CHICAGO JOURNAL OF THEORETICAL COMPUTER SCIENCE 2015, Article 1, pages 1–20
http://cjtcs.cs.uchicago.edu/

Unique Games on the Hypercube

Naman Agarwal* Guy Kindler† Alexandra Kolla‡ Luca Trevisan§

Received September 13, 2014; Revised February 5, 2015; Published February 7, 2015

Abstract: In this paper, we investigate the validity of the Unique Games Conjecture when the constraint graph is the boolean hypercube. We construct an almost optimal integrality gap instance on the Hypercube for the Goemans-Williamson semidefinite program (SDP) for Max-2-LIN($\mathbb{Z}_2$). We conjecture that adding triangle inequalities to the SDP provides a polynomial time algorithm to solve Unique Games on the hypercube.

1 Introduction

The *Unique Games Conjecture* (UGC) was formulated by Khot in 2002 [16], and has since been the focus of great attention. Deciding it either way would have significant implications: Proving the conjecture would give tight bounds to the approximability of several fundamental optimization problems, including Vertex Cover [19], Max-Cut [17] and non-uniform Sparsest-Cut [9, 20]). Refuting the conjecture would yield an approximation algorithm for finding small non-expanding sets [26], and the techniques used in a refutation would be likely to find applications to other graph partitioning problems like Max-Cut and Sparsest-Cut.

Conjecture 1.1. *(UGC) For any constants $\varepsilon, \delta > 0$, there is a $k = k(\varepsilon, \delta)$ such that it is NP-hard to distinguish between instances of Unique Games with alphabet size k where at least $1 - \varepsilon$ fraction of constraints are satisfiable and those where at most δ fraction of constraints are satisfiable.*

*Supported through funding from the Dept. of Computer Science at the University of Illinois. Part of this work was done while the author was visiting the Simons Institute for Computing at Berkeley.

†Supported by Israeli Science Fund grant no. 1692/13, and Binational Science Foundation grant no. 2012220.

‡Supported through funding from the Dept. of Computer Science at the University of Illinois. Part of this work was done while the author was visiting the Simons Institute for Computing at Berkeley.

§This material is based upon work supported by the National Science Foundation under Grant No.1216642 and by the US-Israel Binational Science Foundation under grant 2010451

Key words and phrases: unique games; semi-definite programming; integrality gap; hypercube

DOI: 10.4086/cjtcs.2015.001

A full definition of a Unique Games instance appears in Section 2, but for now one may think of an instance with alphabet size k as a system of constraints where variables take values in $\mathbb{Z}_k$, and each constraint is a linear equation $\bmod k$ involving two variables. The *constraint graph* of such an instance is a graph that has a vertex for every variable and an edge for every pair of variables that appear together in one of the constraints.

In recent years the UGC was found to be intimately connected to the power of semi-definite programming (SDP). One can trace the first instance of this connection to the seminal paper by Goemans and Williamson [12] on the Max-Cut problem (an instance of Max-Cut can be thought of as a linear equation system over $\mathbb{Z}_2$, and thus it is a Unique Games Instance for alphabet size 2). They gave an SDP based algorithm for MaxCut which, on inputs where the maximal cut is of size $(1-\varepsilon)$, produces a cut that satisfies at least $(1-(2/\pi)\sqrt{\varepsilon})$ fraction of the constraints. A matching integrality gap was found by [15] and [11], and in [17] it was proven that if the UGC is correct, than the Goemans-Williamson algorithm is the polynomial-time approximation algorithm with the best possible approximation ratio.

Theorem 1.2. *Assume the Unique Games Conjecture. Then for all sufficiently small $\varepsilon > 0$, it is NP-hard to distinguish instances of Max-2-LIN$(\mathbb{Z}_2)$ that are at least $(1-\varepsilon)$-satisfiable from instances that are at most $(1-(2/\pi)\sqrt{\varepsilon})$-satisfiable.*

The SDP based approximation algorithm of Goemans and Williamson was later extended to work for general Unique Games by [7, 8], and in [17] it was shown that unless the UGC is false, the approximation ratio achieved by [8] is tight. Raghavendra [24] proved further that for *every* constraint satisfaction problem there is a polynomial-time SDP based algorithm which, if UGC is true, achieves the best possible approximation for the problem.

Currently there is limited evidence either in favor or against the Unique Games conjecture. So far, the most successful algorithmic techniques to approximate unique games (and the ones that may lead to a refutation of the conjecture, if it is indeed false) are based on spectral methods and on semidefinite programming.

Spectral techniques for UG. Spectral techniques have been used to develop good polynomial-time or quasi-polynomial-time algorithms for large classes of instances, including expanders [3, 22], local expanders [2, 26], and graphs with few large eigenvalues [21]. Arora, Barak and Steurer [1] used spectral techniques to develop a sub-exponential ($2^{n^{\Omega(1)}}$-time) algorithm for general instances.

Currently, we do not know how to apply spectral techniques to get good polynomial time or quasi-polynomial time approximations to unique games instances whose constraint graph has several small Laplacian eigenvalues (high threshold rank). In particular, all known spectral techniques fail on instances whose constraint graph has high threshold rank. A simple such instance is the hypercube. Thus, understanding how to design algorithms for the hypercube constraint graph is a very natural next step which will likely shed light on what techniques we need to use in order to tackle the UGC .

Semidefinite programming techniques for UG. Several semidefinite programming relaxations and rounding schemes have been studied that provide non-trivial approximation [16, 28, 14, 7, 10, 6] in general instances of Unique Games, and match the performance of known spectral algorithms in special classes of instances. Integrality gap instances are known for basic relaxations [20, 25, 5, 18], showing that such relaxations cannot yield a refutation of the unique games conjecture, but Barak and others [4] show that a polynomial-time solvable relaxation based on the Lasserre hierarchy provides good approximations to all known such integrality gap instances. Given our current knowledge, it might be possible that the polynomial time relaxations studied in [4] are sufficient to refute the unique games conjecture.

It would be a very interesting result to show that semidefinite programming relaxations such as the ones studied in [4] solve unique games instances whose constraint graph is the hypercube[1] because that is a paradigmatic hard case for spectral methods.

In this paper we show that the most basic relaxation is not sufficient for this goal, and we present a family of integrality gap instances over a boolean range such that the value of the Goemans-Williamson SDP is $1-\varepsilon$ while the true optimum is $1-O(\varepsilon^{1.5})$.

Unique Games vs. Small Set Expansion. In general graphs, it often may appear that the Small Set Expansion Problem (SSE)[27] and the Unique Games Problem have roughly the same complexity. Namely, when we look at algorithms for unique games on general graphs, they have a simple counterpart algorithm for SSE, which uses all the same main ideas. However, for the hypercube graph, its Small Set Expansion can be efficiently (an very easily) determined, whereas we don't know of an algorithm that efficiently solves Unique Games on it.

1.1 Our results

In this paper, we consider the Max-2-LIN($\mathbb{Z}_2$) problem. Recall that by theorem 1.2, an improved approximation algorithm for Max-2-LIN($\mathbb{Z}_2$) for general instances would refute the UGC. We study the approximability of Max-2-LIN($\mathbb{Z}_2$) when restricted to instances whose constraint graph is a hypercube.

We construct a family of integrality gap instances of Unique Games on the hypercube constraint graph for the Goemans-Williamson semi-definite program (SDP). In the following, we refer to an edge of the constraint graph of a Max-2-LIN($\mathbb{Z}_2$) instance as an *equality* or an *inequality* edge, depending on whether the constraint on the edge is satisfied when its endpoints have the same or opposite value, respectively.

An interesting contribution of our result. Our construction of integrality gap instances differs from other such constructions in an important aspect. Usually, if one wants to construct a family of instances for a graph maximization problem whose optimum has value ρ_{opt} but whose SDP has value$\rho_{SDP} >> \rho_{opt}$, one constructs a symmetric instance and an SDP solution such that every edge contributes ρ_{SDP} to the value of the solution. In the case of a unique game on the hypercube, however, if we have an instance and an SDP solution such that each edge contributes, say, .9, then the instance must be satisfiable. So to obtain a gap, one needs to construct a somewhat asymmetric instance, where in the SDP solution some edges contribute more to the objective function and some edges contribute less. Asymmetric instances and solution could lead to better integrality gaps in other settings.

[1] Here by "solve" we mean a good approximation such as, for example, the ability to solve a $1-O(\varepsilon)$ fraction of constraints in a $(1-\varepsilon)$-satisfiable boolean instance, or 10% of constraints in a 90%-satisfiable instance over an arbitrary alphabet.

The GW algorithm. As mentioned above, the first non-trivial approximation algorithm for Unique-Games was the SDP based algorithm of Goemans and Williamson [13], and it was later generalized by Charikar-Makarychev-Makarychev [7] to give the best known (worst-case) polynomial time approximation for Unique Games instances. When the Goemans-Williamson algorithm (GW algorithm for short) is applied to a $(1-\varepsilon)$-satisfiable instance of Max-2-LIN$(\mathbb{Z}_2)$, it finds an assignment satisfying at least $1-O(\sqrt{\varepsilon})$ fraction of the constraints. Assuming the UGC, theorem 1.2 implies that this performance is best possible by any algorithm. As also mentioned above, tightly matching integrality gaps were shown in [11] and [15], and in [11] instances were constructed where any reasonable rounding procedure for the GW SDP gives an approximation factor matching the above parameters.

In the context of solving UG on the hypercube, it is natural to ask how well the GW SDP performs. As far as the authors are aware, until this work it was not ruled out that the GW algorithm performs (almost) perfectly, which would imply that instances with a hypercube constraint graph are easy.

Integrality gap on the hypercube. In this paper we prove that the GW SDP has an integrality gap on the hypercube with a behaviour similar to the gap on general graphs, albeit with a different power of ε.

Theorem. *(**Main**) For every sufficiently small constant ε, and for every $d \geq d(\varepsilon)$, there exists a Max-2-LIN$(\mathbb{Z}_2)$ instance on the boolean cube Q_d of dimension d such that the UG combinatorial optimal value for that instance is $1-\Omega(\varepsilon)$, and the GW SDP optimal value is $1-O(\varepsilon^{3/2})$.*

We believe our integrality gap can be extended for the case of more than 2 labels, but have no proof of that at this point. We note that our result is especially interesting since all previously known integrality gap instances for the GW SDP as well as most integrality gap instances for other various SDPs, are known to not be "hard" instances for Unique Games and can be approximately solved using spectral techniques in time at most $2^{n^{o(1)}}$, where n is the number of nodes of the graph. The hypercube graph is unique, in the sense that known state-of-the-art spectral algorithms cannot solve it in time faster than $2^{n^{\Omega(1)}}$, (where n is the number of nodes of the hypercube) and, as shown by this work, there is also an integrality gap instance on the hypercube for the GW SDP.

Adding triangle inequalities. One can increase the power of the GW SDP by adding so called "triangle inequality constraints" – this is a standard manipulation of semidefinite programs, and is a subset of the constraints added by the second Lassere hierarchy. We show that adding these constraints to the GW SDP breaks the integrality gap of our instance. We conjecture that in fact the GW SDP with triangle inequalities solves the Unique Games problem on the hypercube – whether this is indeed the case is a very interesting question.

Our Techniques. We construct our gap instance as follows: We start with an instance for which all edges are equality edges, for which the all-one's assignment is the (combinatorial) optimum assignment with value 1 (since the instance is completely satisfiable). We then convert a small number edges to inequalities ensuring that the all-one's assignment is still roughly the combinatorial optimum assignment. At the same time, the SDP optimum value becomes larger than the fraction of constraints satisfied by the optimum combinatorial assignment. More concretely we show the following lemma :

Lemma 3.3 (Main Lemma)**.** *For every sufficiently large d, there exists a Max-2-LIN*$(\mathbb{Z}_2)$ *Instance on the hypercube* Q_d *of dimension d, such that the combinatorial optimum is* $1-\Omega(d^{-1/2})$ *and the SDP optimum is* $1-\mathcal{O}(d^{-3/4})$.

In the following, we refer to edges of the hypercube connecting vertices that differ in the i-th coordinate as edges *going in the i-th direction*.

To prove the above lemma we define a gap instance $\Delta(k,d)$ on Q_d, where we start from the all-equalities instance on Q_d and introduce inequalities along k directions, for $k \sim \sqrt{d}$. Our goal in choosing which edges to designate as inequality edges is to keep the solutions in which all variables are assigned the same value (say, the value one) to be close to optimal, which implies that the combinatorial optimum is roughly one minus the fraction of inequality edges. At the same time, we want the SDP solution to have value noticeably higher than one minus the fraction of inequalities, thus creating a gap.

We show that if we restrict ourselves to introducing inequality edges in just one direction, then we can show that up to about half the edges going in that direction can be changed to inequality while preserving the property that the all-one assignment to the variables is optimal, and while allowing an SDP solution of higher optimum value.

Next, we further extend the construction by placing these inequality regions along $\mathcal{O}(\sqrt{d})$ number of directions. The parameter $\mathcal{O}(\sqrt{d})$ is chosen such that the all ones assignment is still nearly the optimum. In particular in Lemmas 3.5 and 3.6 we prove that if we place these inequality regions in k directions, the combinatorial optimum grows linearly with k (in particular $\mathcal{O}(\frac{k}{d})$) whereas the SDP optimum grows at most proportionally to $\sqrt{k}$ (in particular $\mathcal{O}(\frac{\sqrt{k}}{d})$). Setting $k \sim \mathcal{O}(\sqrt{d})$ we get a non trivial (super-constant) gap. We formally state and prove the above idea in lemma 3.3.

So far, we have managed to show that a non trivial gap instance exists for sub-constant $\varepsilon \sim d^{-1/2}$. The next task is to blow this instance up to create gap instances for constant ε. We do this by showing the following gap preservation lemma:

Lemma 3.4 (Gap Preservation)**.** *Suppose that I is a Max-2-LIN*$(\mathbb{Z}_2)$ *instance define over the d dimensional hypercube, and let* α *be the combinatorial value of I and* β *be the optimal SDP value for it. Then for every i there exists an instance* I' *defined over the* $d \cdot i$ *dimensional hypercube whose combinatorial value is at least* α *and whose GW SDP optimal value is at most* β.

We prove the above lemma by defining a *tensor* product operation on Max-2-LIN$(\mathbb{Z}_2)$ instances on the cube which allows us to create larger instances of the cube preserving the gap of the original instance.

1.2 Organization

The rest of the paper is organized as follows. Section 2 contains some definitions and notation that we will be using throughout the paper. Sections 3 and 4 contain the description of the integrality gap instance on the hypercube and the proof of our Main theorem. In section 5 we discuss adding triangle inequalities to the GW algorithm, show that our instance no longer gives an integrality gap for this strengthened SDP, and conjecture that actually this SDP solves all instances on the hypercube.

2 Preliminaries

2.1 Notations

We use the following notations throughout the paper. Q_d refers to a hypercube graph (V_d, E_d) of dimension d, with vertex set V_d and edge set E_d. We generally reserve d for the dimension of the hypercube in context. For every vertex $v \in V_d$ of the hypercube we naturally associate vector $\mathbf{v} \in \{0,1\}^d$ (we denote vectors in boldface letters). Let $\mathbf{v}_i$ be the i^{th} coordinate of the vector $\mathbf{v}$. We denote by $H(\mathbf{v})$ the hamming weight of the vector $\mathbf{v}$, i.e. the number of $1's$ in $\mathbf{v}$.

Let $Q_{d-k}(\mathbf{x})$ where $\mathbf{x} \in \{0,1\}^k$ be a $d-k$ dimensional sub-cube of of Q_d obtained by fixing the first k coordinates to be $\mathbf{x}$.

Given any two vectors we define their tensor product as follows

Definition 2.1 (Tensor Product)**.** Given two vectors $\mathbf{x} \in \mathbb{R}^{n_1}, \mathbf{y} \in \mathbb{R}^{n_2}$ define the vector $\mathbf{x} \otimes \mathbf{y} \in \{0,1\}^{n_1 * n_2}$ as follows

$$\mathbf{x} \otimes \mathbf{y}_{(i*n_1+j)} = \mathbf{x}_i \mathbf{y}_j$$

It will well known that the norm of tensor products is multiplicative,

- $\|\mathbf{x} \otimes \mathbf{y}\|^2 = \|x\|^2 \|y\|^2$,

where $\|\mathbf{x}\|$ denotes the l_2-norm of $\mathbf{x}$.

2.2 Unique Games and Max-2-LIN$(\mathbb{Z}_2)$ Definitions

Following is a formal genera definition of the Unique Games problem.

Definition 2.2 (Unique Games)**.** A *Unique Games* instance for alphabet size k is specified by an undirected 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 (u, v) s.t. $\{u, v\} \in E$, with $\pi_{uv} = (\pi_{vu})^{-1}$. An assignment of values in $[k]$ to the variables is said to satisfy the constraint on the edge $\{u, v\}$ if $\pi_{uv}(x_u) = x_v$. The optimization problem is to assign a value in $[k]$ to each variable x_u so as to maximize the number of satisfied constraints. We define the *value* of a solution to be the fraction of constraints that are *not* satisfied by this solution.

An optimal solution for a Unique Games instance which satisfies the maximum number of constraints will also be referred to as the *combinatorial solution* and its value will be referred to as the *combinatorial value*. We note that, while it is slightly more common to define the value of a solution to be the fraction of satisfied constraints by the solution, in this paper we find it more convenient to define the value of a solution as the fraction of constraints that is not satisfied by it.

Definition 2.3 (Max-2-Lin)**.** A Max-2-LIN$(\mathbb{Z}_2)$ instance is a Unique Games instance with alphabet size 2. Note that to specify a Max-2-LIN$(\mathbb{Z}_2)$ instance it is sufficient to specify a graph $G = (V, E)$ and a function $f : E \to \{0,1\}$. We call the edges with the value 0 “equality” edges and the edges with value 1 "inequality" edges in accordance to the constraints implying whether the two labels on the edge should be equal or not.

Since the main focus of this paper are instances where the constraint graphs are Hypercubes, we will be viewing a Max-2-LIN($\mathbb{Z}_2$) instance as a function $I : E_d \to \{0,1\}$.

2.3 Goemans Williamson SDP and Gap Instances

We next describe the Goemans-Williamson semi-definite program [13] for solving Max-2-LIN($\mathbb{Z}_2$) . Note that the original paper by Goemans Williamson defines the SDP for the Max-Cut problem which is essentially a Max-2-LIN($\mathbb{Z}_2$) problem with all edges being inequality edges.

Definition 2.4 (GW SDP)**.** Given a graph $G = (V,E)$ and a Max-2-LIN($\mathbb{Z}_2$) instance $I : E \to \{0,1\}$ on it, let the set of equality edges be E^+ and the set of inequality edges be E^-. The Goemans-Williamson SDP for the instance is defined as

$$\text{minimize } \frac{1}{4|E|}\left(\sum_{(u,v)\in E^+} \|x_u - x_v\|^2 + \sum_{(u,v)\in E^-} \|x_u + x_v\|^2\right)$$

$$\text{subject to } \quad \|x_u\|^2 = 1 \;\; (\forall\, u \in V)$$

Note that the analysis carried out by Goemans-Williamson for Max-Cut essentially holds for Max-2-LIN($\mathbb{Z}_2$) too. In particular given a Max-2-LIN($\mathbb{Z}_2$) instance with combinatorial value ε the above SDP has optimum value $\Omega(\varepsilon^2)$. As mentioned in the introduction, this is tight under the Unique Games Conjecture.

3 Main Theorem

In this section, we describe an instance of Max-2-LIN($\mathbb{Z}_2$) on the Hypercube which is an integrality gap for the GW SDP.

Definition 3.1 ((α,β)-gap instance)**.** An infinite family $\mathcal{F}$ of Max-2-LIN($\mathbb{Z}_2$) instances on the hypercube (of varying dimensions) is called *an* (α,β)*-gap Instance* for Max-2-LIN($\mathbb{Z}_2$) if the combinatorial optimum on any $I \in \mathcal{F}$ is $\Omega(\alpha)$ and the GW SDP has optimum value $\mathcal{O}(\beta)$.

Note that the existence of an (α,β)-gap instance for Max-2-LIN($\mathbb{Z}_2$) proves that the GW SDP has an integrality gap of at least $\Omega(\frac{\alpha}{\beta})$. The following theorem therefore establishes an integrality gap for the GW SDP on the hypercube.

Theorem 3.2. *(Main) For every sufficiently small constant* ε *there is an* $(\varepsilon,\varepsilon^{3/2})$ *Instance for Max-2-LIN*($\mathbb{Z}_2$) .

Theorem 3.2 is obtained from the following two lemmas, which are proven below.

Lemma 3.3. *(Main Lemma) For every sufficiently large d, there exists a Max-2-LIN($\mathbb{Z}_2$) instance defined over the d dimensional hypercube, whose combinatorial value is $\Omega(d^{-1/2})$ but for which the GW SDP optimal value is $O(d^{-3/4})$.*

Lemma 3.4 (Gap Preservation)**.** *Suppose that I is a Max-2-LIN($\mathbb{Z}_2$) instance define over the d dimensional hypercube, and let α be the combinatorial value of I and β be the optimal GW SDP value for it. Then for every i there exists an instance I' defined over the $d \cdot i$ dimensional hypercube whose combinatorial value is at least α and whose GW SDP optimal value is at most β.*

Proof of Theorem 3.2. Let $\varepsilon > 0$ be small enough, and take $d = 1/\varepsilon^2$. By lemma 3.3 we can find an instance I whose combinaorial value is at least $\Omega(d^{-1/2}) \sim \varepsilon$ and whose GW SDP optimal value if $\Omega(d^{-3/4}) \sim \varepsilon^{3/2}$. Considering the family of instances that can be obtained from I by applying Lemma 3.4 we obtain a $(\varepsilon, \varepsilon^{3/2})$ Instance of Max-2-LIN($\mathbb{Z}_2$) , proving the theorem. $\square$

In the rest of this section we prove Lemma 3.3. Lemma 3.4 is proven in the next section.

3.1 Proof of Lemma 3.3

We need to construct a Max-2-LIN($\mathbb{Z}_2$) instance over the hypercube Q_s of dimension d. Let E_d be the set of edges of the hypercube, and let k be a parameter to be fixed later. We define the instance $\Delta[k,d] : E_d \to \{0,1\}$ as follows. For any edge $e = (v_1, v_2)$ let $i(e)$ be the coordinate along which the corresponding vectors $\mathbf{v_1}, \mathbf{v_2}$ differ. Let $H(\mathbf{v}[k])$ be the hamming weight of the vector $\mathbf{v}$ restricted to only coordinates other than the first k coordinates.

- If $i(e) > k$, $\Delta[k,d](e) = 0$.
- if $i(e) \leq k$ and if $H(\mathbf{v_1}[k]) > \frac{d-k}{2}$, $\Delta[k,d](e) = 0$.
- Otherwise $\Delta[k,d](e) = 1$.

We now make some observations about our instance. Note that all edges that are assigned to 1 (i.e. are inequality edges) are between vertices (v, v') that differ in one of the first k coordinates. Therefore for any subcube $Q_{d-k}(\mathbf{x})$ defined by fixing the first k coordinates to be $\mathbf{x}$, we have that the edges inside $Q_{d-k}(\mathbf{x})$ are all set to 0 (i.e. are equality edges). Consider two vectors $\mathbf{x_1}, \mathbf{x_2} \in \{0,1\}^k$ which differ in one coordinate. Every vertex v in the subcube $Q_{d-k}(\mathbf{x_1})$ is connected by an edge to another vertex v' in $Q_{d-k}(\mathbf{x_2})$. The vertex v' can be thought of as a copy of v in the subcube $Q_{d-k}(\mathbf{x_2})$ (restricted to the last $d-k$ coordinates the two vertices are the same). The edge connecting (v, v') is an inequality or equality edge depending on which side of the majority cut v belongs to in its corresponding subcube. Namely, if more than half of the last $d-k$ coordinates of v are 1, then (v, v') is an equality edge, otherwise it is an inequality edge.

We now bound the GW-SDP optimum and the combinatorial optimum of $\Delta[k,d]$ in the following two lemmas.

Lemma 3.5. *For $k \leq O(\sqrt{d})$, $\Delta[k,d]$ has combinatorial optimum $\Omega(\frac{k}{d})$.*

Lemma 3.6. *$\Delta[k,d]$ has GW-SDP optimum $O(\frac{\sqrt{k}}{d})$.*

Lemma 3.3 easily follows from Lemma 3.5 and Lemma 3.6 by setting $k = c\sqrt{d}$. It is thus left to prove the two lemmas above.

3.2 Proof of Lemma 3.6

To prove the lemma it is enough to exhibit a valid solution to the GW-SDP which achieves a value of $O(\frac{\sqrt{k}}{d})$. To this end we exhibit a two dimensional solution $S : V_d \to \mathbb{R}^2$. Our solution will map every vertex v to a unit vector in $\mathbb{R}^2$ and therefore it is enough to just specify the angles α_v between v and the x-axis.

The solution S is symmetric with respect to the $d-k$ dimensional subcubes $Q_{d-k}(\mathbf{x})$ and depends only upon the parity of the k dimensional vector $\mathbf{x}$. Within a subcube, the vector assigned to a vertex depends only on the hamming weight of the vertex restricted to the subcube. Let $L_i(\mathbf{x})$ be the layer in the $d-k$ dimensional subcube $Q_{d-k}(\mathbf{x})$ of hamming weight i (vertices with i ones in the last $d-k$ coordinates). Formally a vertex $v \in L_i(\mathbf{x})$ if $v \in Q_{d-k}(\mathbf{x})$ and $H(\mathbf{v}[k]) = i$ (as a reminder, $H(\mathbf{v}[k])$ is the hamming weight of the vector $\mathbf{v}$ restricted to only coordinates after the k^{th} coordinate).

We now define our solution S to the GW-SDP paramterized by t. We will find a suitable value for t when we analyze the value of the solution.

- For every k-length vector $\mathbf{x}^+$ of parity 1, and for all $v \in L_i(\mathbf{x}^+)$

$$\alpha_v = \begin{cases} 0 & \text{if } i \le \frac{d-k}{2} - t \\ \frac{\pi}{4}\left(1 - \frac{\frac{(d-k)}{2} - i}{t}\right) & \text{if } i \in (\frac{d-k}{2} - t, \frac{d-k}{2} + t) \\ \frac{\pi}{2} & \text{if } i \ge \frac{d-k}{2} + t \end{cases}$$

- For every k-length vector $\mathbf{x}^-$ of parity -1, and for all $v \in L_i(\mathbf{x}^-)$ assign α_v to be $\pi-$ the corresponding value for its neighboring vertex $\mathbf{x}^+$ of parity 1. i.e.

$$\alpha_v = \begin{cases} \pi & \text{if } i \le \frac{d-k}{2} - t \\ \pi - \frac{\pi}{4}\left(1 - \frac{\frac{(d-k)}{2} - i}{t}\right) & \text{if } i \in (\frac{d-k}{2} - t, \frac{d-k}{2} + t) \\ \frac{\pi}{2} & \text{if } i \ge \frac{d-k}{2} + t \end{cases}$$

Following is a schematic of the solution described above. L represents layers of subcubes with parity 1 and the L' represent their counterparts in subcubes of parity -1

We first compute the contribution of a fixed subcube $Q_{d-k}(\mathbf{x})$ to the GW-SDP objective. Consider a vertex $v \in L_i(\mathbf{x})$ where $i \in [0, \frac{(d-k)}{2} - t]$. This vertex is connected with equalities to its neighbours inside the subcube and with inequalities to its neighbours outside the subcube. Since all its neighbours inside the subcube are mapped to the same vector, the contribution to the SDP value of those edges is zero. Moreover all neighbors of v in different subcubes are mapped to the antipodal point of the vector v is mapped to (since neighboring subcubes have different parity). Therefore the contribution of every edge connected to this vertex is 0.

Similarly, for a vertex $v \in L_i(\mathbf{x})$ where $i \in [(d-k)/2 + t, (d-k)]$ the contribution of all its edges is 0.

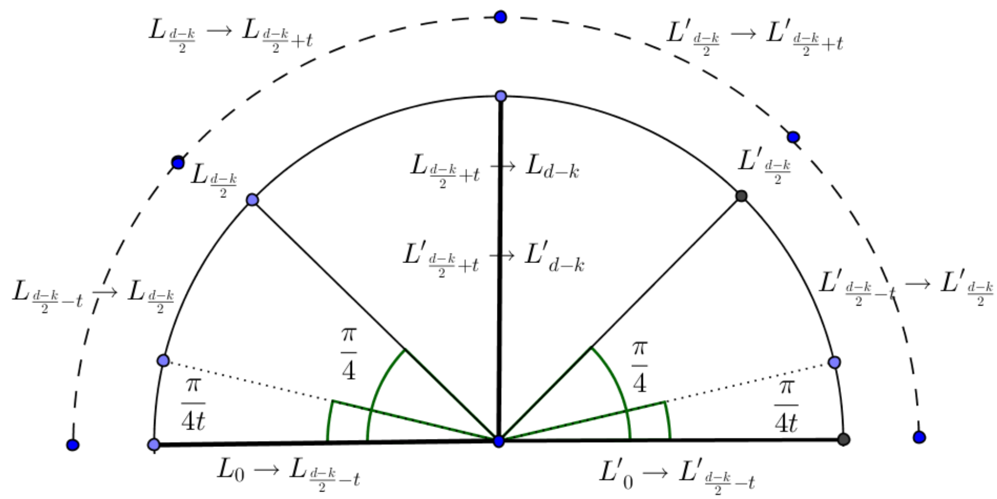

Figure 1: Schematic for SDP solution

Consider a vertex $v \in L_i(\mathbf{x})$ where $i \in ((d-k)/2-t, (d-k)/2)$. The total contribution of the neighbors of this vertex comes from the inequalities going out of the subcube, which is

$$k(1+cos(\pi-2\alpha_v)) \leq 2k$$

and from the equalities inside the subcube, which is

$$(d-k)(1-cos(\frac{\pi}{4t}))$$

The total contribution of edges adjacent to v therefore is

$$2k+(d-k)(1-cos(\frac{\pi}{4t})) \leq \mathcal{O}(k+\frac{(d-k)}{t^2})$$

The total fraction of vertices contained in layers $L_i(\mathbf{x})$ for $i = ((d-k)/2-t, (d-k)/2+t)$ is $\mathcal{O}(t/\sqrt{d-k})$ (for t=1 it is $\theta(\frac{1}{\sqrt{d-k}})$ and that is the layer with the largest fraction of vertices). Therefore the total contribution of a fixed subcube $Q_{d-k}(x)$ is bounded by

$$|V_{d-k}(x)|\mathcal{O}(\frac{t}{\sqrt{d-k}}\left(k+\frac{(d-k)}{t^2}\right)$$

Substituting $t = \sqrt{\frac{d-k}{k}}$ and summing the contribution over all subcubes $Q_{d-k}(x)$ we get that the fractional value of this SDP feasible solution is $\mathcal{O}(\frac{\sqrt{k}}{d})$.

3.3 Proof of Lemma 3.5

We show here one proof of Lemma 3.5. An alternative proof appears in Theorem 5.5: that proof not just shows a bound on the combinatorial optimum but also shows that there is a certificate for this bound using inconsistent cycles.

Consider the $d-k$ dimensional subcubes $Q_{d-k}(\mathbf{x})$ where $\mathbf{x}$ is a k dimensional vector. We first prove that in an optimum assignment, the assignment on any subcube $Q_{d-k}(\mathbf{x})$ is determined only by the parity of $\mathbf{x}$. In other words, if $\mathbf{x},\mathbf{y}$ are k dimensional vectors with the same parity then the assignments on the subcubes $Q_{d-k}(\mathbf{x}),Q_{d-k}(\mathbf{y})$ will be the same. We prove this by contradiction.

Let an optimum assignment be $\Gamma : V_d \to \{0,1\}$. For a subset of edges $E \subseteq E_d$ let $Val_\Gamma(E)$ be the number of unsatisfied edges in E. Let $Val_\Gamma(Q_{d-k}(\mathbf{x}))$ be the number of unsatisfied edges in the subcube $Q_{d-k}(\mathbf{x})$.

Let S be the set of pairs of k dimensional vectors $\mathbf{x_1},\mathbf{x_2}$ which differ in one coordinate. Given any two such vectors $\mathbf{x_1},\mathbf{x_2}$, let $E(\mathbf{x_1},\mathbf{x_2})$ be the set of edges (u,v) that go between the subcubes i.e. $u \in Q_{d-k}(\mathbf{x_1}), v \in Q_{d-k}(\mathbf{x_2})$. Therefore the total combinatorial value of the assignment Γ(i.e. total number of unsatisfied edges) is

$$\sum_{\mathbf{x}} Val_\Gamma(Q_{d-k}(\mathbf{x})) + \sum_{(\mathbf{x_1},\mathbf{x_2})\in S} Val_\Gamma(E(\mathbf{x_1},\mathbf{x_2})) =$$
$$\sum_{(\mathbf{x_1},\mathbf{x_2})\in S} \left(\frac{1}{k}\left(Val_\Gamma(Q_{d-k}(\mathbf{x_1})) + Val_\Gamma(Q_{d-k}(\mathbf{x_2}))\right) + Val_\Gamma(E(\mathbf{x_1},\mathbf{x_2})) \right) \quad (3.1)$$

Given the above expression let $\mathbf{x'_1},\mathbf{x'_2}$ be vectors such that the quantity inside the summation in the RHS above is minimum. Now consider the assignment in which for every vector $\mathbf{x}$ which has the same parity as $\mathbf{x'_1}$, the subcube $Q_{d-k}(\mathbf{x})$ has the same assignment as the subcube $Q_{d-k}(\mathbf{x'_1})$ in Γ. We do the same with $\mathbf{x'_2}$. It is easy to see that the above described assignment satisfies at least as many edges as Γ.

By the above argument for any optimal assignment Γ it is enough to specify two assignment functions $\Gamma_+ : Q_{d-k} \to \{0,1\}$ and $\Gamma_- : Q_{d-k} \to \{0,1\}$, one for subcubes for which the first k coordinates have parity 1 and one for subcubes for which the first k coordinates have parity -1. Let $Val(\Gamma_+)$ and $Val(\Gamma_-)$ be the number of edges not satisfied within the subcubes of positive and negative parity respectively. Let $Val(\Gamma_+,\Gamma_-)$ denote the number of edges not satisfied between a fixed subcube of positive parity and a fixed subcube of negative parity. The total number of edges not satisfied by the assignment Γ therefore is

$$2^{k-1}(Val(\Gamma_+) + Val(\Gamma_-)) + 2^{k-1}k(Val(\Gamma_+,\Gamma_-))$$

We now prove that without loss of generality the assignment Γ_- can be assumed to be the all 1's assignment $\mathbf{1}$ i.e. $\mathbf{1}(v) = 1$ for all $v \in Q_{d-k}$.

Consider any optimal assignment (Γ'_+,Γ'_-). Consider the assignment such that $\Gamma_- = \mathbf{1}$ and $\Gamma_+ = \Gamma'_+ \oplus \Gamma'_-$. Note that $Val(\Gamma'_+,\Gamma'_-) = Val(\mathbf{1},\Gamma'_+ \oplus \Gamma'_-)$. Also note that $Val(\mathbf{1}) = 0$ and $Val(\Gamma'_+ \oplus \Gamma'_-) \leq Val(\Gamma'_+) + Val(\Gamma'_-)$. Therefore the assignment $(\mathbf{1},\Gamma'_+ \oplus \Gamma'_-)$ is at least as good as (Γ'_+,Γ'_-).

In accordance with the above observations for an optimal assignment it is enough to specify the assignment Γ_+ for the positive parity subcubes.

Consider an optimal solution $(\Gamma'_+, \mathbf{1})$. Let $d' = d - k$. Let $V_0 \subseteq V_{d'}$ be the set of vertices v such that $\Gamma_+(v) = 0$ and $H_{1/2} \in V_{d'}$ be the set of vertices v such that $H(v) \leq d'/2$. Now it is easy to see that the number of edges unsatisfied by the assignment $(\Gamma'_+, \mathbf{1})$ is

$$\begin{aligned}
&2^{k-1}\left(Val(\Gamma_+) + Val(\mathbf{1}) + k * Val(\Gamma_+, \mathbf{1})\right) \\
&= 2^{k-1}\left(E[V_0, V_{d'} \backslash V_0] + 0 + k(|H_{1/2} - V_0| + |V_0 - H_{1/2}|)\right) \\
&= 2^{k-1}\left(E[V_0, V_{d'} \backslash V_0] + 0 + k(|H_{1/2}| - |H_{1/2} \cap V_0| + |V_0 - H_{1/2}|)\right) \\
&= 2^{k-1}\left(\frac{k * 2^{d'}}{2} - A(k, d')\right)
\end{aligned}$$

where $A(k,d') = k(|V_0 \cap H_{1/2}| - |V_0 - H_{1/2}|) - E[V, V_{d'} - V_0]$. We show in lemma 3.7 that the above defined quantity $A(k,d') \leq \frac{k\alpha}{2} * 2^{d'}$ for $k \leq \mathcal{O}(\sqrt{d})$ where α is a universal constant.

Therefore the total fraction of edges unsatisfied by the any optimum assignment is $\mathcal{O}(k/d)$ for $k = \mathcal{O}(\sqrt{d})$

Lemma 3.7. *Let Q_d be the hypercube of dimension d. Let V_d be the vertex set of the cube and let $V \subseteq V_d$. Let $H_{1/2}$ be the set of vertices with hamming weight $\leq d/2$. Let $k \leq \frac{\pi}{2}\mathcal{I}(Maj_d)$, where $\mathcal{I}(Maj_d) = \Theta(\sqrt{d})$ is the influence of the majority function on d coordinates. Then*

$$A(k,d) \stackrel{def}{=} k(|V \cap H_{1/2}| - |V \backslash H_{1/2}|) - E[V, V_d \backslash V] \leq \alpha \frac{k2^d}{2}$$

where α is a constant < 1.

Proof. Let Min_d, Maj_d be the minority/majority function over d variables. Let $\mathcal{I}(Maj_d) = \mathcal{I}(Min_d)$ be the influences of the functions Maj_d, Min_d. Note that Min_d is the indicator function of the $H_{1/2}$. Let $f : Q_d \to \{0,1\}$ be the indicator function of the set V. Then

$$\begin{aligned}
A(k,d) &= k(|V \cap H_{1/2}| - |V - H_{1/2}|) - E[V, V_d \backslash V] \\
&= \frac{2^{d'}}{2}\left(k\langle f, Min_{d'}\rangle) - \mathcal{I}(f)\right)
\end{aligned}$$

We now show the following inequality on Boolean functions over the cube, which proves the lemma when $k \leq \frac{\pi}{2}\mathcal{I}(Min_d) = \mathcal{O}(\sqrt{d})$.

Claim 3.8. *There exists a constant $\alpha < 1$, such that for any Boolean function f on the d-dimensional cube,*

$$\frac{\pi}{2} \cdot I(Min_d) \cdot (\langle f, Min_d \rangle - \alpha) \leq \mathcal{I}(f)$$

Proof. To prove the claim we use some well known facts about Fourier coefficients of Boolean functions. Note that for a Boolean Function $f : \{0,1\}^d \to \{+1,-1\}$ there is a well known and studied change of bases called the Walsh-Fourier transform. For any subset $S \subseteq [d]$ let $f(\hat{S})$ be the corresponding Fourier coefficient of f. Following are some standard facts about the Fourier coefficients proofs of which can be found in[23].

- $|\hat{Min}_d(\{i\})| \sim \sqrt{\frac{2}{\pi d}}$
- $\sum_{|S|\geq 2} \hat{Min}_d(S)^2 \leq (1-\frac{2}{\pi})$
- $\mathfrak{I}(Min_d) \sim \sqrt{\frac{2}{\pi}d}$
- $\sum_i |\hat{f}(\{i\})| \leq \mathfrak{I}(f)$

Now

$$\begin{aligned}
\frac{\pi}{2}\mathfrak{I}(Min_d)\langle f, Min_d\rangle &= \frac{\pi}{2}\mathfrak{I}(Min_d)\left(\sum \hat{f}(S)\hat{Min}_d(S)\right) \\
&\leq \frac{\pi}{2}\mathfrak{I}(Min_d)\left(\sum_{|S|=1}|\hat{f}(S)||\hat{Min}_d(S)| + \sum_{|S|\geq 2}\hat{f}(S)\hat{Min}_d(S)\right) \\
&= \frac{\pi}{2}\mathfrak{I}(Min_d)\left(\sum_{|S|=1}|\hat{f}(S)|\sqrt{\frac{2}{\pi d}} + \sum_{|S|\geq 2}\hat{f}(S)\hat{Min}_d(S)\right) \\
&\leq \frac{\pi}{2}\mathfrak{I}(Min_d)\left(\sum_{|S|=1}|\hat{f}(S)|\sqrt{\frac{2}{\pi d}} + \sqrt{\sum_{|S|\geq 2}\hat{f}^2(S)}\sqrt{\sum_{|S|\geq 2}\hat{Min}_d{}^2(S)}\right) \\
&\leq \sum_{|S|=1}|\hat{f}(S)| + \frac{\pi}{2}\mathfrak{I}(Min_d)\left(\sqrt{1-\frac{2}{\pi}}\right) \\
&\leq \mathfrak{I}(f) + \frac{\pi}{2}\mathfrak{I}(Min_d)\left(\sqrt{1-\frac{2}{\pi}}\right)
\end{aligned}$$

Putting $\alpha = \sqrt{1-\frac{2}{\pi}}$ proves the claim, and thus also completes the proof of Lemma 3.7. □

□

4 Proof of Lemma 3.4

In this section, we prove the gap preservation lemma 3.4. To prove the lemma we define the following general operation on Max-2-LIN($\mathbb{Z}_2$) instances on the cube.

Definition 4.1 (Tensor Max-2-LIN($\mathbb{Z}_2$))**.** Given two Max-2-LIN($\mathbb{Z}_2$) instances $\Gamma_1 : E_{d_1} \to \{0,1\}$ and $\Gamma_2 : E_{d_2} \to \{0,1\}$ supported on Hypercubes of dimensions d_1 and d_2, define a Max-2-LIN($\mathbb{Z}_2$) instance $\Gamma_1 \otimes \Gamma_2 : E_{d_1+d_2} \to \{0,1\}$ on the hypercube of dimension d_1+d_2 as follows. For an edge (v_1, v_2) let

$i(v_1,v_2)$ be the coordinate on which the corresponding vectors $\mathbf{v_1},\mathbf{v_2}$ differ. Let $\mathbf{v_i^{d_1}}$,$\mathbf{v_i^{d_2}}$ be the vector $\mathbf{v_i}$ restricted on the first d_1 coordinates and the last d_2 coordinates respectively. Then

$$\Gamma_1 \otimes \Gamma_2((v_1,v_2)) = \begin{cases} \Gamma_1((\mathbf{v_1^{d_1}},\mathbf{v_2^{d_1}})) & \text{if } i(v_1,v_2) \in [0,d_1-1] \\ \\ \Gamma_2((\mathbf{v_1^{d_2}},\mathbf{v_2^{d_2}})) & \text{if } i(v_1,v_2) \in [d_1,d_1+d_2-1] \end{cases}$$

Note that the above tensor product defines an edge according to the first instance or the second instance depending upon the coordinate along which the edge crosses. We prove the following lemmas about the above defined tensor product.

Lemma 4.2. *Let Γ_1 have combinatorial optimum $\geq \beta_1$ and Γ_2 have combinatorial optimum $\geq \beta_2$. Then the combinatorial optimum of $\Gamma_1 \otimes \Gamma_2$ is $\geq \frac{d_1\beta_1+d_2\beta_2}{d_1+d_2}$.*

Lemma 4.3. *Let Γ_1 have SDP optimum $\leq \alpha_1$ and Γ_2 have SDP optimum $\leq \alpha_2$. Then the SDP optimum of $\Gamma_1 \otimes \Gamma_2$ is $\leq \frac{d_1\alpha_1+d_2\alpha_2}{d_1+d_2}$.*

Note that given any (α,β) Max-2-LIN($\mathbb{Z}_2$) Instance Γ on the cube of dimension d define $\Gamma_i = \otimes_1^i \Gamma$. By lemmas 4.3 and 4.2 we get that Γ_i is an (α,β) Instance on the hypercube of dimension $i.d$. This proves lemma 3.4.

Proof of lemma 4.2. We prove the lemma by proving that the instance $\Gamma_1 \otimes \Gamma_2$ can be partitioned into edge disjoint copies of the instances Γ_1 and Γ_2. Consider any d_1+d_2 dimensional vector. Fix the last d_2 coordinates and vary the first d_1 coordinates within the space $\{0,1\}^{d_1}$. Note that the vectors generated by the above process naturally define a subset of edges of the cube $Q_{d_1+d_2}$. Also note that the subset of edges generated is an exact copy of Γ_1. Therefore repeating the above process for all choices of the last d_2 coordinates gives us 2^{d_2} edge disjoint copies of Γ_1 within $\Gamma_1 \otimes \Gamma_2$. Fixing the first d_1 coordinates and repeating the same line of argument as above we get 2^{d_1} edge disjoint copies of Γ_2 within $\Gamma_1 \otimes \Gamma_2$. Note that the copies described above form an edge partition of $\Gamma_1 \otimes \Gamma_2$. Therefore the combinatorial optimum of the instance is

$$\begin{aligned} &\geq \frac{1}{E_{d_1+d_2}}\left(2^{d_2}\frac{d_1 2^{d_1}}{2}\beta_1 + 2^{d_1}\frac{d_2 2^{d_2}}{2}\beta_2\right) \\ &\geq \frac{d_1\beta_1+d_2\beta_2}{d_1+d_2} \end{aligned}$$

□

Proof of lemma 4.3. It is enough to give one SDP solution which has the required value. Let the optimal *SDP* solution for Γ_1 be $S_1 : V_{d_1} \to \mathbb{R}^{2^{d_1}}$ and for Γ_2 be $S_2 : V_{d_2} \to \mathbb{R}^{2^{d_2}}$.

Define the following solution $S_1 \otimes S_2 : V_{d_1+d_2} \to \mathbb{R}^{2^{d_1+d_2}}$.

$$S_1 \otimes S_2(\mathbf{v}) = S_1(\mathbf{v^{d_1}}) \otimes S_2(\mathbf{v^{d_2}})$$

It is immediate by the properties of tensor products of vectors that $S_1 \otimes S_2$ is a valid SDP solution. We now compute the SDP value of $S_1 \otimes S_2$. Let $E_1 \in E_{d_1+d_2}$ be the set of edges which go through the first

d_1 coordinates and $E_2 \in E_{d_1+d_2}$ be the set of edges which go through the last d_2 coordinates. Note that $|E_1| = \frac{d_1}{2} 2^{d_1+d_2}$ and $|E_2| = \frac{d_2}{2} 2^{d_1+d_2}$. Therefore the SDP value achieved by the solution is

$$\begin{aligned}
&\frac{1}{E_{d_1+d_2}} \left(\sum_{(v_1,v_2)\in E_1} \|S_1 \otimes S_2(v_1) \pm S_1 \otimes S_2(v_2)\|^2 + \sum_{(v_1,v_2)\in E_2} \|S_1 \otimes S_2(v_1) \pm S_1 \otimes S_2(v_2)\|^2 \right) \\
&= \frac{1}{E_{d_1+d_2}} \left(\sum_{(v_1,v_2)\in E_1} \|S_2(\mathbf{v_1^{d_2}})\|^2 \| \|S_1(\mathbf{v_1^{d_1}}) \pm S_2(\mathbf{v_1^{d_1}})\|^2 + \sum_{(v_1,v_2)\in E_2} \|S_1(\mathbf{v_1^{d_1}})\|^2 \| \|S_2(\mathbf{v_1^{d_2}}) \pm S_2(\mathbf{v_1^{d_2}})\|^2 \right) \\
&= \frac{1}{E_{d_1+d_2}} \left(\alpha_1 d_1 2^{d_1+d_2} + \alpha_2 d_2 2^{d_1+d_2} \right) \\
&= \frac{d_1\alpha_1 + d_2\alpha_2}{d_1+d_2}
\end{aligned}$$

□

5 Towards solving Unique Games on the Hypercube

In this section we propose a candidate algorithm for solving Unique Games on the Hypercube. Our candidate algorithm is simply augmenting the Goemans Williamson SDP with appropriate triangle inequalities. We conjecture that the augmented SDP is strong enough to solve unique games on the Hypercube. In particular we show that our proposed instance $\Delta(k,d)$ defined in the previous section is indeed solved by this SDP. The motivation behind our conjecture comes from the fact which we show next that on a cycle of any length the augmented SDP has at best a constant gap. This implies in particular that an inconsistent cycle in a graph acts as a certificate for an unsatisfied edge in the SDP solution as well. Therefore the property of necessarily having many inconsistent cycles makes Max-2-LIN($\mathbb{Z}_2$) instances on a graph solvable by SDP. We end the section by showing that our instance indeed has a lot of inconsistent cycles and by conjecturing that in fact any instance on the Boolean Cube satisfies this property.

We begin by defining our augmented SDP.

Definition 5.1 (GW+). Given a graph $G = (V,E)$ and a Max-2-LIN($\mathbb{Z}_2$) instance $I : E \to \{0,1\}$ on it, let the set of equality edges be E^+ and the set of inequality edges be E^-. The augmented Goemans-Williamson SDP for the instance is defined as

$$\text{minimize } \frac{1}{4|E|} \left(\sum_{(u,v)\in E^+} \|x_u - x_v\|^2 + \sum_{(u,v)\in E^-} \|x_u + x_v\|^2 \right)$$

$$\begin{aligned}
\text{subject to } \; & \|x_u\|^2 = 1 \;\; (\forall\, u \in V) \\
& \|a_i - a_j\|^2 \le \|a_i - a_k\|^2 + \|a_k - a_j\|^2 \;\; (\forall\, i,j,k \in V, a_i = \pm x_i, a_j \pm x_j, a_k \pm x_k)
\end{aligned}$$

One way in which the GW+algorithm improves on the Goemans-Williamson algorithm is that is takes inconsistent cycles into account, as is formalised below.

Definition 5.2 (inconsistent cycles)**.** Let I be a Max-2-LIN($\mathbb{Z}_2$) instance defined on a graph G. A cycle in G is said to be inconsistent if no assignment can satisfy all edges of the cycle (note that there is always an assignment that satisfies all edges of a cycle but one).

Theorem 5.3. *Consider a Max-2-LIN($\mathbb{Z}_2$) instance I defined on a graph $G = (V,E)$, and suppose that there are $\varepsilon \cdot |E|$ edge-disjoint inconsistent cycles in the instance. Then given I, the value returned by GW+is at least ε.*

Theorem 5.3 is well known, but we give a proof for completeness.

Proof. First consider the case where the instance just contains one cycle C. If it is consistent, it is easy to see that the SDP achieves a value of 0. We now focus on inconsistent cycles. Let $u_0, u_1 \dots u_{n-1}$ be the vertices of the cycle in order and let $u_n = u_0$. Let E_i be edge connecting $u_i \to u_{i+1}$ and let $C(E_i)$ be defined to be 1 if there is an equality constraint on E_i and -1 otherwise. Define

$$sign(i) = \Pi_{j=0}^{i} C(E_i)$$

Note that w.l.o.g. $sign(0) = 1$ and $sign(n) = -1$ because the cycle is inconsistent. The objective function of GW+now is thus

$$\begin{aligned} \frac{1}{4}\left(\sum_{i=0}^{n} \|sign(i)X_{u_i} - sign(i+1)X_{u_{i+1}}\|^2\right) &\geq \frac{1}{4}\left(\|sign(0)X_{u_0} - sign(n)X_{u_n}\|^2\right) \\ &= \frac{1}{4}\|2(X_{u_0})\|^2 \\ &= 1 \end{aligned}$$

The first inequality follows from the triangle inequalities added to GW+. The above implies that GW+has no gap on a cycle.

Now for a general instance, note that the above implies that any inconsistent cycle in the given instance must contribute at least 1 to the value of the objective function in GW+. In particular if we can find $\varepsilon \cdot |E|$ inconsistent edge disjoint cycles in the given instance we can be assured that the GW+optimum is at least ε, as required. □

An interesting question is whether there are instances of Max-2-LIN($\mathbb{Z}_2$) on the hypercube which are ε unsatisfiable, and yet there are not enough disjoint inconsistent cycles that certify the value to be at least $\Omega(\varepsilon)$. We conjecture that in fact there are no such instances, and therefore that the GW+algorithm gives a constant approximation algorithm for Max-2-LIN($\mathbb{Z}_2$) on the hypercube.

Conjecture 5.4. *Given a Max-2-LIN($\mathbb{Z}_2$) instance on the Hypercube (V_d, E_d) such that in any labeling at least ε fraction of its edges are unsatisfied, then there are at least $\Omega(\varepsilon|E_d|)$ edge disjoint inconsistent cycles in the instance.*

One motivation behind our conjecture is the presence of a large number of cycles containing every edge – each edge is contained in $(d-1)$ 4-cycles. At least it is true that for our instance $\Delta(k,d)$, defined previously, the statement of the conjecture holds. Recall that when $k \leq \mathcal{O}(\mathcal{I}(Maj_d))$, the combinatorial optimum is $\Theta(\frac{k}{d})$

Theorem 5.5. *Let $I = \Delta(k,d)$ be the Max-2-LIN($\mathbb{Z}_2$) instance defined in Section 3, where $k \leq \mathcal{O}(I(Maj_d)) = \mathcal{O}(\sqrt{d})$. Then there are at least $\Omega(\frac{k}{d} \cdot |E|)$ edge disjoint inconsistent cycles in I, where E is the set of edges in I.*

Proof. We first investigate the number of inconsistent edge disjoint cycles in our instance between two subcubes of dimension $d-k$. The inconsistent edge disjoint cycles in the whole instance will just be their union over all subcubes.

Edge-disjoint paths. To find the required cycles, we first consider a $d-k$ dimensional subcube $Q_{d-k}(\mathbf{x})$ inside our instance, and let $H_{1/2}$ be the set of vertices with Hamming weight $\leq \frac{d-k}{2}$ inside it (we only consider the Hamming weight relative to the subcube). We would like to find many edge disjoint simple paths in the cube Q_{d-k} such that for every vertex $v \in H_{1/2}$ there are at least $\ell = \Theta(\sqrt{d})$ paths that start from it and end in a vertex outside of $H_{1/2}$, and such that at most ℓ paths end at any one vetrex. Note that if P is a path of this type, and if Q is taken to be the same path but on a neighbouring subcube $Q_{d-k}(\sigma_i(\mathbf{x}))$ (σ_i flips the i'th bit of $\mathbf{x}$), then the two paths can be joined to create an inconsistent cycle.

Note that the above problem is equivalent to the following flow system. Let every edge within Q_{d-k} have capacity 1, and add a source s that connects to every vertex $v \in H_{1/2}$ with an edge of capacity ℓ and a target t that connects to every vertex outside of $H_{1/2}$ with an edge of capacity ℓ. If this system has a flow that saturates the edges going out of s and into t, then we can find the needed paths in our instance: that follows since if such a flow exists there must also be an equivalent integral flow. Once an integral flow is achieved, it is easy to see that it can be broken into edge-independent paths inside the subcube.

To see whether the flow system is satisfiable or not we simply need to check the whether every $s-t$ cut is flow sufficient. Consider any cut $V \subset V_{d-k}$. Note that the demand of the cut is $\ell|(|V-H_{1/2}| - |H_{1/2}-V|)|$ and the capacity of the cut is $E(V, V_{d-k}-V)$. Note that lemma 3.7 implies that for $\ell \leq \mathcal{O}(\sqrt{d})$ the cut is flow sufficient.

Stitching paths together. For every path $P = P(\mathbf{x})$ that we found in $Q_{d-k}(\mathbf{x})$, we can take a corresponding path $P(\mathbf{y})$ in any other subcube. We thus have a system of disjoint paths in the subcubes of our instance. Let us show how to stitch them together to get edge disjoint cycles. For this purpose, consider the graph G on the subcube $Q_{d-k}(\mathbf{x})$ which connects two points when they are connected by one of our chosen paths. G is a bipartite graph, and because of the way the paths were selected, it is regular and each vertex has degree ℓ. It is well known that such a graph can always be partitioned into ℓ matchings (e.g. using Hall's theorem): this means that we can choose a color $i = i(P)$ for each path, $i \in \{1, \ldots, \ell\}$, such that no vertex connects to two paths with the same color.

Now each path $P(\mathbf{y})$ in a subcube $Q_{d-k}(\mathbf{y})$ can be matched to the similar path $P(\mathbf{y}')$ in $Q_{d-k}(\mathbf{y}')$, where $\mathbf{y}' = \sigma_i(\mathbf{y})$ and $i = i(P)$ is the index chosen by P (since $\ell \leq k$, also $i \leq k$). Joining the endpoints of those paths creates an inconsistent cycle, and it is easy to verify that this indeed gives a system of edge-disjoint inconsistent cycles in $\Delta(k,d)$.

Counting cycles. As we constructed $\ell \cdot 2^{d-k}$ disjoint paths in each subcube, and since each cycle consists of two such paths, the total number of cycles is $\ell \cdot 2^{d-k} \cdot 2^k/2 = \Omega(k \cdot 2^d)$. Since the number of edges in $\Delta(k,d)$ is $d \cdot 2^d$, the number of cycles is $\Omega(\frac{k}{d} \cdot |E|)$ as required. □

Remark 5.6. The use of Lemma 3.7 in the proof above can be replaced by a simple and direct probabilistic argument for constructing the disjoint paths.

References

[1] SANJEEV ARORA, BOAZ BARAK, AND DAVID STEURER: Subexponential algorithms for unique games and related problems. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science*, 2010. 2

[2] SANJEEV ARORA, RUSSELL IMPAGLIAZZO, WILLIAM MATTHEWS, AND DAVID STEURER: Improved algorithms for unique games via divide and conquer. In *In Electronic Colloquium on Computational Complexity*, pp. TR10–041, 2010. 2

[3] 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 40th ACM Symposium On The Theory Of Computing*, volume 40, pp. 21–28, 2008. 2

[4] BOAZ BARAK, FERNANDO G. S. L. BRANDÃO, ARAM WETTROTH HARROW, JONATHAN A. KELNER, DAVID STEURER, AND YUAN ZHOU: Hypercontractivity, sum-of-squares proofs, and their applications. In *Proceedings of the 44th Symposium on Theory of Computing Conference*, pp. 307–326, 2012. 3

[5] BOAZ BARAK, PARIKSHIT GOPALAN, JOHAN HÅSTAD, RAGHU MEKA, PRASAD RAGHAVENDRA, AND DAVID STEURER: Making the long code shorter, with applications to the unique games conjecture. *Electronic Colloquium on Computational Complexity (ECCC)*, 18:142, 2011. 3

[6] BOAZ BARAK, PRASAD RAGHAVENDRA, AND DAVID STEURER: Rounding semidefinite programming hierarchies via global correlation. In *FOCS*, pp. 472–481, 2011. 2

[7] 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, pp. 205–214, 2006. 2, 4

[8] MOSES CHARIKAR, KONSTANTIN MAKARYCHEV, AND YURY MAKARYCHEV: Near-optimal algorithms for Unique Games. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pp. 205–214, 2006. 2

[9] 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), pp. 94–114, 2006. 1

[10] EDEN CHLAMTAC, KONSTANTIN MAKARYCHEV, AND YURY MAKARYCHEV: How to play unique games using embeddings. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 48, pp. 687–696, 2006. 2

[11] U. FEIGE AND G. SCHECHTMAN: On the optimality of the random hyperplane rounding technique for max-cut. *Random Structures and Algorithms*, 20:403–440, 2002. 2, 4

[12] M. GOEMANS AND D. WILLIAMSON: A 0.878 approximation algorithm for MAX-2SAT and MAX-CUT. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pp. 422–431, 1994. 2

[13] MICHEL X. GOEMANS AND DAVID P. WILLIAMSON: Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6), 1995. 4, 7

[14] ANUPAM GUPTA AND KUNAL TALWAR: Approximating unique games. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, volume 24, pp. 99–106, 2006. 2

[15] H. KARLOFF: How good is the goemans-williamson max-cut algorithm? *SIAM Journal on Computing*, 29(1):336–350, 1999. 2, 4

[16] 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, pp. 767–775, 2002. 1, 2

[17] 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 IEEE Symposium on Foundations of Computer Science*, volume 45, pp. 146–154, 2004. 1, 2

[18] SUBHASH KHOT, PREYAS POPAT, AND RISHI SAKET: Approximate lasserre integrality gap for unique games. In *APPROX-RANDOM*, Lecture Notes in Computer Science, pp. 298–311. Springer, 2010. 3

[19] SUBHASH KHOT AND ODED REGEV: Vertex cover might be hard to approximate to within $2-\varepsilon$. In *Proceedings of the IEEE Conference on Computational Complexity*, volume 18, pp. 379–386, 2003. 1

[20] 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 IEEE Symposium on Foundations of Computer Science*, volume 46, pp. 53–62, 2005. 1, 3

[21] ALEXANDRA KOLLA: Spectral algorithms for Unique Games. In *Proceedings of the 25th ACM Conference on Computational Complexity*, volume 25, pp. 122–130, 2010. 2

[22] KONSTANTIN MAKARYCHEV AND YURY MAKARYCEV: How to play unique games on expanders. In *In Proceedings of the eighth Workshop on Approximation and Online Algorithms*, 2010. 2

[23] RYAN O'DONNELL: *Analysis of Boolean Functions*. Cambridge University Press, 2014. 13

[24] PRASAD RAGHAVENDRA: Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th ACM Symposium On The Theory Of Computing*, volume 40, pp. 245–254, 2008. 2

[25] PRASAD RAGHAVENDRA AND DAVID STEURER: Integrality gaps for strong sdp relaxations of unique games. In *FOCS*, pp. 575–585. IEEE Computer Society, 2009. 3

[26] PRASAD RAGHAVENDRA AND DAVID STEURER: Graph expansion and the Unique Games conjecture. In *Proceedings of the 42nd ACM Symposium On The Theory Of Computing*, volume 42, pp. 755–764, 2010. 1, 2

[27] PRASAD RAGHAVENDRA AND DAVID STEURER: Graph expansion and the unique games conjecture. In *STOC*, pp. 755–764, 2010. 3

[28] LUCA TREVISAN: Approximation algorithms for unique games. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, volume 46, pp. 197–205, 2005. 2

AUTHORS

Naman Agarwal
Graduate Student
Princeton University, Princeton, NJ, USA
namana@cs.princeton.edu
http://www.cs.princeton.edu/~namana

Guy Kindler
Researcher
Hebrew University, Jerusalem, Israel
gkindler@cs.huji.ac.il
http://www.cs.huji.ac.il/~gkindler/

Alexandra Kolla
Assistant Professor
University of Illinois Urbana-Champaign, Urbana-Champaign, Illinois, USA
akolla@illinois.edu
http://akolla.cs.illinois.edu/

Luca Trevisan
Professor
University of California Berkeley, Berkeley, CA, USA
luca@berkeley.edu
http://www.eecs.berkeley.edu/~luca/