to get edge-expansion $\frac{1}{2} - \epsilon$. In fact, to get edge-expansion $\frac{1}{2} - \epsilon$ one needs the degree to be Poly$(1/\epsilon)$.

Vertex expansion: It is clear that if G is an $[n, d, \delta]$-expander, then for any set of vertices of size at most $n/2$, there are at least $\delta|S|$ vertices outside S that are connected to some vertex of S. Our construction thus also supplies constructions of vertex-expanders with expansion close to $\frac{1}{2}$. By adding loops and taking a power one can, in fact, obtain vertex expansion close to 1.

Eigenvalue gap: As we have mentioned before all the previous constructions of bounded-degree expanders did so via constructing a graph, whose second eigenvalue is bounded away from d. Theorem 3.1 implies that if G is an $[n, d, \delta]$-expander then its second largest eigenvalue is at most $d(1 - \frac{1}{2}\delta^2)$. As we can construct expanders with edge expansion close to $\frac{1}{2}$, these graphs have second largest eigenvalue at most roughly $\frac{7}{8}d$. By adding loops and raising the resulting graphs to an appropriate power one can get expanders in which all eigenvalues are, in absolute value, at most some fractional power of the degree of regularity.

Expanders with smaller degree: The expanders we construct have constant degree larger than 3. In order to get a 3-regular expander one can take any constant degree d-regular expander and apply a replacement product with a cycle of length d. Definition 3.3 implies that the new degree is 4, but it is easy to see that when d is a constant we do not have to duplicate each edge of the "large" graph d times, as keeping a single edge guarantees a positive expansion. This way we can get a 3-regular expander, which is clearly the smallest possible degree of regularity.

Simple Combinatorial Proofs. It is interesting whether there exists a simple construction of such constant degree expander, so that its analysis is entirely combinatorial, but yet simple. In particular, it is interesting to come up with a construction for an $[n, \text{Poly} \log n, \frac{1}{4}]$-expander such that its analysis is both simple and combinatorial. Finding simple combinatorial proofs for other building blocks useful in this context (e.g, graph-powering) and for (near) Ramanujan graphs is, of course, a worthy goal.

Chapter 4

On the Complexity of Approximating k-Set Packing and Related Problems

4.1 Introduction

Bounded variants of optimization problems are often easier to approximate than the general, unbounded problems. The INDEPENDENTSET problem illustrates this well: it cannot be approximated to within $O(N^{1-\varepsilon})$ unless $P = NP$ [Hås99, Zuc07]. Nevertheless, once the input graph has a bounded degree d, much better approximations exist (e.g, a $\frac{d \log \log d}{\log d}$ approximation by [Vis96]). Another example is the bounded covering problem (hyper-graph vertex-cover) which has been studied thoroughly [Hol02, DGKR05].

We next examine some bounded variants of the SETPACKING problem and try to illustrate the connection between the bounded parameters (e.g, sets size, occurrences of elements) and the complexity of the bounded problem.

In the problem of SP, the input is a family of sets $S_1, ..., S_N$, and the objective is to find a maximum packing, namely a maximum number of pairwise disjoint sets from the family. This problem is often phrased in terms of Hyper-graphs: we have a vertex v_x for each element x and a hyper-edge e_S for each set S of the family (containing all vertices v_x which correspond the elements x in the set S). The objective is to find a maximum matching. Alternatively one can formulate this problem using the dual-graph: a vertex v_S

for each set S and a hyper-edge e_x for each element (v_S is contained in all edges e_x such that $x \in S$). The objective is to find a maximum independent set (namely, a maximum number of vertices, such that no two of them are contained in the same edge).

The general problem of SP has been extensively studied (for example [Wig83, BYM84, BH92, Hås99, Zuc07]). Quite tight approximation algorithms and inapproximability factors are known for this problem. Håstad [Hås99] and Zuckerman [Zuc07] proved that Set-Packing cannot be approximated to within $O(N^{1-\varepsilon})$ unless $NP \subseteq ZPP$ and $P \neq NP$ respectively (for every $\varepsilon > 0$, where N is the number of sets). The best approximation algorithm achieves an approximation ratio of $O(\frac{N}{\log^2 N})$ [BH92]. In contrast, the case of bounded variants of this problem seems to be of a different nature.

4.1.1 Bounded Variants of Set-Packing

For bounded variant it seems natural to think of SP using hyper-graph notions. One may think of two natural bounds: the size of the edges (size of the sets) and the degree of the vertices (number of occurrences of each element). For example, k-Set-Packing is this problem where the size of the hyper-edges is bounded by k. If we also bound the degree of the vertices by two this becomes the problem of maximum IndependentSet in k bounded degree graphs denoted by k-IndependentSet(recall the dual-graph defined above).

Another natural bound is the colorability of the input graph. Consider the problem of 3-DimensionalMatching. It is a variant of $3 - SP$ where the vertices of the input hyper-graph are a union of three disjoint sets, $V = V_1 \uplus V_2 \uplus V_3$, and each hyper-edge contains exactly one vertex from each set, namely, $E \subseteq V_1 \times V_2 \times V_3$. In other words, the vertices of the hyper-graph can be colored using 3 colors, so that no hyper-edge contains the same color twice. A graph having this property is called *3-strongly-colorable* (in general - k-strongly-colorable). Thus the color-bounded version of k-SetPacking, namely the problem of k-DimensionalMatching, is

Definition 4.1 (k-DimensionalMatching). ***k-Dimensional Matching***
Input: *A k-uniform k-strongly colorable hyper-graph $H = (V^1, ..., V^k, E)$.*
Problem: *Find a matching of maximum size in H .*

These bounded variants of SP are known to admit approximation algorithms better than their general versions, the quality of the approximation being a function of the bounds.

An extensive body of algorithmic work has been devoted to these restricted problems (for example, [HS89]). Matching inapproximability results have also been studied (e.g, implicit in [Tre01]).

With some abuse of notations, one can say that hardness of approximation factor of SP is a monotonous increasing function in each of the bounded parameters: the edges size, the vertices degree and the colorability (of edges and vertices). For example, inapproximability factor for graphs of degree bounded by 3 holds for graphs with degree bounded by 4. We next try to overview what is known regarding the complexity of this problem as a function of these bounds.

4.1.2 Related Studies

2-DM is known to be solvable in polynomial time, say by a reduction to network flow problems [Pap94]. Polynomial time algorithms are also known for graphs that are not bipartite [Edm65]. In contrast, for all $k \geq 3$, k-DimensionalMatching is NP-hard [Kar72, Pap94]. Furthermore, for $k = 3$, the problem is known to be APX-hard [Kan91].

For large k values, we are usually interested in the asymptotic dependence of the approximation ratio (and inapproximability factor) on k. Currently, the best polynomial time approximation algorithm for k-SetPacking achieves an approximation ratio of $\frac{k}{2}$ [HS89]. This is, to date, the best approximation algorithm for k-DimensionalMatching as well.

Alon et al. [AFWZ95] proved that for suitably large k, k-IS is NP-hard to approximate to within $k^c - \varepsilon$ for some $c > 0$. This was later improved to the currently best asymptotical inapproximability factor [Tre01] of $\frac{k}{2^{\Omega(\sqrt{\ln k})}}$. All hardness factors for k-IS hold in fact for $(k+1)$-DM as well (by a simple reduction). The best known approximation algorithm for k-IS achieves an approximation ratio of $O(k \log\log k / \log k)$ [Vis96].

4.1.3 Our Contribution

We improve the inapproximability factor for the k-SetPacking, and show:

Theorem 4.2. *It is NP-hard to approximate k-*SetPacking *to within* $O\left(\frac{k}{\ln k}\right)$

These results extend to k-DimensionalMatching and IndependentSet in $(k+1)$-claw-free graphs *($(k+1)$-ISCFG)* (see [Hal98] for definition of $(k+1)$-ISCFG and reduction from k-SetPacking). They do not hold, however, for k-IS.

4.1.4 Outline

Some preliminaries are given in section 4.2. Section 4.2.1 presents the notion of hypergraph-dispersers. Section 4.3 contains the proof of the asymptotic hardness of approximation for k-SetPacking. Section 4.4 extends the proof to hold for k-DimensionalMatching. The existence of a good hyper-disperser is proved in section 4.5. The optimality of its parameter is shown in the same section. Section 4.6 contains a discussion on the implications of our results, the techniques used and some open problems.

4.2 Preliminaries

Our main result in this chapter is derived by a reduction from the following problem.

Definition 4.3 (Linear Equations). ***MAX-3-LIN-q*** *is the following optimization problem:*
Input: *A set* Φ *of linear equations modulo an integer* q*, each depending on 3 variables.*
Problem: *Find an assignment that satisfies the maximum number of equations.*

The following central theorem stems from an extensive line of research, using the PCP theorem (see [AS92, ALM$^+$98]) and the parallel repetition theorem [Raz98] as a starting point:

Theorem 4.4 (Håstad [Hås01]). *Gap-*$MAX-3-LIN-q$*-*$\left[\frac{1}{q}+\varepsilon, 1-\varepsilon\right]$ *is* NP*-hard for every* $q \in \mathbb{N}$ *and* $\varepsilon > 0$*. Furthermore, the result holds for instances of MAX-3-LIN-q in which the number of occurrences of each variable is a constant (depending on* ε *only).*

We denote an instance of MAX-3-LIN-q by $\Phi = \{\varphi_1, ..., \varphi_m\}$. Φ is over the set of variables $X = \{x_1, ..., x_n\}$. Let $\Phi(x)$ be the (multi) set of all equations in Φ depending on x (i.e. it can be seen as all the occurrences of x). Denote by $Sat(\Phi, A)$ the set of all equation in Φ satisfied by the assignment A. For an assignment A to an equation $\varphi \in \Phi(x)$, we denote by $A[\varphi]_{|x}$ the corresponding assignment to x.

We next explain the reduction from Linear equations to our problem. The reduction gives an inapproximability factor for k-SetPacking. We later amend it to hold for k-DimensionalMatching too.

4.2.1 Dispersers and Hyper Dispersers

Recall the definition of disperser-graphs (for further definitions and results regarding dispersers see [RTS00]):

Definition 4.5 ((n, m, d, k, δ)-Disperser). *We call a bipartite graph* $G = (V_1, V_2, E)$ *an* (n, m, d, k, δ)*-Disperser if it is* d *left regular and* $|V_1| = n, |V_2| = m$ *and every subset of size at least* k *of* V_1 *is connected to at least* $1-\delta$ *fraction of* V_2*.*

In this chapter we are more interested in balanced bipartite dispersers, with k equals to $(1-\delta)n$. We therefore use the following definition:

Definition 4.6 (δ-Disperser). *We call a balanced bipartite graph $G = (V_1, V_2, E)$ a δ-Disperser if every large independent-set $I \subseteq \{V_1 \cup V_2\}$ of G is (almost) concentrated in one part of the vertices. Formally, there exists i so that*

$$|I \setminus V_i| \leq \delta |V|$$

Consider the following generalization of disperser graphs:

Definition 4.7 ((q, δ)-Hyper-Disperser). *We call a hyper graph $H = (V, E)$ a (q, δ)-Hyper-Disperser if there exists a partition of its vertices: $V = V_1 \cup\!\!\!\cdot \ldots \cup\!\!\!\cdot\ V_q$, $|V_1| = \ldots = |V_q|$, such that every large independent-set I of H is (almost) concentrated in one part of the vertices. Formally, there exists i so that*

$$|I \setminus V_i| \leq \delta |V|$$

And consider the dual definition:

Definition 4.8 ((q, δ)-Hyper-Edge-Disperser). *We call a hyper graph $H = (V, E)$ a (q, δ)-Hyper-Edge-Disperser if there exists a partition of its edges: $E = E_1 \cup \ldots \cup \cdot\ E_q$, $|E_1| = \ldots = |E_q|$, such that every large matching M of H is (almost) concentrated in one part of the edges. Formally, there exists i so that*

$$|M \setminus E_i| \leq \delta |E|$$

Note that this generalizes the notion of dispersers: if we take two (of the q) parts of the edges E_i, E_j and every vertex that appears in the intersection of an edge from E_i with an edge of E_j, then the resulting graph is the dual of a δ-disperser graph. See section 4.5.1 for further discussion.

Lemma 4.9. *For every $q > 1$ and $t > 1$ there exists a hyper-graph $H = (V, E)$ such that*

- *$V = [t] \times [d]$, whereas $d = \Theta(q \ln q)$.*
- *H is $(q, \frac{1}{q^2})$-hyper-edge-disperser*
- *H is d-uniform, d-strongly-colorable.*
- *H is q-regular, q-strongly-edge-colorable.*

We denote this graph by $(t, q) - \mathcal{D}$ *and name its edges* $e[i, j]$ *where* $j \in [q]$ *is the color of the edge by an arbitrary strong edge coloring (a coloring where no two edges of the same color share a vertex) and* $i \in [t]$ *is an arbitrary indexing of the* t *edges of each color. Note that the* t *edges of any single color, exactly cover all the vertices of* $(t, q) - \mathcal{D}$.

For proof see section 4.5. Note that $(t, 2) - \mathcal{D}$ is the dual graph of a standard disperser.

Dispersers and Expanders. When comparing expanders to dispersers it may be more convenient to consider the bipartite variant of both. In some sense, every such expander is a disperser (but not the other way round). Both have some expansion properties. For expanders we claim that every (not too large) set of on part expands on the other part, and therefore large set on one side hits a large set on the other side. For dispersers a large set on one side hits a large set on the other side, but for small sets no expansion is guaranteed.

Where in some reductions one needs to utilize the good mixing of expanders (e.g., Chapter 2), in other cases hitting properties are sufficient (e.g., Chapter 4). Note, however, that the dispersers schemed in this chapter are also good expanders.

4.3 Proof of the Asymptotic Inapproximability Factor for k-SetPacking

This section provides a deterministic polynomial time reduction from MAX-3-LIN-q to k-SetPacking. The constructed instance of k-SetPacking is a hyper-graph with a hyper-edge for each equation and a satisfying assignment to it. In addition, the graph will have common vertices for edges that correspond to contradicting assignments. Thus, intuitively, a large matching should translate to a consistent satisfying assignment.

The sparsity and uniformity of the constructed graph ultimately relate to the quality of the hardness result. In order to obtain a sparse graph with small edge size, but still retain edge-intersection properties, we utilize a form of expander graphs defined in the previous section.

4.3.1 The construction

Let $\Phi = \{\varphi_1, ..., \varphi_n\}$ be an instance of MAX-3-LIN-q over the sets of variables X,where each variable $x \in X$ occurs a constant number of times c_x (recall Theorem 4.4). We now describe how to deterministically construct, in polynomial time, an instance of k-SetPacking - the hyper-graph $H_\Phi = (V, E)$.

For every variable $x \in X$ we have a copy of a hyper-edge disperser $(c_x, q) - \mathcal{D}$ (which exist by lemma 4.9), which is denoted by $\mathcal{D}_x$. The vertices of H_Φ are the union of the vertices of all these hyper-disperses (recall that $d = \Theta(q \lg q)$):

$$V \;=\; X \times [c_x] \times [d]$$

as $c_x = |\Phi(x)|$ we use $\varphi \in \Phi$ to denote an occurrence of a variable, namely,

$$V = \{v_{x,\varphi,i} \mid x \in X, \varphi \in \Phi(x), i \in [d]\}$$

The Edges of H_Φ. We have an edge for each equation $\varphi \in \Phi$ and a satisfying assignment to it. Consider an equation $\varphi = x + y + z = a \mod q$, and a satisfying assignment A to that equation (note that there are q^2 such assignments, as assigning the first two variables, determines the third). The corresponding edge, $e_{\varphi,A}$, is composed of three edges, one from the hyper-graph $\mathcal{D}_x$, one from $\mathcal{D}_y$ and the last from $\mathcal{D}_z$. Formally:

$$e_{\varphi,A} = e_{x,\varphi,A_{|x}} \uplus e_{y,\varphi,A_{|y}} \uplus e_{z,\varphi,A_{|z}}$$

Where $A_{|x}$ is the restrictions of the assignment A to the variable x, and $e_{x,\varphi,A_{|x}}$ is the edge $e[\varphi, A_{|x}]$ of $\mathcal{D}_x$ (and similarly for y and z). In other words, an edge $e[i,j]$ of a hyper-edge-disperser $\mathcal{D}_x$ (the i^{th} edge of color j) is related to assigning j to x in its i^{th} occurrence in $\Phi(x)$; the vertices of $e[i,j]$ are included in every edge that correspond to assigning j to x in its i^{th} occurrence in $\Phi(x)$.

The edges of H_Φ are

$$E = \{e_{\varphi,A} \mid \varphi \in \Phi, A \text{ is a satisfying assignment to } \varphi\}$$

Clearly, the cardinality of $e_{\varphi,A}$ is $3d$ (and note that each of the three composing edges participates in creating q edges). This concludes the construction.

Notice that the construction is indeed deterministic, as each variable occurs a constant number of times (see Theorem 4.4). Hence, the sizes of $\mathcal{D}_x$ is constant and its existence (see lemma 4.9) suffices, as one can enumerate all possible hyper-graphs, and verify their properties.

Claim 4.10. *[Completeness] If there is an assignment to Φ which satisfies $1-\varepsilon$ of its equations, then there is a matching in H_Φ of size $\left(\frac{1-\varepsilon}{q^2}\right)|E|$.*

Proof. Let A be an assignment that satisfies $1-\varepsilon$ of the equations. Consider the matching $M \subseteq E$ comprised of all edges corresponding to A, namely

$$M = \{e_{\varphi,A(\varphi)} \mid \varphi \in Sat(\Phi, A)\}$$

Trivially, $|M| = \left(\frac{1-\varepsilon}{q^2}\right)|E|$, as we took one edge corresponding to each satisfied equation. To see that these edges are indeed a matching take any two edges of M. If they do not relate to the same variables then they do not contain vertices from a joint hyper-edge-disperser. On the other hand, if they do relate to a joint variable, then they relate to different occurrences i_1, i_2, but the same assignment $j \in [q]$ to it. Hence they contain vertices of the same hyper-edge-disperser $\mathcal{D}_x$, but from two distinct edges of the same color $(e[i_1,j], e[i_2,j])$, so they do not share a vertex. ∎

Lemma 4.11. *[Soundness] If every assignment to Φ satisfies at most $\frac{1}{q}+\varepsilon$ fraction of its equations, then every matching in H_Φ is of size $O\left(\frac{1}{q^3}|E|\right)$.*

Proof. Denote by E_x the edges of H_Φ corresponding to equations φ containing the variable x, namely,

$$E_x = \{e_{\varphi,A} \mid \varphi \in \Phi(x), e_{\varphi,A} \in E\}$$

Denote by $E_{x=a}$ the subset of E_x corresponding to an assignment of a to x, that is,

$$E_{x=a} = \{e_{\varphi,A} \mid e_{\varphi,A} \in E_x, A_{|x} = a\}$$

Let M be a matching of maximum size in H_Φ. Let A_{maj} be the most popular assignment. That is, for every $x \in X$ choosing the assignment of x to be such that it corresponds to maximum number of edges. Formally, choose

$$A_{maj}(x) \in [q] \text{ s.t. } |E_{x=a} \cap M| \text{ is maximized}$$

Let M_{maj} be the set of edges in M that agree with A_{maj}, and M_{min} be all the other edges in M, namely

$$M_{maj} = \{e_{\varphi,A_{maj}}\}_{\varphi\in\Phi}$$

$$M_{min} = M \setminus M_{maj}$$

As $|Sat(\Phi, A_{maj})| \leq \frac{1}{q}+\varepsilon$, we have $|M_{maj}| < (\frac{1}{q}+\varepsilon)\frac{E}{q^2}$.
For every $x \in X$, $\mathcal{D}_x$ is a $(q, \frac{1}{q^2})$-hyper-edge-disperser. That is, in a subset of edges of $\mathcal{D}_x$ which is a matching, all but at most $\frac{1}{q^2}$ of the edges are of one color. Clearly, if two edges e_1 and e_2 of $\mathcal{D}_x$ intersect, then so do any two edges containing e_1 and e_2 respectively. Hence,

$$\sum_{a\neq A_{maj}(x)} |M_{min} \cap E_{x=a}| \leq \frac{1}{q^2}E(\mathcal{D}_x)$$

Every edge of $\mathcal{D}_x$ is a subset of q hyper edges in E_x. However, no more than one of these q edges may be taken to M (as M is a matching). Therefore,

$$\sum_{a\neq A_{maj}(x)} |M_{min} \cap E_{x=a}| \leq \frac{1}{q^3}|E_x|$$

$$|M_{min}| \leq \sum_{x \in X, a \neq A_{maj}(x)} |M_{min} \cap E_{x=a}| \leq \frac{1}{q^3} \sum_{x \in X} |E_x| = \frac{3}{q^3}|E|$$

and thus

$$|M| = |M_{min}| + |M_{maj}| \leq (\frac{4}{q^3} + \varepsilon)|E|$$

■

By claim 4.10 and lemma 4.11 we showed that $Gap\text{-}k - SetPacking\text{-}\left[\frac{4}{q^3} + \varepsilon, \frac{1}{q^2} - \varepsilon\right]$ is NP-hard. Since each edge is of size $k = 3d = \Theta(q \log q)$ it is NP-hard to approximate k-SetPacking to within $O(\frac{k}{\ln k})$.

4.4 Extending the Proof for k-DimensionalMatching

The proof for k-DimensionalMatching follows the steps of the proof for k-SetPacking, the difference being that we use three dispersers for each variable (instead of one) - a different disperser for each location in the equations. Denote by $\Phi(x,l)$ the subset of $\Phi(x)$ where x is the l'th variable in the equation (clearly $l \in [3]$). Note that w.l.o.g. we may assume that for every $x \in X$, $\Phi(x,1) = \Phi(x,2) = \Phi(x,3)$ (as we can take three copies of each equation, and shift the location of the variables).

For every variable $x \in X$ and position $l \in [3]$, we have a hyper disperser $(\frac{c_x}{3}, q) - \mathcal{D}$ (as stated in lemma 4.9), which is denoted by $\mathcal{D}_{x,l}$.

$$V \quad = \quad X \times V(\mathcal{D}_x) \times [3]$$

namely,

$$V = \{v_{x,\varphi,i} \mid x \in X, \varphi \in \Phi(x), i \in [d]\}$$

where the index $i \in [q]$ is given by a strong-coloring of the edges with q colors (recall that such a coloring exists as $(t,q) - \mathcal{D}$ is q-strongly colorable).

The Edges of H_Φ. We have an edge for each equation $\varphi \in \Phi$ and a satisfying assignment to it. Consider an equation $\varphi = x + y + z = a \mod q$, and a satisfying assignment A to that equation. The corresponding edge, $e_{\varphi,A}$, is composed of three edges, one from the hyper-graph $\mathcal{D}_{x,1}$, one from $\mathcal{D}_{y,2}$ and the last from $\mathcal{D}_{z,3}$. Formally:

$$e_{\varphi,A} = e_{x,\varphi,A_{|x}} \uplus e_{y,\varphi,A_{|y}} \uplus e_{z,\varphi,A_{|z}}$$

Where $e_{x,\varphi,A_{|x}}$ is the edge $e[\varphi, A_{|x}]$ of $\mathcal{D}_{x,1}$, $e_{y,\varphi,A_{|y}}$ is the edge $e[\varphi, A_{|y}]$ of $\mathcal{D}_{y,2}$ and $e_{z,\varphi,A_{|z}}$ is the edge $e[\varphi, A_{|z}]$ of $\mathcal{D}_{z,3}$. The edges of H_Φ are

$$E = \{e_{\varphi,A} \mid \varphi \in \Phi, A \text{ is a satisfying assignment to } \varphi\}$$

This concludes the construction for k-DimensionalMatching. We next show that the graph constructed is indeed a k-DimensionalMatching instance:

Proposition 4.12. *H_Φ is $3d$-strongly-colorable.*

Proof. We show how to partition V into $3d$ independent sets of equal size. Let the sets be $P_{l,i}$ whereas $i \in [d]$ and $l \in [3]$:

$$P_{l,i} = \{v_{x,\varphi,i} \mid x \in X, \varphi \in \Phi(x,l)\}$$

$P_{l,i}$ is clearly a partition of the vertices, as each vertex belongs to a single part.
We now explain why each part is an independent set. Let $P_{l,i}$ be an arbitrary part, and let $e_{\varphi,A} \in E$ be an arbitrary edge, where $\varphi \equiv x + y + z = a \mod q$:

$$e_{\varphi,A} = e_{x,\varphi,A[\varphi]_{|x}} \uplus e_{y,\varphi,A[\varphi]_{|y}} \uplus e_{z,\varphi,A[\varphi]_{|z}}$$

$P_{l,i} \cap e_{\varphi,A}$ may contain vertices corresponding only to one of the variables x, y, z, since it contains variables corresponding to a single location (first, second or third). Let that variable be, w.l.o.g, x. The edge $e_{x,\varphi,A[\varphi]_{|x}}$ contains exactly one vertex from each of the d parts, as the graph $D_{x,1}$ is d-partite. Therefore, the set $P_{l,i} \cap e_{\varphi,A}$ contains exactly one vertex. Since $|P_{l,i} \cap e_{\varphi,A}| = 1$ for every edge and every set $P_{l,i}$, the graph H_Φ is $3d$-partite-balanced. ■

The completeness claim for k-SetPacking (claim 4.10) holds here too. The soundness lemma for k-SetPacking holds with minor changes:

Lemma 4.13. *[Soundness] If every assignment to Φ satisfies at most $\frac{1}{q} + \varepsilon$ fraction of its equations, then every matching in G is of size $O\left(\frac{1}{q^3}E\right)$.*

Proof. We repeat the soundness proof of k-SetPacking but the definition of the most-popular assignment is slightly different, and takes into account the three different dispersers per variable.
Denote by $E_{x,l}$ the edges of H_Φ corresponding to equations φ containing the variable x in location l, namely,

$$E_{x,l} = \{e_{\varphi,A} \mid \varphi \in \Phi(x,l), A \in [q^2]\}$$

Denote by $E_{x=a,l}$ the subset of $E_{x,l}$ corresponding to an assignment of a to x, that is,

$$E_{x=a,l} = \{e_{\varphi,A} \mid \varphi \in \Phi(x,l), A[\varphi]_{|x} = a\}$$

Let M be a matching of maximum size. Let A_{maj} be the most popular of most popular assignment. That is, for every $x \in X$ choose the location (of equations of edges of M) in which x appears maximum number of times,

$$\widehat{l}(x) \in [3] \text{ s.t. } |E_{x,\widehat{l}(x)} \cap M| \text{ is maximized} \tag{4.1}$$

Then choose an assignment for x such that it corresponds to maximum number of those edges. Formally, choose

$$A_{maj}(x) \in [q] \text{ s.t. } |E_{x=a,\widehat{l}(x)} \cap M| \text{ is maximized}$$

As before, let M_{maj} be the set of edges in M that agree with A_{maj}, and M_{min} be all the other edges in M, namely

$$M_{maj} = \{e_{\varphi,A_{maj}}\}_{\varphi\in\Phi}$$

$$M_{min} = M \setminus M_{maj}$$

For the exact same reasons as in the k-SetPacking proof, we have

$$|M_{maj}| < (\frac{1}{q} + \varepsilon)\frac{|E|}{q^2} \tag{4.2}$$

and for every x,

$$\sum_{a\neq A_{maj}(x)} |M_{min} \cap E_{x=a,\widehat{l}(x)}| \le \frac{1}{q^3}|E_{x,\widehat{l}(x)}| \tag{4.3}$$

Therefore,

$$\begin{aligned}
|M| &= \sum_{x,l} |M \cap E_{x,l}| \\
&\le \sum_{x,l} |M_{maj} \cap E_{x,l}| + \sum_{x,l,a\neq A_{maj}(x)} |M_{min} \cap E_{x=a,l}| \text{by (4.1) we have} \\
&\le 3 \cdot \sum_{x} |M_{maj} \cap E_{x,\widehat{l}(x)}| \; + \; 3 \cdot \sum_{x,a\neq A_{maj}(x)} |M_{min} \cap E_{x=a,\widehat{l}(x)}| \\
&\le 3 \cdot |M_{maj}| \; + \; 3 \cdot \sum_{x,a\neq A_{maj}(x)} |M_{min} \cap E_{x=a,\widehat{l}(x)}|
\end{aligned}$$

thus by (4.2) and (4.3)

$$\begin{aligned} &< 3(\frac{1}{q} + \varepsilon)\frac{|E|}{q^2} + \frac{3}{q^3}\sum_x |E_{x,\widehat{l}(x)}| \\ &= (\frac{12}{q^3} + 3\varepsilon)|E| \end{aligned}$$

■

By claim 4.10 and lemma 4.13 we showed that $Gap\text{-}k - DimensionalMatching\text{-}\left[\frac{12}{q^3} + 3\varepsilon, \frac{1}{q^2} - \varepsilon\right]$ is NP-hard, thus it is NP-hard to approximate k-DIMENSIONALMATCHING to within $O(\frac{k}{\ln k})$.

4.5 Hyper-Edge-Dispersers

In this section, we prove lemma 4.9. As stated before, these are generalizations of disperser graphs. In section 4.5.1, we prove that these are the best (up to a constant) parameters for a Hyper-Edge-Disperser one can hope to achieve.

Lemma 4.9 *For every $q > 1$ and $t > 1$ there exists a hyper-graph $H = (V, E)$ such that*

- $V = [t] \times [d]$, *whereas* $d = \Theta(q \ln q)$.
- H *is* $(q, \frac{1}{q^2})$*-hyper-edge-disperser*
- H *is* d*-uniform,* d*-strongly-colorable.*
- H *is* q*-regular,* q*-strongly-edge-colorable.*

We denote this graph by $(t, q) - \mathcal{D}$.

Proof. We follow the probabilistic method to prove that the probability that a randomly generated graph is not a $(t, q) - \mathcal{D}$ graph, is strictly smaller then 1, from which follows the existence of such graphs. Let

$$V = [t] \times [d]$$

and denote $V_i = [t] \times \{i\}$.
We next randomly construct the edges of the hyper-graph, so that it is d-uniform, q-regular. Let S_t be all permutation over t elements, and let

$$\Pi_{i_1,i_2} \in_R S_t \ , \ (i_1, i_2) \in [q] \times [d]$$

(that is, qd permutations, chosen uniformly from S_t). Define

$$e[i, j] = \{ \ (\Pi_{j,1}(i), 1), \ (\Pi_{j,2}(i), 2) \ , ..., \ (\Pi_{j,d}(i), d) \ \} \tag{4.4}$$

and let

$$E = \{e[i, j] \mid (i, j) \in [t] \times [q]\}$$

Hence $|E| = tq$. Define a partition of the edges as follows: $E_j = \{e[i, j] \mid i \in [t]\}$. Thus $|E_1| = ... = |E_q| = t$ and each set of edges E_j covers every vertex exactly once. Therefore,

H is q strongly-edge-colorable. On the other hand, every edge contains exactly one vertex from each set of vertices V_i. Thus H is d-strongly-colorable.

We next show that with high probability H has the disperser property, namely, every matching M of H is concentrated on a single part of the edges, except for maybe $\frac{1}{q^2}|E| = \frac{t}{q}$ edges of M. Denote by P the probability that H does *not* have the disperser property.

Definition 4.14. *Let $\mathcal{M}$ be the family of all subsets $M \subseteq E$ such that:*

$$\mathcal{M} = \{M \mid M \subseteq E, |M| = \frac{t}{q} + \frac{t}{q^2}, \exists i, |M \setminus E_i| = \frac{t}{q}\}$$

Proposition 4.15. *If all sets $M \in \mathcal{M}$ are not matchings, then H is a $(q, \frac{1}{q^2})$-hyper-edge-disperser.*

Proof. This follows from the downward monotonicity of the matching property. To be more precise, suppose that H is not a $(q, \frac{1}{q^2})$-hyper-edge-disperser. Namely, there exists a matching $M' \subseteq E$ such that it is not concentrated on one color of edges: $\forall i$, $|M' \setminus E_i| > \frac{1}{q^2}|E| = \frac{t}{q}$. Let i be so that $|M' \cap E_i|$ is maximal, and hence larger than $\frac{t}{q^2}$. As any subset of a matching is a matching, we can remove edges of $M' \cap E_i$ until we are left with exactly $\frac{t}{q^2}$ edges there, and remove edges from $M' \setminus E_i$ until we are left with exactly $\frac{t}{q}$ edges there. But this new set is in $\mathcal{M}$, thus cannot be a matching. ∎

Following the above proposition, we proceed with the proof considering only sets in $\mathcal{M}$. Denote by $\Pr[M]$ the probability (over the random choice of H) that M is a matching. By union bound,

$$P = \Pr_H[\exists M \in \mathcal{M},\ M \text{ is a matching }] \leq$$

$$\leq \sum_{M \in \mathcal{M}} \Pr[M] \leq |\mathcal{M}| \Pr[\widehat{M}] \tag{4.5}$$

where $\widehat{M} \in \mathcal{M}$ is the set which maximizes $\Pr[\widehat{M}]$. Clearly (using the known inequality $\binom{n}{k} \leq (\frac{en}{k})^k$),

$$|\mathcal{M}| \leq q\binom{(q-1)t}{\frac{t}{q}}\binom{t}{\frac{t}{q^2}} \leq q(eq^2)^{\frac{t}{q}}(eq^2)^{\frac{t}{q^2}} \leq (eq)^{\frac{3t}{q}} \tag{4.6}$$

We next bound $\Pr[\widehat{M}]$. Let $M_i = \widehat{M} \cap E_i$. Let $B_{i,j}$ be the event that the sets of edges M_i and M_j do not share a vertex, and $A_i = \cap_{j<i} B_{i,j}$. Then

$$\Pr[\widehat{M}] = \Pr\left[\bigcap_i A_i\right] = \prod_i \Pr\left[A_i \mid \bigcap_{l<i} A_l\right]$$

Note however, that the event A_i is independent of the event $\bigcap_{l<i} A_l$ as A_i is determined by (the independently chosen permutations) $\{\Pi_{i,j} \mid j \in [d]\}$, whereas $\bigcap_{l<i} A_l$ is determined by the permutations $\{\Pi_{l,j} \mid l < i, j \in [d]\}$. Thus

$$\Pr[\widehat{M}] = \prod_i \Pr[A_i] \tag{4.7}$$

Let $C_{i,j}$ be the event that there is no collision (common vertex) of M_i and $\bigcup_{l<i} M_l$ on the subset of vertices V_j (clearly $A_i = \bigcap_{j\in[d]} C_{i,j}$). Hence, as for $j_1 \neq j_2$, C_{i,j_1} and C_{i,j_2} are determined by independent sets of permutations (recall (4.4)) we have

$$\Pr[A_i] = \prod_{j\in[d]} \Pr[C_{i,j}] = (\Pr[C_{i,1}])^d \leq (1 - \frac{|M_i|}{t})^{d|\bigcup_{l<i} M_l \cap V_1|} = (1 - \frac{|M_i|}{t})^{d\sum_{l<i}|M_l|}$$

where the sum in the exponent of the rightmost expression is by assuming no collisions between edges of $\bigcup_{l<i} M_l$ on V_j (which is implied by $\bigcap_{l<i} A_l$). Thus by equation (4.7) we have (as $1 - x \leq e^{-x}$):

$$\Pr[\widehat{M}] \leq \prod_i \left(1 - \frac{|M_i|}{t}\right)^{d\sum_{j<i}|M_j|} \leq e^{-\frac{d}{t}\sum_{i=2}^{q}(|M_i|\sum_{j=1}^{i-1}|M_j|)}$$

Under the constraint that $\widehat{M} \in \mathcal{M}$ the sum $\sum_{i=2}^{q}(|M_i|\sum_{j=1}^{i-1}|M_j|)$ is minimized for $|M_2| = |M_3| = \frac{t}{2q}$ hence

$$\Pr[\widehat{M}] \leq e^{-\frac{dt}{4q^2}} \tag{4.8}$$

Therefore by equations (4.5),(4.6),(4.8),

$$P \leq (eq)^{\frac{3t}{q}} e^{-\frac{dt}{4q^2}}$$

Any d which guarantees that $(eq)^{\frac{3t}{q}} e^{-\frac{dt}{4q^2}} \ll 1$ suffices (for example $d \geq 20q \ln q$) as $P < 1$, thus there exists H with the disperser properties.

■

4.5.1 Optimality of Hyper-Edge-Disperser Construction

We now turn to see why the Hyper-Edge-Disperser from lemma 4.9 has optimal parameters. We base our observation on a lemma from [RTS00]:

Definition 4.16. *A bipartite graph $G = (V_1, V_2, E)$ is called a δ-disperser if for every $U_1 \subseteq V_1, U_2 \subseteq V_2, |U_1|, |U_2| \geq \delta|V_1| = \delta|V_2|$, the subset $U_1 \cup U_2$ is* not *an independent set.*

Lemma 4.17. *Every bipartite d-regular $\frac{1}{k}$-disperser must satisfy $d = \Omega(k \ln k)$.*

Proposition 4.18. *Every d-uniform q-strongly-edge-colorable q-regular d-strongly colorable $(q, \frac{1}{q^2})$-hyper-edge-disperser must satisfy $d = \Omega(q \ln q)$.*

Proof. We prove that if there exists such a hyper-graph which satisfies $d = o(q \ln q)$, then there exists a bipartite $o(q \ln q)$-regular $\frac{1}{q}$-disperser, in contrast to lemma 4.17. We transform a d-partite d-uniform q-regular q-strongly-edge colorable $(q, \frac{1}{q^2})$-Hyper-Edge-Disperser $H = (V_H, E_1, E_2, ..., E_q)$ into a bipartite d-regular $\frac{1}{q}$-disperser $G = (V_1, V_2, E_G)$ in the following way. Let

$$V_1 = E_1$$

$$V_2 = E_2$$

$$E_G = \{(e_1, e_2) \mid e_1 \cap e_2 \neq \phi\}$$

Obviously G is a bipartite d-regular graph (we allow multi-edges). In addition, suppose two sets of fractional sizes:

$$S_1 = \frac{1}{q} V_1, S_2 = \frac{1}{q} V_2$$

are an independent set in G. Then the corresponding sets of edges in H are disjoint and are of fractional size $\frac{2}{q^2}$, thus contradicting the fact that H is a $(q, \frac{1}{q^2})$-Hyper-Edge-Disperser.
■

4.6 Concluding Remarks and Open Problems

An interesting property of our construction (for both asymptotic and low bound values results) is the *almost perfect completeness*. This property refers to the fact that the matching proved to exist in the completeness claim 4.10 is an almost perfect matching, that is, it covers $1 - \varepsilon$ of the vertices. Knowing the location of a gap is interesting by itself and may prove useful (in particular if it is extreme on either the completeness or the soundness parameters, see for example [Pet94]). In fact, applying our reduction on other PCP variants instead of Max-3-Lin-q (e.g. parallel repetition of 3-SAT) yields perfect completeness for k-DimensionalMatching (but with weaker hardness factors).

The ratio between the asymptotic inapproximability factor presented herein for k-DimensionalMatching and k-SetPacking, and the tightest approximation algorithm known, was reduced to $O(\ln k)$. The open question of where in the range, from $\frac{2}{k}$ to $O(\frac{\ln k}{k})$ is the approximability threshold is interesting by itself, as well as its implications to the difference between k-DimensionalMatching and k-IS. The current asymptotic inapproximability factor of $O(\frac{\ln k}{k})$ for k-DimensionalMatching approaches the tightest approximation ratio known for k-IS, namely $\Omega\left(\frac{\log k}{k \log \log k}\right)$ [Vis96]. Thus, a small improvement in either the approximation ratio or the inapproximability factor will show these problems to be of inherently different complexity.

Bibliography

[AAS06] A. Avidor, A. Armon, and O. Schwartz. Cooperative TSP. In *Proc. 14th Annual European Symposium on Algorithms (ESA)*, pages 40–51, 2006.

[AFWZ95] N. Alon, U. Fiege, A. Wigderson, and D. Zuckerman. Derandomized Graph Products. *Computational Complexity*, 5:60–75, 1995.

[Ajt94] M. Ajtai. Recursive Construction for 3-Regular Expanders. *Combinatorica*, 14(4):379–416, 1994.

[AKK+08] S. Arora, S. A. Khot, A. Kolla, D. Steurer, M. Tulsiani, and N. K. Vishnoi. Unique Games on Expanding Constraint Graphs are Easy. To appear in Proc. 40th ACM Symp. on Theory of Computing, 2008.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and Intractability of Approximation Problems. *Journal of the ACM*, 45:501–555, 1998.

[Alo86] N. Alon. Eigenvalues and Expanders. *Combinatorica*, 6:83–96, 1986.

[AM85] N. Alon and V. D. Milman. λ_1, Isoperimetric Inequalities for Graphs and Superconcentrators. *J. Combinatorial Theory, Ser. B*, 38:73–88, 1985.

[AR94] N. Alon and Y. Roichman. Random Cayley Graphs and Expanders. *Random Struct. Algorithms*, 5(2):271–285, 1994.

[AR05] D. Aharonov and O. Regev. Lattice Problems in NP intersect coNP. *Journal of the ACM*, 52(5):749–765, 2005.

[AS92] S. Arora and S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. In *Proc. 33rd IEEE Symp. on Foundations of Computer Science*, pages 2–13, 1992.

[ASS07] N. Alon, O. Schwartz, and A. Shapira. An Elementary Construction of Constant Degree Expanders. In *Proc. 18th ACM-SIAM symposium on Discrete algorithms*, pages 454–458, 2007.

[Ban93] W. Banaszczyk. New Bounds in some Transference Theorems in the Geometry of Numbers. *Mathematische Annalen*, 296(4):625–635, 1993.

[BASTS07] A. Ben-Aroya, O. Schwartz, and A. Ta-Shma. An Explicit Construction of Quantum Expanders. arXiv:0709.0911v1 [quant-ph], 2007.

[BH92] R. Boppana and M. M. Halldorsson. Approximating Maximum Independent Sets by Excluding Subgraphs. *Bit 32*, pages 180–196, 1992.

[BHZ87] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP Have Short Interactive Proofs? *Information Processing Letters*, 25:127–132, 1987.

[BLar] Y. Bilu and N. Linial. Lifts, Discrepancy and Nearly Optimal Spectral Gaps. *Combinatorica*, to appear.

[Bog05] A. Bogdanov. Gap Amplification Fails Below 1/2, 2005. Comment on ECCC TR05-046.

[Bou02] J. Bourgain. On the Distribution of the Fourier Spectrum of Boolean Functions. *Israel J. of Math.*, 131:269–276, 2002.

[BYM84] R. Bar-Yehuda and S. Moran. On Approximation Problems Related to the Independent Set and Vertex Cover Problems. *Discrete Applied Mathematics*, 9:1–10, 1984.

[Che52] H. Chernoff. A Measure of Asymptotic Efficiency for Tests of a Hypothesis Based on the Sum of Observations. *Annals of Mathematical Statistics*, 23:493–507, 1952.

[CMM06] M. Charikar, K. Makarychev, and Y. Makarychev. Near-Optimal Algorithms for Unique Games. In *Proc. 38th ACM Symp. on Theory of Computing*, pages 205–214, New York, NY, USA, 2006. ACM Press.

[Coo71] S. A. Cook. The Complexity of Theorem-Proving Procedures. In *Proc. 3rd ACM Symp. on Theory of Computing*, pages 151–158, 1971.

[DGKR05] I. Dinur, V. Guruswami, S. A. Khot, and O. Regev. A New Multilayered PCP and the Hardness of Hypergraph Vertex Cover. *SIAM Journal of Computing*, 34(5):1129–1146, 2005.

[Din07] I. Dinur. The PCP Theorem by Gap Amplification. *Journal of the ACM*, 54(3):12, 2007.

[Dod84] J. Dodziuk. Difference Equations, Isoperimetric Inequality and Transience of Certain Random Walks. *Trans. Amer. Math. Soc.*, 284:787–794, 1984.

[DS02] I. Dinur and S. Safra. The Importance of Being Biased. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 33–42, New York, 2002. ACM Press.

[Edm65] J. Edmonds. Paths, Trees and Flowers. *Canadian Journal of Mathematics*, 17:449–467, 1965.

[Fei95] U. Feige. Error Reduction by Parallel Repetition – The State of the Art. Technical report, The Weizmann Institute, 1995.

[FGL$^+$96] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive Proofs and the Hardness of Approximating Cliques. *J. ACM*, 43(2):268–292, 1996.

[FK00] U. Feige and J. Kilian. Two-Prover Protocols—Low Error at Affordable Rates. *SIAM Journal of Computing*, 30(1):324–346, 2000.

[FKO07] U. Feige, G. Kindler, and R. O'Donnell. Understanding Parallel Repetition Requires Understanding Foams. In *CCC '07: Proceedings of the Twenty-Second Annual IEEE Conference on Computational Complexity*, pages 179–192, Washington, DC, USA, 2007. IEEE Computer Society.

[FL92] U. Feige and L. Lovåsz. Two-Prover One-Round Proof Systems: their Power and their Problems (Extended Abstract). In *Proc. 24th ACM Symp. on Theory of Computing*, pages 733–744, New York, NY, USA, 1992. ACM Press.

[FRS88] L. Fortnow, J. Rompel, and M. Sipser. On the Power of Multi-Prover Interactive Protocols. In *Structure in Complexity Theory Conference*, pages 156–161, 1988.

[FRS90] L. Fortnow, J. Rompel, and M. Sipser. Errata for On the Power of Multi-Prover Interactive Protocols. In *Structure in Complexity Theory Conference*, pages 318–319, 1990.

[GG81] O. Gabber and Z. Galil. Explicit Constructions of Linear-Sized Superconcentrators. *J. Comput. Syst. Sci.*, 22(3):407–420, 1981.

[GG00] O. Goldreich and S. Goldwasser. On the Limits of Nonapproximability of Lattice Problems. *Journal of Computer and System Sciences*, 60(3):540–563, 2000.

[Gro83] M. Gromov. Filling Riemannian manifolds. *J. of Differential Geometry*, 18(1):1–147, 1983.

[GT06] A. Gupta and K. Talwar. Approximating Unique Games. In *Proc. 17th ACM-SIAM symposium on Discrete algorithms*, pages 99–106, New York, NY, USA, 2006. ACM Press.

[GW95] M. X. Goemans and D. P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. Assoc. Comput. Mach.*, 42:1115–1145, 1995.

[Hal98] M. M. Halldorsson. Approximations of Independent Sets in Graphs. *APPROX*, 1998.

[Hås99] J. Håstad. Clique is Hard to Approximate within $n^{1-\epsilon}$. *Acta Math.*, 182(1):105–142, 1999.

[Hås01] J. Håstad. Some Optimal Inapproximability Results. *Journal of the ACM*, 48(4):798–859, 2001.

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander Graphs and Their Applications. *Bulletin of the AMS*, 43(4):439–561, 2006.

[Hol02] J. Holmerin. Vertex Cover on 4-Regular Hyper-Graphs is Hard to Approximate Within $2 - \varepsilon$. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 544–552, 2002.

[Hol07] T. Holenstein. Parallel Repetition: Simplifications and the No-Signaling Case. In *Proc. 39th ACM Symp. on Theory of Computing*, pages 411–419, New York, NY, USA, 2007. ACM Press.

[HS89] C. A. J. Hurkens and A. Schrijver. On the Size of Systems of Sets Every T of which have an SDR, with an Application to the Worst-Case Ratio of Heuristics for Packing Problems. *SIAM Journal Discrete Math*, 2:68–72, 1989.

[HSS06] E. Hazan, O. Schwartz, and S. Safra. On the Hardness of Approximating k-Set-Packing. *Computational Complexity*, 15(1):20–39, 2006.

[Kan91] V. Kann. Maximum Bounded 3-Dimensional Matching is Maxsnp-Complete. *Information Processing Letters*, 37:27–35, 1991.

[Kar72] R. M. Karp. Reducibility Among Combinatorial Problems. *Complexity of Computer Computations*, pages 83–103, 1972.

[Kho02] S. A. Khot. On the Power of Unique 2-Prover 1-Round Games. In *Proc. 34th ACM Symp. on Theory of Computing*, pages 767–775, New York, NY, USA, 2002. ACM Press.

[Kho05] S. A. Khot. Guest Column: Inapproximability Results via Long Code Based PCPs. *SIGACT News*, 36(2):25–42, 2005.

[KKMO04] S. A. Khot, G. Kindler, E. Mossel, and R. O'Donnell. Optimal Inapproximability Results for Max-Cut and Other 2-Variable CSPs? In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS'04)*, pages 146–154, Washington, DC, USA, 2004. IEEE Computer Society.

[KR03] S. A. Khot and O. Regev. Vertex Cover Might be Hard to Approximate to within $2 - \varepsilon$. In *Proc. of 18th IEEE Annual Conference on Computational Complexity (CCC)*, pages 379–386, 2003.

[Lad75] R. Ladner. On the Structure of Polynomial Time Reducibility. *Journal of the ACM*, 22:155–171, 1975.

[Lev73] L. A. Levin. Universal Search Problems. *Problemy Peredachi Informatsii*, 9(3):265–266, 1973.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan Graphs. *Combinatorica*, 8(3):261–277, 1988.

[Mar73] G. A. Margulis. Explicit Constructions of Expanders. *Problemy Peredav ci Informacii*, 9:71–80, 1973.

[Mar88] G. A. Margulis. Explicit Group-Theoretical Constructions of Combinatorial Schemes and Their Application to the Design of Expanders and Superconcentrators. *Problems of Information Transmission*, 24:39–46, 1988.

[Pap94] C. H. Papadimitriou. *Computational Complexity.* Addison Wesley, 1994.

[Pet94] E. Petrank. The Hardness of Approximation: Gap Location. *Computational Complexity*, 4(2):133–157, 1994.

[Pin73] M. Pinsker. On the Complexity of a Concentrator. In 7^{th} *annual teletraffic conference*, volume 8, pages 1–4, 1973.

[PY88] C. Papadimitriou and M. Yannakakis. Optimization, Approximation, and Complexity Classes. In *Proc. 20th ACM Symp. on Theory of Computing*, pages 229–234, New York, NY, USA, 1988. ACM Press.

[Raz95] R. Raz. Personal communication to U. Feige. Appears in [Fei95], 1995.

[Raz98] R. Raz. A Parallel Repetition Theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998.

[Raz07] R. Raz. Personal communication. 2007.

[Raz08] R. Raz. A Counterexample to Strong Parallel Repetition, 2008. ECCC Report TR08-018.

[RTS00] J. Radhakrishnan and A. Ta-Shma. Bounds for Dispersers, Extractors, and Depth-Two Superconcentrators. *SIAM Journal on Discrete Mathematics*, 13:2–24, 2000.

[RVW02] O. Reingold, S. Vadhan, and A. Wigderson. Entropy Waves, the Zig-Zag Graph Product, and New Constant-Degree Expanders. *Annals of Mathematics*, 155(1):157–187, 2002.

[Sch87] C. P. Schnorr. A Hierarchy of Polynomial Time Lattice Basis Reduction Algorithms. *Theoretical Computer Science*, 53(2-3):201–224, 1987.

[SS05] S. Safra and O. Schwartz. On the Complexity of Approximating TSP with Neighborhoods and Related Problems. *Computational Complexity*, 14:281–307, 2005.

[SS07] S. Safra and O. Schwartz. On Parallel-Repetition, Unique-Game and Max-Cut. Submitted, 2007.

[Tre01] L. Trevisan. Non-approximability Results for Optimization Problems on Bounded Degree Instances. *Proc. 33rd ACM Symp. on Theory of Computing*, 2001.

[Tre05] L. Trevisan. Approximation Algorithms for Unique Games. In *Proc. 46nd IEEE Symp. on Foundations of Computer Science*, pages 197–205, Washington, DC, USA, 2005. IEEE Computer Society.

[TSSW00] L. Trevisan, G. B. Sorkin, M. Sudan, and D. P. Williamson. Gadgets, Approximation, and Linear Programming. *SIAM Journal on Computing*, 29(6):2074–2097, 2000.

[Vis96] S. Vishwanathan. Personal communication to M. Halldorsson. Cited in [Hal98], 1996.

[Wig83] A. Wigderson. Improving the Performance Guarantee for Approximate Graph Coloring. *Journal of the Association for Computing Machinery*, 30(4):729–735, 1983.

[Zuc07] D. Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(6):103–128, 2007.