Electronic Colloquium on Computational Complexity, Report No. 29 (2010)

Spectral Algorithms for Unique Games

Alexandra Kolla*

Abstract

We present a new algorithm for Unique Games which is based on purely *spectral* techniques, in contrast to previous work in the area, which relies heavily on semidefinite programming (SDP). Given a highly satisfiable instance of Unique Games, our algorithm is able to recover a good assignment. The approximation guarantee depends only on the completeness of the game, and not on the alphabet size, while the running time depends on spectral properties of the *Label-Extended* graph associated with the instance of Unique Games.
In particular, we show how our techniques imply a quasi-polynomial time algorithm that decides satisfiability of a game on the Khot-Vishnoi([KV05]) integrality gap instance. Notably, when run on that instance, the standard SDP relaxation of Unique Games *fails*. As a special case, we also show how to re-derive a polynomial time algorithm for Unique Games on expander constraint graphs (similar to [AKK$^+$08]) and a sub-exponential time algorithm for Unique Games on the Hypercube.

*School of Mathematics, Institute for Advanced Study. Research supported by NSF Grant CCF-0832797.

ISSN 1433-8092

1 Introduction

A Unique Game is defined in terms of a constraint graph $G = (V, E)$, a set of variables $\{x_u\}_{u \in V}$, one for each vertex u and a set of permutations (constraints) $\pi_{uv} : [k] \to [k]$, one for each edge (u, v). An assignment to the variables is said to satisfy the constraint on the edge $(u, v) \in E$ if $\pi_{uv}(x_u) = x_v$. The edges are taken to be undirected and hence $\pi_{uv} = (\pi_{vu})^{-1}$. The goal is to assign a value from the set $[k]$ (alphabet) to each variable x_u so as to maximize the number of satisfied constraints.

Khot [Kho02] conjectured that it is NP-hard to distinguish between the cases when almost all the constraints of a unique game are satisfiable and when very few of the constraints are satisfiable. Formally, the statement of the conjecture is the following:

Conjecture 1 *(Unique Games Conjecture-*UGC*) For any constants $\epsilon, \delta > 0$, there is a $k(\epsilon, \delta)$ such that for any $k > k(\epsilon, \delta)$, it is NP-hard to distinguish between instances of unique games with alphabet size k where at least $1 - \epsilon$ fraction of constraints are satisfiable and those where at most δ fraction of constraints are satisfiable.*

Since its origin, the UGC has been successfully used to prove (often optimal) hardness of approximation results for several important NP-hard problems such as MIN-2SAT-DELETION [Kho02], Vertex Cover [KR03], Maximum Cut [KKMO04] and non-uniform Sparsest Cut [CKK+06, KV05]. In addition, in recent years, UGC has also proved to be intimately connected to the limitations of semidefinite programming (SDP). Making this connection precise, [Aus07, Rag08] shows that if UGC is true, then for every constraint satisfaction problem (CSP) the best approximation ratio is given by a certain simple SDP.

For arbitrary graphs, several approximation algorithms using linear and semidefinite programming have been developed for approximating Unique Games (see [Kho02], [Tre05], [GT06], [CMM06a], [CMM06b]). These algorithms start with an instance where the value of the SDP or LP relaxation is $1 - \epsilon$ and round it to a solution with value ν. Here, value of the game refers to the maximum fraction of satisfiable constraints. For $\nu > \delta$, this would give an algorithm to distinguish between the two cases. However, most of these algorithms give good approximations only when ϵ is very small ($\epsilon = O(1/\log n)$ or $\epsilon = O(1/\log k)$) and their approximation guarantee depends on the alphabet size k [1]. For constant ϵ however, only the algorithm of [CMM06a] gives interesting parameters with $\nu \approx k^{-\epsilon/(2-\epsilon)}$. We refer the reader to [CMM06a] for a comparison of parameters of various algorithms. In some special cases, however, it has been shown that there are efficient algorithms that solve Unique Games. One such example is when the constraint graph is a *spectral expander*. In that case, [AKK+08] (and later [MM09] with improved parameters) showed that one can find a highly satisfying assignment (with, say, $\nu \geq 90\%$) in polynomial time. Notably, the approximation guarantee of their algorithm depends on the expansion parameters of the graph rather than the alphabet size. One notion of expansion that the authors in [AKK+08] consider is when the second smallest eigenvalue of the normalized Laplacian of a graph G, denoted by $\lambda = \lambda_2(\mathcal{L}_G)$ and their algorithmic requirement is that $\lambda \gg \epsilon$. As another example, the authors in [AIMS09] presented efficient algorithms for constraint graphs with large *local expansion*.

So far, as seen by the previous line of work, the most promising techniques for developing good approximation algorithms for Unique Games have consistently been based on linear or semidefinite

[1] It might be good to think of k as $O(\log n)$ since this is the range of interest for most reductions

programming. However, it seems like semidefinite programming algorithms are nearing a bottleneck, as characterized by the existance of *integrality gap* instances [KV05, RS09] for semidefinite programming relaxations for Unique Games that belong to a certain natural family of SDPs. Those instances are Unique Games instances that *fool* the SDP since, even though they have no good satisfying assignment, the respective SDP solution is high. The existence of such instances implies, in particular, that there is no hope for a "good" approximation algorithm based on those SDPs that solves Unique Games on the underlying constraint graphs. We note that, in an unpublished manuscript, [KT07] developed a *spectral* algorithm that runs in polynomial time and finds highly satisfying assignments on expander constraint graphs. With the realization of the limitations of several natural SDPs and the existence of good spectral algorithms a natural question rears its head: is it possible that there are spectral algorithms for Unique Games that perform better than SDPs?

Our results and Comparison to Previous Work

In this paper, we make progress toward answering the above question. Namely, we present a purely *spectral* algorithm for Unique Games that finds highly satisfying assignments when they exist. The running time of our algorithm depends on spectral properties of the *Label-Extended* graph associated with the instance of Unique Games. Our algorithm runs in polynomial or quasi-polynomial time for a large class of constraint graphs, including graphs where the standard SDP provably fails and expander graphs. At a high level, we show that given $\epsilon > 0$, there is a $\delta = \delta(\epsilon)$ such that the algorithm is able to distinguish between the following two cases:

- **YES** case: There exists an assignment that satisfies $(1 - \epsilon)$ fraction of the constraints.
- **NO** case: Every assignment satisfies less that δ fraction of the constraints.

In particular, our algorithm runs in quasi-polynomial time on input the underlying constraint graph of the SDP integrality gap instance as it appears in [KV05] (hereafter denoted $\mathcal{KV}$). We note that the authors in [KV05] roughly showed that, when run on a certain highly unsatisfiable instance of Unique Games with underlying constraint graph $\mathcal{KV}$, the standard SDP relaxation has very high objective value and consequently fails to distinguish between the two cases above. As another special case, our algorithm runs in sub-exponential time on the Hypercube constraint graph and in polynomial time when the constraint graph is an expander (therefore implying results similar to [AKK+08, KT07] as a special case). Moreover, similarly to [AKK+08, MM09, KT07, AIMS09], the performance of the algorithm does not depend on the alphabet size k.

Our main contribution is a technique to analyze the full spectrum and eigenvectors of the relevant graph rather than just the first few eigenvalues. We believe that this approach is of independent interest and could contribute to developing better algorithms for a number of other problems. We also believe that our results give evidence that spectral techniques might be a more powerful tool than SDPs for designing algorithms for Unique Games, and hope that the present work will result in further exploration of the use of graph spectra towards understanding the validity of the Conjecture. We present the algorithm and the precise statements for its performance and running time in section 3. In section 4, we present the algorithms for the Khot-Vishnoi graph and the Hypercube graph.

2 Preliminaries

2.1 Spectra of Graphs

We remind the reader that for a graph G, the adjacency matrix $A = A_G$ is defined as

$$A_G(u,v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{if } (u,v) \notin E \end{cases}$$

We will also be dealing with weighted graphs, where edge (u,v) has weight $w_{uv} \geq 0$ and the adjacency matrix is defined as $A_G(u,v) = w_{uv}$. We will also assume, w.l.o.g. that $w_{uv} \leq 1$ (if not, we just re-scale the weights of the edges by the maximum weight). If the graph has n vertices, A_G has n real eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$. The eigenvectors that correspond to these eigenvalues form an orthonormal basis of $\mathbb{R}^n$. We note that if the graph is d-regular (the total weight of edges adjacent to every node is d) then the largest eigenvalue is equal to d and the corresponding eigenvector is the all-one's vector.

2.2 The Label-Extended Graph

For a given instance of Unique Games on a constraint graph $G = (V,E)$, let A be the adjacency matrix of G and let M denote the $nk \times nk$ symmetric matrix such that the $k \times k$ block M_{uv} is equal to $w_{uv}\Pi_{uv}$. Here Π is the matrix of the permutation π_{uv}. Then M is the adjacency matrix of the *Label-Extended* graph for that instance of Unique Games. Let k denote the size of the alphabet. We will denote this instance by $\mathcal{U} = (G, M, k)$.

2.3 Γ-Max-Lin Instances

Let Γ be an abelian group. A Γ-Max-Lin instance of Unique Games on graph $G(V,E)$ has, for each edge $(u,v) \in E$ a constraint of the form $x_u - x_v = c_{uv}$, where x_u, x_v are variables taking values in Γ and $c_{uv} \in \Gamma$. The alphabet k is the size of the group Γ.

2.4 The Spectrum of Cayley Graphs

For the purposes of this paper, we will use a generalized definition for Cayley graphs. For background on the standard definition and properties of Cayley Graphs see, for example, [Kas02, Spi].

Definition 2 *(Adjacency matrix of a Cayley graph) Let Ω be a group with n elements. The vertex set of a Cayley graph $C(\Omega)$ is the group Ω. The adjacency matrix of $C(\Omega)$ is the $n \times n$ matrix $A_{C(\Omega)}$ defined as follows: For $g_1, g_2 \in \Omega$, $A_{C(\Omega)}(g_1, g_2) = f(g_1 - g_2)$, where "$-$" refers to the group operation and f is some function. Note that the above definition allows $C(\Omega)$ to have weighted edges, with the only constraint being that the weight of an edge between g_1 and g_2 depends on their group-theoretic difference.*

We note that the standard definition of a Cayley graph, given a set S of generators, is a special case of the above, where $f(g_1 - g_2) = 1$ if $g_1 - g_2 \in S$ and 0 otherwise.

For what follows, we will only be interested in Ω being abelian. It is well known, that for the abelian case the eigenvectors of $A_{C(\Omega)}$ are the group-theoretic *characters* (e.g. [Kas02]). However, since definition 2 is slightly different than the usual, we repeat the proof for completeness.

Lemma 3 *Let $C(\Omega)$ be Cayley graph of an abelian group Ω, as defined in 2, and $A_{C(\Omega)}$ its adjacency matrix. Then for every $g \in \Omega$, $A_{C(\Omega)}\chi_g = |\Omega|\hat{f}(g)\chi_g$. Namely, every character χ_g is an eigenvector of $A_{C(\Omega)}$ and $|\Omega|\hat{f}(g)$ is its corresponding eigenvalue. Here, with $\hat{f}$ we denote fourier coefficients.*

PROOF: It is enough to prove the equality for each entry of $A_{C(\Omega)}\chi_g$, that is for every vertex $x \in \Omega$ we will show $\{A_{C(\Omega)}\chi_g\}_x = |\Omega|\hat{f}(g)\chi_g(x)$. We calculate:

$$\sum_{y\in\Omega} A_{C(\Omega)}(x,y)\chi_g(y) = \sum_{y\in\Omega} f(x-y)\chi_g(y) = \sum_{z\in\Omega} f(z)\chi_g(x)\chi_g(-z) = |\Omega|\hat{f}(g)\chi_g(x)$$

□

2.5 The Khot-Vishnoi Graph

In [KV05], the authors considered the following family of graphs ($\mathcal{KV}$ Graphs):

For parameters n and ϵ, let $N = 2^n$ and $n = 2^k$. Denote by $\mathcal{F}$ the family of all boolean functions on $\{-1,1\}^k$. For $f, g \in \mathcal{F}$ define the product fg as $(fg)(x) = f(x)g(x)$. Consider the equivalence relation $\equiv$ on $\mathcal{F}$ defined as $f \equiv g$ iff there is an $S \subseteq [k]$ such that $f = g\chi_S$. Here $H = \{\chi_S | S \subseteq [k]\}$ denote the set of characters of the group ${\mathbf{F}_2}^k$. This relation partitions $\mathcal{F}$ into equivalence classes $\mathcal{P}_1, \cdots, \mathcal{P}_m$, where $m = \frac{N}{n} = \frac{2^n}{n}$. For each equivalence class $\mathcal{P}_i$, we could pick an arbitrary representative $p_i \in \mathcal{P}_i$, so that

$$\mathcal{P}_i = \{p_i\chi_S | S \subseteq [k]\}$$

For $\rho > 0$ let $f \in_\rho \mathcal{F}$ denote a random boolean function on $\{-1,1\}^k$ where for every $x \in \{-1,1\}^k$, independently, $f(x) = 1$ with probability $1-\rho$ and $f(x) = -1$ with probability ρ. For the given parameter ϵ and boolean function $f, g \in \mathcal{F}$ let

$$\mathbf{wt}_\epsilon(\{f,g\}) := \mathbf{Pr}_{f' \in_{1/2}\mathcal{F}, \mu \in_\epsilon \mathcal{F}}\big[(\{f = f'\} \wedge \{g = f'\mu\}) \vee (\{g = f'\} \wedge \{f = f'\mu\})\big]$$

The $\mathcal{KV}_{n,\epsilon}$ graph with parameters n and ϵ can now be defined to have vertex set $V = \{\mathcal{P}_1, \cdots, \mathcal{P}_m\}$ and there is an edge between the vertices $x = \mathcal{P}_i$ and $y = \mathcal{P}_j$ with weight

$$A(x,y) = \sum_{f\in\mathcal{P}_i, g\in\mathcal{P}_j} \mathbf{wt}_\epsilon(\{f,g\})$$

It will be useful, for the purposes of this paper, to consider an equivalent definition of $\mathcal{KV}$, by translating to the $\{0,1\}$ language. Let $H_n = \{0,1\}^n$ the group ${\mathbf{F}_2}^n$ with addition (modulo 2) as group operation. Then the set H is the *Hadamard* code on n bits, and it is a subgroup of H_n. The graph $\mathcal{KV}_{n,\epsilon}$ is just the quotient group of H_n by H, $Q = H_n / H$. The vertex set V consists of the $\frac{N}{n}$ cosets of H and the edge between two cosets x, y has weight

$$A(x,y) = \sum_{h_1,h_2 \in H} \epsilon^{|x+h_1+y+h_2|}(1-\epsilon)^{|n-(x+h_1+y+h_2)|}$$

Note that the graph is n-regular, namely the total weight of edges adjacent to any node x is

$$\sum_{y\in Q}\sum_{h_1,h_2\in H} \epsilon^{|x+h_1+y+h_2|}(1-\epsilon)^{|n-(x+h_1+y+h_2)|} = n\sum_{y\in Q}\sum_{h_2\in H} \epsilon^{|x+y+h_2|}(1-\epsilon)^{|n-(x+y+h_2)|}$$

$$= \quad n\sum_{z\in H_n} \epsilon^{|z|}(1-\epsilon)^{|n-z|} = n\sum_{i=0}^{n}\binom{n}{i}\epsilon^i(1-\epsilon)^{n-i} = n(1-\epsilon+\epsilon)^n = n$$

It is immediate from the above description that $\mathcal{KV}$ is a Cayley graph of Q, since $A(x,y)$ depends only on $x - y = x + y$.
The following appears in [KV05]. We give the informal statement here, for simplicity. We refer to the reader to [KV05] for the full statement and details.

Claim 4 *(Integrality Gap for Unique Games, Informal Statement) Let n be an integer and $\epsilon > 0$ be a parameter. Then there is an instance of Unique Games with underlying graph $\mathcal{KV}_{n,\epsilon}$, and alphabet size n such that the following hold:*

- *Every labeling $L : V \to n$ satisfies* at most *a $\frac{1}{n^\epsilon}$ fraction of the edges.*
- *The standard SDP relaxation for Unique Games has objective value greater than $1 - 9\epsilon$.*

3 Recovering Solutions by Spectral Methods

In this section, we will show how, given a $(1-\epsilon)$ satisfiable instance of Unique Games $\mathcal{U} = (G, M, k)$, the eigenvectors of M may be used to recover good assignments. We assume, as earlier, that G is a (possibly weighted) graph with weight $1 \geq w_{uv} \geq 0$ on edge (u, v). For simplicity of the presentation, we will also assume that G is a regular graph, namely $\sum_v w_{uv} = d$ for all vertices u. The results can easily be generalized to non-regular graphs by considering the eigenvectors of the matrix $D - M$ instead of M. Here D denotes an $nk \times nk$ diagonal matrix with $D_{uu} = deg(u) \cdot I$. If we think of M as the adjacency matrix of graph with vertex set $V \times [k]$ and each edge of G replaced by a matching, then $D - M$ can be thought of as the Laplacian matrix of that graph. We will include the full proof for non-regular graph in the final version of the paper.

We show the following:

Theorem 5 *(Main) Let $\mathcal{U} = (G, M, k)$ be a $(1-\epsilon)$ satisfiable instance of Unique Games and W the eigenspace of M with eigenvalues greater than $(1-\gamma)d$, for $\gamma \geq 2\epsilon$. There is an algorithm that runs in time $2^{O(\frac{\gamma}{\epsilon} dim(W))}$ and finds an assignment that satisfies at least $(1 - O(\frac{2\epsilon}{\gamma - 2\epsilon} + \epsilon))$ fraction of the constraints.*

In particular, the theorem implies that for every $1 - \epsilon$ satisfiable instance of Unique Games that satisfies the assumptions of the theorem, one can find an assignment that satisfies more than 90 percent of the constraints in time that depends on the spectral profile of M.

For the sake of the results in section 4, we also consider the special case where the constraints are arbitrary Γ-Max-Lin and the first few eigenvectors of G satisfy some "smoothness" condition.

Theorem 6 *Let $\mathcal{U} = (G, M, k)$ be a $(1-\epsilon)$ satisfiable Γ-Max-Lin instance of Unique Games and $S_{(1-\gamma)}$ the eigenspace of G with eigenvalues greater than $(1-\gamma)d$. Assume moreover that there is an eigenbasis of G such that every unit-length eigenvector $\phi \in S_{1-\gamma}$ has $\|\phi\|_\infty \leq \frac{C}{\sqrt{n}}$ for some constant C. Then, for $\gamma = \Omega(\sqrt[3]{\epsilon})$, there is an algorithm that runs in time $2^{O(k \cdot D_S)}$ and finds an assignment that satisfies at least $(1-\zeta)$ fraction of the constraints for some $\zeta \leq 0.1$. Here D_S denotes the dimension of $S_{1-\gamma}$.*

Note that the algorithm in theorem 6 has running time that solely depends on the spectral profile of G.

We note that the result for expander graphs as it appears in [KT07] can be derived as a corollary of theorem 6. Again, we present the result for regular graphs for simplicity.

Corollary 7 *(Unique Games are Easy on Expanders) Let $\mathcal{U} = (G, M, k)$ be a $(1-\epsilon)$ satisfiable Γ-Max-Lin instance of Unique Games. Assume, moreover that G is a d-regular spectral expander. Namely, the second eigenvalue of the adjacency matrix of G is $\lambda \leq (1-\gamma)d$, for $\gamma = \Omega(\sqrt[3]{\epsilon})$. Then, there is a polynomial time algorithm that finds an assignment that satisfies at least $(1-\zeta)$ fraction of the constraints for some $\zeta \leq 0.1$.*

Proof:(Sketch) The eigenspace $S_{(1-\gamma)}$ of theorem 6 consists solely of the all 1's vector. The ℓ_∞ norm assumption of theorem 5 is trivially satisfied with $C = 1$. Then, the conclusion of the theorem implies that for $\gamma = \Omega(\sqrt[3]{\epsilon})$, there is an algorithm that runs in time $2^{O(k \cdot 1)} = 2^{O(k)}$ and finds an assignment that satisfies at least 90 per cent of the constraints. Since k is at most logarithmic in n, the algorithm runs in $2^{O(\log n)} = \mathrm{poly}(n)$ time. $\square$

As another corollary of theorem 6, we also obtain a result for any Cayley graph of an abelian group G of size n: the eigenvectors of a Cayley graph are just the *characters* of the underlying group (normalized by $1/\sqrt{n}$). For every character, its coordinates are roots of unity and therefore they all have magnitude equal to 1. We get the assumption of the theorem by normalizing by $1/\sqrt{n}$. Namely:

Corollary 8 *(Cayley Graphs) Let G be a Cayley graph of an abelian group of size n. Let $\mathcal{U} = (G, M, k)$ be a $(1-\epsilon)$ satisfiable Γ-Max-Lin instance of Unique Games and $S_{(1-\gamma)}$ the eigenspace of G with eigenvalues greater than $(1-\gamma)$. Then, if $\gamma = \Omega(\sqrt[3]{\epsilon})$, there is an algorithm that runs in time $2^{O(k \cdot D_S)}$ and finds an assignment that satisfies at least $(1-\zeta)$ fraction of the constraints for some $\zeta \leq 0.1$. Here D_S denotes the dimension of $S_{1-\gamma}$.*

We will use the corollary later on to derive quasi-polynomial time algorithms for Unique Games on the Khot-Vishnoi graph and sub-exponential time algorithms for Unique Games on the Hypercube.

Proof Overview. Our proof generalizes the approach that the authors in the unpublished manuscript [KT07] used to recover satisfying assignments on expanders. Assume we are given a $(1-\epsilon)$ satisfiable instance $\mathcal{U} = (G, M, k)$ that satisfies the assumptions of theorem 5. First, let us define a completely satisfiable game $\widetilde{\mathcal{U}} = (G, \widetilde{M}, k)$ (with value 1) that "corresponds" to $\mathcal{U}$ as follows:

Definition 9 *(Completion of a Game) Let $\mathcal{U} = (G, M, k)$ as above and let $\mathcal{L} = \{L(u)\}$ be an assignment that satisfies $(1-\epsilon)$ fraction of the constraints. Let $e = (u, v)$ be an edge with constraint $x_v = \pi_{uv}(x_u)$ that is not satisfied by this assignment, with labels $L(u)$, $L(v)$ on its endpoints. Construct a new game by changing the constraint on every such edge e by replacing π_{uv} with a permutation $\tilde{\pi}_{uv}$ for which $\tilde{\pi}_{uv}(L(u)) = L(v)$. Let $\widetilde{\mathcal{U}} = (G, \widetilde{M}, k)$ be this new completely satisfiable game. We say that $\widetilde{\mathcal{U}}$ is a "completion" of $\mathcal{U}$.*

For the sake of the proof, we will assume that an *almost* satisfiable instance $\mathcal{U} = (G, M, k)$ is constructed as follows:

- Let $\widetilde{\mathcal{U}} = (G, \widetilde{M}, k)$ be a completion of $\mathcal{U}$.
- Let an adversary pick the ϵ fraction of edges that were unsatisfied in $\mathcal{U}$ and change their constraints back to the original ones. We can now think of M as a "perturbation" of $\widetilde{M}$ and $\mathcal{U}$ as the "perturbed" game of $\widetilde{\mathcal{U}}$.

Let W the span of the eigenvectors of M with eigenvalue at least $(1-\gamma)d$, for some $\gamma \geq 2\epsilon$. The algorithm simply looks at a set $\mathcal{S} \subseteq W$ of appropriately many candidate vectors and "reads-off" an assignment. We describe later on how to choose the set $\mathcal{S}$.

$\texttt{Recover-Solution}_{\mathcal{S}}(\mathcal{U})$

- For each $x \in \mathcal{S}$, construct an assignment L_x by assigning to each vertex u, the index corresponding to the largest entry in the block $(x_{u1}, \ldots, x_{uk})$ i.e. $L_x(u) = \arg\max_i x_{ui}$.
- Out of all assignments L_x for $x \in \mathcal{S}$, choose the one satisfying the maximum number of constraints.

To choose $\mathcal{S}$, we look at the highest eigenvectors for the matrix $\widetilde{M}$. Those are the *assignment* eigenvectors, namely the characteristic vectors of the (perfectly) satisfying assignments of $\widetilde{\mathcal{U}}$ (since they all have eigenvalue equal to d). We will first observe that every such eigenvector y is close to some vector in W, and the length of the projection of y onto W depends on ϵ, γ . We then identify a set of *nice* vectors $\mathcal{N} \subseteq W$ such that the above algorithm works for any vector v close to some vector in $\mathcal{N}$. These are going to be the vectors that are close to the assignment eigenvectors. We then construct a set $\mathcal{S} \subseteq W$ of test vectors such that at least one vector $v \in \mathcal{S}$ is close to a vector in $\mathcal{N}$. Lastly, we go over all vectors in $\mathcal{S}$ until we find v. The running time of the algorithm will depend on the size of $\mathcal{S}$.

3.1 Proof of Theorem 5

For the matrix $\widetilde{M}$, let $\mathcal{L} = \{L(u)\}_{u \in V(G)}$ be a completely satisfying assignment. Define vector $y^{(\mathcal{L})}$ as

$$y_{ui}{}^{(\mathcal{L})} = \begin{cases} \frac{1}{\sqrt{n}} & \text{if} \quad i = L(u) \\ 0 & \text{otherwise} \end{cases}$$

It is easy to see that $y^{(\mathcal{L})}$ is an eigenvector of $\widetilde{M}$ with eigenvalue d. Since $y^{(\mathcal{L})}$ corresponds to the satisfying assignment $\mathcal{L}$, it can be seen as the characteristic vector of the assignment. We refer to such vectors as the "assignment" eigenvectors. Our next goal is to show that, for some appropriate choice of γ, the eigenspace W contains vectors close to an assignment vector.

Claim 10 *(Closeness) For every completely satisfying assignment $\mathcal{L}$ there is a unit vector $v_{\mathcal{L}} \in W$ such that $v_{\mathcal{L}} = \alpha y^{(\mathcal{L})} + \beta y^{(\mathcal{L})}{}_{\perp}$, with $|\beta| \leq \sqrt{\frac{2\epsilon}{\gamma}}$. By taking, for example, $\gamma \geq 200\epsilon$, we have $|\beta| \leq \frac{1}{10}$.*

Proof: We can easily see that $(y^{(\mathcal{L})})^T M y^{(\mathcal{L})} \geq d(1-2\epsilon)$. We can now write $y^{(\mathcal{L})} = a v_{\mathcal{L}} + b(v_{\mathcal{L}})_{\perp}$, with $v_{\mathcal{L}} \in W$ and $(v_{\mathcal{L}})_{\perp} \in W^{\perp}$. We calculate:

$$(1-2\epsilon)d \leq (y^{(\mathcal{L})})^T M y^{(\mathcal{L})} = a^2 (v_{\mathcal{L}})^T M v_{\mathcal{L}} + b^2((v_{\mathcal{L}})_{\perp})^T M (v_{\mathcal{L}})_{\perp} \leq a^2 d + b^2(1-\gamma)d$$

from which we get that $|b| \leq \sqrt{\frac{2\epsilon}{\gamma}}$.

Now, we can in turn express $v_{\mathcal{L}} = \alpha y^{(\mathcal{L})} + \beta y^{(\mathcal{L})}{}_{\perp}$, where $\alpha = \langle v_{\mathcal{L}}, y^{(\mathcal{L})} \rangle = |a| = \sqrt{1-b^2} \geq \sqrt{1-\frac{2\epsilon}{\gamma}}$ or, equivalently, $|\beta| = \sqrt{1-\alpha^2} \leq \sqrt{\frac{2\epsilon}{\gamma}}$. □

We have now managed to show that there exists a set of "nice" vectors $\mathcal{N} = \{v_{\mathcal{L}}\} \subseteq W$. The next claim shows that, if we knew the $v_{\mathcal{L}}$ then we could set $\mathcal{S} = \mathcal{N}$ and the algorithm $\texttt{Recover-Solution}_{\mathcal{N}}(\mathcal{U})$ would return a 99% satisfying assignment for $\mathcal{U}$.

Claim 11 *If v is a vector such that $v = \alpha y^{(\mathcal{L})} + \beta y_{\perp}$ for some $y^{(\mathcal{L})}$ with $\alpha > 0$, then the coordinate $x_{uL(u)}$ is maximum in at least $(1 - \frac{\beta^2}{\alpha^2})n$ blocks. In particular, for every $v_{\mathcal{L}} \in \mathcal{N}$ the coordinate $x_{uL(u)}$ is maximum in at least $(1 - \frac{2\epsilon}{\gamma - 2\epsilon})n$ blocks. By taking, for example $\gamma \geq 202\epsilon$ we get that there is a unique maximum in 99% of the blocks.*

Proof: Within each block u, in order for coordinate $L(u)$ to be no longer the maximum one, it must happen that for some j

$$\alpha \frac{1}{\sqrt{n}} \leq \beta \cdot (y_{\perp})_{uj}$$

However, this gives

$$\|(y_{\perp})_u\| \;\geq\; (y_{\perp})_{uj}^2 \;\geq\; \frac{\alpha^2}{\beta^2 n}$$

Since $\|y_{\perp}\| = 1$, this can only happen for at most $\frac{\beta^2}{\alpha^2} n$ blocks. The statement for $v_{\mathcal{L}}$ follows immediately. □

3.1.1 Finding the Set $\mathcal{S}$ of Test Vectors.

However, since we don't know the $y^{(\mathcal{L})}$ (if we did, we would be done), and the space W contains infinitely many unit vectors, we cannot identify $\mathcal{N}$. To get around this, we *discretize* W with an appropriate epsilon-net. If we let $w^{(0)}, \ldots, w^{(\dim(W)-1)}$ be an eigenbasis for W. We define the set $\mathcal{S}$ as

$$\mathcal{S} \;=\; \left\{ v = \sum_{s=0}^{(\dim(W)-1)} \alpha_s w^{(s)} \;\middle|\; \alpha_s \in \sqrt{\frac{2\epsilon}{\gamma \dim(W)}}\mathbb{Z},\ \|v\| \leq 1 \right\}$$

It can be calculated (see, for instance [FO05]) that the number of points in the set $\mathcal{S}$ is at most $e^{O(\frac{\gamma}{\epsilon}\dim(W))}$.

To conclude the proof of theorem 5, it remains to show that $\mathcal{S}$ has a vector close to a nice vector $v_s \in \mathcal{N}$. By construction, $\mathcal{S}$ contains at least one vector v such that $v = \alpha v_s + \beta v_{s\perp}$ for some s and $|\beta| \leq \sqrt{\frac{2\epsilon}{\gamma}}$. Together with claim 10, this implies that we can also write $v = ay^{(s)} + by_{\perp}^{(s)}$, with $|b| \leq \sqrt{\frac{2\epsilon}{\gamma}} + \sqrt{\frac{2\epsilon}{\gamma}} = 2\sqrt{\frac{2\epsilon}{\gamma}}$. Thus, for this vector v, $\texttt{Recover-Solution}_{\mathcal{S}}(\mathcal{U})$ recovers an assignment which agrees with $y^{(s)}$ in $(1 - O(\frac{2\epsilon}{\gamma-2\epsilon}))$ fraction of the blocks. Hence, the assignment violates constraints on edges that have total weight at most $O(\frac{2\epsilon}{\gamma-2\epsilon})nd + \epsilon nd$. Since the total weight of constraints is $nd/2$, the theorem follows. Note that by choosing $\gamma \geq 200\epsilon$ the assignment we recover satisfies more than 90 percent of the constraints.

Finally, it follows that the algorithm runs in $2^{O(\frac{\gamma}{\epsilon}\dim(W))}$ time.

3.2 Proof of Theorem 6

Let W be the eigenspace of M with eigenvalues $\geq (1 - c\epsilon)d$ for some c to be determined. Let Y denote the span of eigenvectors of $\widetilde{M}$ with eigenvalue at least $(1-\gamma)d$, for some γ to be determined. In particular, Y contains all the *assignment* eigenvectors, namely the characteristic vectors of the (perfectly) satisfying assignments of $\widetilde{\mathcal{U}}$ (since they all have eigenvalue equal to d). We will use theorem 5 together with a bound on the dimension of W. Towards that end, we will apply some results from matrix perturbation theory.

3.2.1 Perturbation of Eigenspaces

To find an appropriate γ such that the eigenspaces W and Y are close, we use the following claim which essentially appears in [DK70] as the $\sin\theta$ theorem and was used in [KT07]. We give the proof for self-containment.

Claim 12 *Let w be a unit length eigenvector of M with eigenvalue $\lambda \geq (1 - c\epsilon)d$, for some $c \geq 2$ and let λ_s denote the largest eigenvalue of $\widetilde{M}$ which is smaller than $(1 - \gamma)d$. Then, w can be written as $\alpha y + \beta y_\perp$ with $|\beta| \leq \left\|(\widetilde{M} - M)w\right\| / (\lambda - \lambda_s)$. Here $y \in Y$ and $y_\perp \in Y^\perp$.*

Proof: We have

$$(\widetilde{M} - M)w \;=\; \alpha\widetilde{M}y + \beta\widetilde{M}y_\perp - \lambda w \;=\; \alpha(\widetilde{M}y - \lambda y) + \beta(\widetilde{M}y_\perp - \lambda y_\perp)$$

Since $(\widetilde{M} - \lambda I)y$ and $(\widetilde{M} - \lambda I)y_\perp$ are in orthogonal eigenspaces, we have

$$\left\|(\widetilde{M} - M)w\right\|^2 = \alpha^2 \left\|(\widetilde{M} - \lambda I)y\right\|^2 + \beta^2 \left\|(\widetilde{M} - \lambda I)y_\perp\right\|^2 \geq \beta^2 \left\|(\widetilde{M} - \lambda I)y_\perp\right\|^2$$

However, $\left\|(\widetilde{M} - \lambda)y_\perp\right\| \geq (\lambda - \lambda_s)$ which proves the claim. $\square$

Hence, to prove that the space Y does not change by much due to the perturbation, we simply need to bound $\left\|(\widetilde{M} - M)w\right\|$. We will need the fact that w is somewhat "uniform" across blocks. To formalize this, let $\bar{w}$ be the n-dimensional vector such that $\bar{w}_u = \|w_u\|$ where w_u is the k-dimensional vector $(w_{u1}, \ldots, w_{uk})^T$. We then show that $\bar{w}$ is very close to a vector in $S_{(1-\gamma)}$.

Claim 13 *If w is an eigenvector of M with eigenvalue more than $(1-c\epsilon)d$ and $\bar{w}$ as above, then we can write $\bar{w}$ as $\bar{w} = a\phi + b\phi_\perp$ with $|b| \leq \sqrt{\frac{c\epsilon}{\gamma}}$ and $\phi \in S_{(1-\gamma)}$, $\phi_\perp \in S^\perp_{(1-\gamma)}$.*

Proof:[Of Claim 13] Since, w corresponds to a large eigenvalue, we have that

$$(1-c\epsilon)d \ \leq \ (w)^T M w \ \leq \ \sum_{u,v} \|w_u\| \, A_{uv} \, \|w_v\| \ = \ (\bar{w})^T A \bar{w}$$

Writing $\bar{w}$ as $a\phi + b\phi_\perp$, we get

$$\begin{aligned} (\bar{w})^T A \bar{w} \ &\leq \ a^2 d + b^2(1-\gamma)d \\ \Rightarrow (1-c\epsilon)d \ &\leq a^2 d + b^2(1-\gamma)d \ \Rightarrow \ |b| \leq \sqrt{\frac{c\epsilon}{\gamma}} \end{aligned}$$

□ Using the above, and the fact that the matrix $\widetilde{M}$ is only perturbed in ϵ fraction of the edges, we can now bound $\left\|(\widetilde{M}-M)w\right\|$ as follows.

Claim 14 $\left\|(\widetilde{M}-M)w\right\| \leq (2C + 2\sqrt{c})\sqrt{\frac{\epsilon}{\gamma}}d$

Proof: Define the $n \times n$ matrix R as $R_{uv} = w_{uv} \leq 1$ when the block $(\widetilde{M}-M)_{uv}$ has any non-zero entries, and $R_{uv} = 0$ otherwise. Note that if $(\widetilde{M}-M)_{uv}$ is non-zero, then it must be the (scaled) difference of two permutation matrices. Thus, for all v we have $\left\|(\widetilde{M}-M)_{uv} w_v\right\| \leq 2R_{uv}\|w_v\|$. We calculate:

$$\begin{aligned} \left\|(\widetilde{M}-M)w\right\| \ = \ \sqrt{\sum_u \left\|\sum_v (\widetilde{M}-M)_{uv} w_v\right\|^2} \ &\leq \ \sqrt{\sum_u \left(\sum_v \left\|(\widetilde{M}-M)_{uv} w_v\right\|\right)^2} \\ &\leq \ \sqrt{\sum_u \left(\sum_v 2R_{uv}\|w_v\|\right)^2} \\ &\leq \ 2\|R\bar{w}\| \end{aligned}$$

To estimate $\|R\bar{w}\|$, we break it up as

$$\|R\bar{w}\| \leq a\|R\cdot\phi\| + b\|R\cdot\phi_\perp\|$$

Since each row of R has total sum of entries at most d, $b\|R\cdot\phi_\perp\| \leq \sqrt{\frac{c\epsilon}{\gamma}}d$. Also,

$$\|R\cdot\phi\| = \sqrt{\sum_u |\sum_v R_{uv}\phi_v|^2} \leq \sqrt{\sum_u \left(\sum_v R_{uv}|\phi_v|\right)^2} \leq \frac{C}{\sqrt{n}}\sqrt{\sum_u \left(\sum_v R_{uv}\right)^2}$$

Since R has a total sum of entries at most ϵnd, this expression is maximized when it has d 1s in ϵn rows. This gives $\|R\cdot\phi\| \leq C\sqrt{\epsilon}d$. And, putting everything together we obtain

$$\left\|(\widetilde{M}-M)w\right\| \ \leq \ 2C\sqrt{\epsilon}d + 2\sqrt{\frac{c\epsilon}{\gamma}}d \ \leq \ (2C + 2\sqrt{c})\sqrt{\frac{\epsilon}{\gamma}}d$$

□

Combining the above bound with claim 12, we get that any unit-length vector $w \in W$ can be expressed as $\alpha y + \beta y_\perp$ where $y \in Y$ and $|\beta| \le (2C + 2\sqrt{c})\sqrt{\frac{\epsilon}{\gamma}d} \cdot \frac{1}{(1-c\epsilon)d - \lambda_s}$. Recall that λ_s was smaller than $(1-\gamma)d$, which implies

$$|\beta| \le (2C + 2\sqrt{c})\sqrt{\frac{\epsilon}{\gamma}} \cdot \frac{1}{\gamma - c\epsilon} \tag{1}$$

We will again use the set $\mathcal{S}$ as it appears in the proof of the Main theorem 5. In order to bound the number of points in $\mathcal{S}$, we show in the next couple of claims that $\dim(W)$ is not too large.

The next claim easily follows from the fact that $\widetilde{M}$ is the adjacency matrix of k disconnected copies of G.

Claim 15 *Let Φ be an eigenbasis for G. For every eigenvector $\phi = (\phi_u)_{u\in V} \in \Phi$ define $\widetilde{\phi}$ to be the kn-dimensional vector $\widetilde{\phi}_{ui} = \phi_u$, for $i = 1, \cdots, k$ and let $\mathcal{E}(\phi) = \{\widetilde{\phi} \cdot y^{(i)} | \quad \text{for} \quad i = 0, \cdots, k-1\}$. Here $\cdot$ denotes entry-wise vector product. Then $\bigcup_{\phi\in\Phi} \mathcal{E}(\phi)$ is an eigenbasis of $\widetilde{M}$.*

Note that this claim also shows that the dimension of Y is at most $k \cdot D_S$. Now, from equation 1, by choosing an appropriate γ, we have that the subspace W is very close to Y.

Claim 16 *Any unit-length vector $w \in W$ can be expressed as $\alpha y + \beta y_\perp$ where $y \in Y$ and $|\beta| \le 4(C + \sqrt{c})\sqrt{\frac{\epsilon}{\gamma^3}}$. And, by taking $\gamma \ge 8(C + \sqrt{c})\epsilon^{1/3}$ we get $|\beta| \le \frac{1}{8}$.*

Proof: From equation 1 we have

$$|\beta| \le (2C + 2\sqrt{c})\sqrt{\frac{\epsilon}{\gamma}} \cdot \frac{1}{\gamma - c\epsilon} \le 4(C + \sqrt{c})\sqrt{\frac{\epsilon}{\gamma^3}}$$

The claim follows from substituting the value for γ and from the fact that $C, c \ge 1$. □

Combining claims 15 and 16, we can argue that W has dimension at most $k \cdot D_S$: otherwise, we would find a vector orthogonal to all the vectors in Y which cannot be close to their span.
To conclude the proof of theorem 6, we apply theorem 5 with some appropriate choice of constant c.

Finally, it remains to argue about the running time of the algorithm. Since $\dim(W) = kD_S$, the number of points is exponential in kD_S. Hence, the algorithm runs in $2^{O(k\cdot D_S)}$ time.

4 Solving Unique Games on the Khot-Vishnoi Graph

In this section, we present a quasi-polynomial time algorithm that, when run on a $(1-\epsilon)$ satisfiable instance of Unique Games on the graph $\mathcal{KV}_{n,\epsilon}$, with alphabet size n, finds a 90 percent satisfying assignment. We use the equivalent definition of the graph $\mathcal{KV}_{n,\epsilon}$ as a quotient H_n/H of the n-dimensional hypercube $H_n = \{0,1\}^n$ by the Hadamard code H. We will need the following claim for the spectrum of $\mathcal{KV}$.

Claim 17 *The following is true for the spectrum of $\mathcal{KV}_{n,\epsilon}$.*

- *The eigenvectors are the characters of the group H_n / H. They can be "thought of" as characters χ_ω of H_n with $\omega \in H^\perp$ or, equivalently, seen as n-dimensional boolean vectors, $\omega \cdot h = 0 \quad \forall h \in H$. (here $\cdot$ denotes vector inner product over $\mathbf{F}_2$).*
- *The eigenvalue that corresponds to χ_ω is $(1-2\epsilon)^r n$, where $r = |\omega|$ is the hamming weight of ω, and appears with multiplicity C_r. Here C_r denotes the number of elements in the Hamming code of weight r([Kim]). Moreover, $C_r \leq \binom{n}{r}$.*

PROOF: For item (1) above, we observe that χ_ω are precisely the $m = \frac{2^n}{n}$ characters of H_n that are constant on every coset of H. This is equivalent to

$$\chi_\omega(x) = \chi_\omega(x+h), \quad \forall h \in H \Leftrightarrow (-1)^{\omega \cdot x} = (-1)^{\omega \cdot (x+h)} \Leftrightarrow (-1)^{\omega \cdot h} = 1 \Leftrightarrow \omega \cdot h = 0$$

For item (2), we just need to calculate the eigenvalues corresponding to χ_ω or, equivalently, the quadratic form

$$\begin{aligned} & \frac{1}{\sqrt{m}}(\chi_\omega)^T A_{\mathcal{KV}}(\frac{1}{\sqrt{m}}\chi_\omega) = \frac{1}{m}\sum_{x,y} \chi_\omega(x) A_{\mathcal{KV}}(x,y)\chi_\omega(y) \\ = & \frac{n}{2^n} \sum_{x,y \in Q} \sum_{h_1,h_2 \in H} \chi_\omega(x) \epsilon^{|x+h_1+y+h_2|}(1-\epsilon)^{|n-(x+h_1+y+h_2)|}\chi_\omega(y) \\ = & \frac{n}{2^n} \sum_{x,y \in H_n} \chi_\omega(x+y)\epsilon^{|x+y|}(1-\epsilon)^{|n-(x+y)|} = \frac{n}{2^n} 2^n \sum_{z \in H_n} \chi_\omega(z)\epsilon^{|z|}(1-\epsilon)^{|n-z|} \\ = & n\mathbf{E}_{z \in_\epsilon \mathcal{F}}\left[\chi_\omega(z)\right] = n(1-2\epsilon)^{|\omega|} \end{aligned}$$

To conclude the proof, we need to argue about the multiplicity of each eigenvalue or, equivalently, the number of characters χ_ω of a given hamming weight that are orthogonal to the Hadamard code. This is exactly the number of elements of the Hamming code of weight $r = |\omega|$, and their cardinality C_r can be computed by a recursive formula (see, e.g [Kim]). The fact that $C_r \leq \binom{n}{r}$ easily follows from the fact that there can be at most $\binom{n}{r}$ characters of H_n that correspond to hamming weight r. $\square$

We will directly apply corollary 8 of theorem 5. For this purpose we need to calculate the dimension of the eigenspace $S_{(1-\gamma)}$ of $\mathcal{KV}$ for some appropriate $\gamma \geq 8(1+\sqrt{c})\epsilon^{1/3} \geq 8(1+10)\epsilon^{1/3} = 88\epsilon^{1/3}$. Thus it suffices to take $\gamma = 88\epsilon^{1/3}$.

Lemma 18 *Let $\gamma = 88\epsilon^{1/3}$. The dimension of the eigenspace $S_{(1-\gamma)}$ of the graph $\mathcal{KV}$ with eigenvalues $\geq (1-\gamma)$ is at most $D_S \leq \frac{1}{\epsilon} n^{\frac{1}{\epsilon}} = poly(n) = poly(\log N)$.*

PROOF: We first need to identify an upperbound on r such that $(1-2\epsilon)^r \leq (1-\gamma)$, for $\gamma = 88\epsilon^{1/3}$. It is enough to take $r = \frac{\log \frac{1}{1-88\epsilon^{1/3}}}{\log \frac{1}{1-2\epsilon}}$ which, for ϵ small enough, is $\leq \frac{1}{\epsilon}$. The dimension of $S_{(1-\gamma)}$ can now be bounded, using claim 17, as follows: $D_S \leq \sum_{r=0}^{1/\epsilon} \binom{n}{r} \leq \frac{1}{\epsilon} n^{\frac{1}{\epsilon}}$ $\square$

We conclude by arguing, according to corrolary 8, that the running time of our algorithm on the graph $\mathcal{KV}_{n,\epsilon}$ is bounded by

$$2^{nD_S} = 2^{n\frac{1}{\epsilon}\cdot n^{\frac{1}{\epsilon}}} = 2N^{\frac{1}{\epsilon}\cdot(\log N)^{\frac{1}{\epsilon}}} = N^{(\text{poly}(\log N))}$$

4.0.2 Sub-exponential time Algorithm for the Hypercube

We also note that similarly, we can argue that the algorithm of corollary 8 implies a sub-exponential time algorithm when the constraint graph is the Hypercube H_n.

Claim 19 *There is an algorithm that, on input a $(1-\epsilon)$ satisfiable instance of Unique Games on the graph H_n, runs in time $2^{O(N^{\epsilon^{1/3}})}$ and outputs a 90% satisfying assignment. Here $N = 2^n$ the number of nodes of H_n.*

Proof: We again apply immediately corollary 8. Again, it suffices to take $\gamma = 88\epsilon^{1/3}$. It is well known that eigenvectors of H_n are the characters χ_r with corresponding eigenvalues $(1 - \frac{2r}{n})n$. The eigenvalues appear with multiplicity $\binom{n}{r}$. It suffices to take $S_{(1-\gamma)}$ to be the eigenspace of eigenvalues greater than $(1 - \frac{2r}{n})n$ for $r = 44\epsilon^{1/3}n$. The dimension of D_S is at most $\sum_{i=0}^{r} \binom{n}{i}$. By Stirling's formula we get

$$\sum_{i=0}^{r} \binom{n}{i} \leq 2^{O(\epsilon^{1/3}n)}$$

The running time of the algorithm can now be calculated to be at most $2^{kD_S} \leq 2^{O(N^{\epsilon^{1/3}})}$.
□

Acknowledgements

The author would like to thank Madhur Tulsiani for his contribution to the ideas and techniques for developing spectral algorithms for Unique Games.

References

[AIMS09] Sanjeev Arora, Russell Impagliazzo, William Matthews, and David Steurer. Improved algorithms for unique games via divide and conquer, 2009. manuscript.

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth Vishnoi. Unique Games on Expanding Constraint Graphs are Easy. In *Proceedings of the Annual ACM Symposium On The Theory Of Computing*, 2008.

[Aus07] Per Austrin. Towards sharp inapproximability for any 2-csp. In *IEEE Symposium on Foundations of Computer Science (FOCS)*, 2007.

[CKK+06] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. In *Computational Complexity*, volume 15(2), pages 94–114, 2006.

[CMM06a] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Near-optimal algorithms for unique games. In *Proceedings of the ACM Symposium on the Theory of Computing*, volume 38, pages 205–214, 2006.

[CMM06b] Eden Chlamtac, Konstantin Makarychev, and Yury Makarychev. How to play unique games using embeddings. In *Proceedings of the IEEE Annual Symposium on Foundations of Computer Science*, volume 48, pages 687–696, 2006.

[DK70] Chandler Davis and W.M. Kahan. The Rotation of Eigenvectors by a Perturbation. III. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970.

[FO05] Uriel Feige and Eran Ofek. Spectral techniques applied to sparse random graphs. *Random Structures and Algorithms*, 27(2):251–275, 2005.

[GT06] Anupam Gupta and Kunal Talwar. Approximating unique games. In *Proceedings of the eventeenth Annual ACM-SIAM Symposium on Discrete Algorithms*, volume 24, pages 99–106, 2006.

[Kas02] Petteri Kaski. Eigenvectors and Spectra of Cayley Graphs, 2002.

[Kho02] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the ACM Symposium on the Theory of Computing*, volume 34, pages 767–775, 2002.

[Kim] Dae San Kim. Weight Distributions of Hamming Codes. Available from http://arxiv.org/PScache/arxiv/pdf/0710/0710.1467v1.pdf.

[KKMO04] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable CSPs? In *Proceedings of the Annual Symposium on Foundations of Computer Science*, volume 45, pages 146–154, 2004.

[KR03] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. In *Proceedings of the Annual IEEE Conference on Computational Complexity*, volume 18, pages 379–386, 2003.

[KT07] Alexandra Kolla and Madhur Tulsiani. Playing Unique Games Using Graph Spectra, 2007. Manuscript.

[KV05] Subhash Khot and Nisheeth K. Vishnoi. The unique games conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, volume 46, pages 53–62, 2005.

[MM09] Yury Makarychev and Konstantin Makarychev. How to Play Unique Games on Expanders, 2009. ECCC Report No. 21.

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the Annual ACM Symposium On The Theory Of Computing*, 2008.

[RS09] Prasad Raghavendra and David Steurer. Integrality Gaps for Strong SDP Relaxations of Unique Games. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 2009.

[Spi] Daniel Spielman. Cayley graphs. Letcure Notes.

[Tre05] Luca Trevisan. Approximation algorithms for unique games. In *Proceedings of the Annual Symposium on Foundations of Computer Science*, volume 46, pages 197–205, 2005.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de