# How to Play Unique Games against a Semi-Random Adversary

## Study of Semi-Random Models of Unique Games

Alexandra Kolla
Microsoft Research

Konstantin Makarychev
IBM Research

Yury Makarychev
TTIC

**Abstract**— In this paper, we study the average case complexity of the Unique Games problem. We propose a semi-random model, in which a unique game instance is generated in several steps. First an adversary selects a completely satisfiable instance of Unique Games, then she chooses an $\varepsilon$–fraction of all edges, and finally replaces ("corrupts") the constraints corresponding to these edges with new constraints. If all steps are adversarial, the adversary can obtain any $(1-\varepsilon)$–satisfiable instance, so then the problem is as hard as in the worst case. We show however that we can find a solution satisfying a $(1-\delta)$ fraction of all constraints in polynomial–time if at least one step is random (we require that the average degree of the graph is $\tilde{\Omega}(\log k)$). Our result holds only for $\varepsilon$ less than some absolute constant. We prove that if $\varepsilon \geq 1/2$, then the problem is hard in one of the models, that is, no polynomial–time algorithm can distinguish between the following two cases: (i) the instance is a $(1-\varepsilon)$–satisfiable semi–random instance and (ii) the instance is at most $\delta$–satisfiable (for every $\delta > 0$); the result assumes the 2–to–2 conjecture.

Finally, we study semi-random instances of Unique Games that are at most $(1-\varepsilon)$–satisfiable. We present an algorithm that distinguishes between the case when the instance is a semi-random instance and the case when the instance is an (arbitrary) $(1-\delta)$–satisfiable instances if $\varepsilon > c\delta$ (for some absolute constant $c$).

### 1. Introduction

In this paper, we study the average case complexity of the Unique Games problem in a semi-random model. In the Unique Games problem, we are given a graph $G = (V, E)$, a set of labels $[k] = \{0, \dots, k-1\}$ and a set of permutations $\pi_{uv}$ on $[k]$, one permutation for every edge $(u, v)$. Our goal is to assign a label (or state) $x_u \in [k]$ to every vertex $u$ so as to maximize the number of satisfied constraints of the form $x_v = \pi_{uv}(x_u)$. The value of the solution is the number of satisfied constraints.

The problem is conjectured to be very hard in the worst case. The Unique Games Conjecture (UGC) of Khot [17] states that for every positive $\varepsilon$, $\delta$ and sufficiently large $k$, it is NP-hard to distinguish between the case where at least a $1-\varepsilon$ fraction of constraints is satisfiable, and the case where at most a $\delta$ fraction of all constraints is satisfiable. The conjecture implies strong inapproximability results for many problems such as MAX CUT [18], Vertex Cover [19], Maximum Acyclic Subgraph [14], Max $k$-CSP [24], [15], [26], which are not known to follow from standard complexity assumptions.

This is an extended abstract. The full version of this paper is available at http://arxiv.org/abs/1104.3806.

There has been a lot of research on the Unique Games Conjecture in recent years. One direction of research has focused on developing polynomial-time approximation algorithms for arbitrary instances of unique games. The first algorithm was presented by Khot in his original paper on the Unique Games Conjecture [17], and then several algorithms were developed in papers by Trevisan [27], Gupta and Talwar [10], Charikar, Makarychev and Makarychev [8], Chlamtac, Makarychev, and Makarychev [9]. Another direction of research has been to study subexponential approximation algorithms for Unique Games. The work was initiated by Kolla [21] and Arora, Impagliazzo, Matthews and Steurer [4] who proposed subexponential algorithms for certain families of graphs. Then, in a recent paper, Arora, Barak and Steurer [3] gave a subexponential algorithm for arbitrary instances of Unique Games.

These papers, however, do not disprove the Unique Games Conjecture. Moreover, Khot and Vishnoi [20] showed that it is impossible to disprove the Conjecture by using the standard semidefinite programming relaxation for Unique Games, the technique used in the best currently known polynomial-time approximation algorithms for general instances of Unique Games. Additionally, Khot, Kindler, Mossel, and O'Donnell [18] proved that the approximation guarantees obtained in [8] cannot be improved if UGC is true (except possibly for lower order terms).

All that suggests that Unique Games is a very hard problem. Unlike many other problems, however, we do not know any specific families of hard instances of Unique Games. In contrast, we do know many specific hard instances of other problems. Many such instances come from cryptography; for example, it is hard to invert a one-way function $f$ on a random input, it is hard to factor the product $z = xy$ of two large prime numbers $x$ and $y$. Consequently, it is hard to satisfy SAT formulas that encode statements "$f(x) = y$" and "$xy = z$". There are even more natural families of hard instances of optimization problems; e.g.

- **3-SAT:** Feige's 3-SAT Conjecture [11] states that no randomized polynomial time algorithm can distinguish random instances of 3-SAT (with a certain clause to variable ratio) from $(1-\varepsilon)$–satisfiable instances of 3-SAT (with non-negligible probability).
- **Linear Equations in $\mathbb{Z}/2\mathbb{Z}$:** Alekhnovich's Conjecture [1] implies that given a random $(1-\varepsilon)$–satisfiable

instance of a system of linear equations in $\mathbb{Z}/2\mathbb{Z}$, no randomized polynomial time algorithm can find a solution that satisfies a $1/2 + \delta$ fraction of equations (for certain values of parameters $\varepsilon$ and $\delta$).

- **Maximum Clique Problem:** It is widely believed [16] that no randomized polynomial time algorithm can find a clique of size $(1+\varepsilon)\log_2 n$ in a $G(n, 1/2)$ graph with a planted clique of size $m = n^{1/2-\delta}$ (for every constant $\varepsilon, \delta > 0$).

No such results are known or conjectured for Unique Games. In order to better understand Unique Games, we need to identify, which instances of the problem are easy and which are potentially hard. That motivated the study of specific families of Unique Games. Barak, Hardt, Haviv, Rao, Regev and Steurer [6] showed that unique game instances obtained by parallel repetition are "easy" (we say that a family of $(1-\varepsilon)$–satisfiable instances is *easy* if there is a randomized polynomial-time algorithm that satisfies a constant fraction of constraints). Arora, Khot, Kolla, Steurer, Tulsiani, and Vishnoi [5] showed that unique games on spectral expanders are easy (see also Makarychev and Makarychev [23], and Arora, Impagliazzo, Matthews and Steurer [4]).

In this paper, we continue this line of research by investigating the hardness of semi-random (semi-adversarial) instances of Unique Games. In a semi-random model, an instance is generated in several steps; at each step, choices are either made adversarially or randomly. Semi-random models were introduced by Blum and Spencer [7] (who considered semi-random instances of the $k$-coloring problem) and then studied by Feige and Kilian [12], and Feige and Krauthgamer [13].

In this paper, we propose and study a model, in which a $(1-\varepsilon)$–satisfiable unique game instance is generated as follows:

1) **Graph Selection Step.** Choose the constraint graph $G = (V, E)$ with $n$ vertices and $m$ edges.
2) **Initial Instance Selection Step.** Choose a set of constraints $\{\pi_{uv}\}_{(u,v)\in E}$ so that the obtained instance is completely satisfiable.
3) **Edge Selection Step.** Choose a set of edges $E_\varepsilon$ of size $\varepsilon m = \varepsilon|E|$.
4) **Edge Corruption Step.** Replace the constraint for every edge in $E_\varepsilon$ with a new constraint.

Note that if an adversary performs all four steps, she can obtain an arbitrary $(1-\varepsilon)$–satisfiable instance, so, in this fully–adversarial case, the problem is as hard as in the worst case. The four most challenging semi-random cases are when choices at one out of the four steps are made randomly, and all other choices are made adversarially. The first case – when the graph $G$ is random and, in particular, is an expander – was studied by Arora, Khot, Kolla, Steurer, Tulsiani, and Vishnoi [5], who showed that this case is easy.

We present algorithms for the other three cases that satisfy a $(1-\delta)$ fraction of constraints (if the average degree of $G$ is at least $\tilde{\Omega}(\log k)$ and $\varepsilon$ is less than some absolute constant).

**Theorem 1.1.** *There exist $\varepsilon_0 > 0$, $k_0 \geq 2$, $C > 0$, such that for every $k \geq k_0$, $\varepsilon \in (0, \varepsilon_0)$, $\delta > C\max(\varepsilon, 1/k)$, there exists a randomized polynomial time algorithm that given a semi-random instance of Unique Games with parameter $\varepsilon$ generated in one of the three models (see Section 2.2 for details) on a graph $G$ with average degree at least $\tilde{\Omega}(\log k)\delta^{-3}$, finds a solution of value $(1-\delta)$ with probability $1 - o(1)$ taken over random choices of the semi-random adversary and the random choices of the algorithm.*

We analyze each model separately and establish more precise bounds on the parameters for each model. In the conference version of this paper, we present the algorithm for the "random edges, adversarial constraints" model for specific values of $\varepsilon$ and $\delta$ in Theorem 3.1. We also briefly describe the algorithm for "adversarial edges, random constraints" in Theorem 4.1. We present more general results and full proofs in the full version of the paper [22].

In our opinion, this is a very surprising result since the adversary has a lot of control over the semi-random instance. We want to point out that previously known approximation algorithms for Unique Games cannot find good solutions of semi-random instances. Also techniques developed for analyzing semi-random instances of other problems such as local analysis, statistical analysis, spectral gap methods, standard semidefinite programming techniques seem to be inadequate to deal with semi-random instances of Unique Games. To illustrate this point, consider the following example. Suppose that the set of corrupted edges is chosen at random, and all other steps are adversarial ("the random edges, adversarial constraints case"). The adversary generates a semi-random instance as follows. It first prepares two instances $\mathcal{I}_1$ and $\mathcal{I}_2$ of Unique Games. The first instance $\mathcal{I}_1$ is the Khot–Vishnoi instance [20] on a graph $G$ with the label set $[k] = \{0, \ldots, k-1\}$ and permutations $\{\pi^1_{uv}\}$ whose SDP value is $\varepsilon' < \varepsilon/2$ but which is only $k^{-\Omega(\varepsilon')}$ satisfiable. The second instance $\mathcal{I}_2$ is a completely satisfiable instance on the same graph $G$ with the label set $\{k, \ldots, 2k-1\}$ and permutations $\pi^2_{uv} = \mathrm{id}$. She combines these instances together: the combined instance is an instance on the graph $G$ with the label set $[2k] = \{0, \ldots, 2k-1\}$, and permutations $\{\pi_{uv} : \pi_{uv}(i) = \pi^1_{uv}(i) \text{ if } i \in [k], \text{ and } \pi_{uv}(i) = \pi^2_{uv}(i), \text{ otherwise}\}$. Once the adversary is given a random set of edges $E_\varepsilon$, she randomly changes ("corrupts") permutations $\{\pi^2_{uv}\}_{(u,v)\in E_\varepsilon}$ but does not change $\pi^1_{uv}$, and then updates permutations $\{\pi_{uv}\}_{(u,v)\in E_\varepsilon}$ accordingly. It turns out that the SDP value of $\mathcal{I}_2$ with corrupted edges is very close to $\varepsilon$, and therefore, it is larger than $\varepsilon'$, the SDP value of $\mathcal{I}_1$ (if we choose parameters properly). So in this case the SDP solution assigns positive weight only to the labels in $[k]$ from the first instance. That means that the SDP solution does not reveal any information about the optimal solution (the only integral solution we can obtain from the SDP solution has value $k^{-\Omega(\varepsilon)}$). Similarly, algorithms that analyze the spectral gap of the label extended graph cannot deal with this instance. Of course, in this example, we let our first instance, $\mathcal{I}_1$, to be the Khot–Vishnoi instance because it "cheats" SDP based algorithms. Similarly, we can take as $\mathcal{I}_1$ another instance that cheats another type of algorithms. For instance, if UGC is true, we can let $\mathcal{I}_1$ to be a $(1-\varepsilon')$–satisfiable unique game that is indistinguishable in polynomial-time from a $\delta$–satisfiable unique game.

Our algorithms work only for values of $\varepsilon$ less than some absolute constants. We show that this restriction is essential. For every $\varepsilon \geq 1/2$ and $\delta > 0$, we prove that no polynomial time algorithm satisfies a $\delta$ fraction of constraints in the "adversarial constraints, random edges" model (only the third step is random), assuming the 2–to–2 conjecture.

One particularly interesting family of semi-random unique games (captured by our model) are *mixed instances*. In this model, the adversary prepares a satisfiable instance, and then chooses a $\delta$ fraction of edges and replaces them with adversarial constraints (corrupted constraints); i.e. she performs all four steps in our model and can obtain an arbitrary $(1-\delta)$–satisfiable instance. Then the "nature" replaces every corrupted constraint with the original constraint with probability $1-\varepsilon$. In our model, this case corresponds to an adversary who at first prepares a list of corrupted constraints $\pi'_{uv}$, and then at the fourth step replaces constraints for edges in $E_\varepsilon$ with constraints $\pi'_{uv}$ (if an edge from $E_\varepsilon$ is not in the list, the adversary does not modify the corresponding constraint).

**Distinguishing Semi-Random At Most $(1-\varepsilon)$ Satisfiable Instances From Almost Satisfiable Instances** We also study whether semi-random instances of Unique Games that are at most $(1-\varepsilon)$–satisfiable can be distinguished from almost satisfiable instances of Unique Games. This question was studied for other problems. In particular, Feige's "Random 3-SAT Conjecture" states that it is impossible to distinguish between random instances of 3-SAT (with high enough clause density) and $(1-\delta)$–satisfiable instances of 3-SAT. In contrast, we show that in the "adversarial edges, random constraints" case (the fourth step is random), semi-random $(1-\varepsilon)$–satisfiable instances can be efficiently distinguished from (arbitrary) $(1-\delta)$–satisfiable instances when $\varepsilon > c\delta$ (for some absolute constant $c$). (This problem, however, is meaningless in the other two cases — when the adversary corrupts the constraints — since then she can make the instance almost satisfiable even if $\varepsilon$ is large.)

**Linear Unique Games** We separately consider the case of Linear Unique Games (MAX $\Gamma$-LIN). In the semi-random model for Linear Unique Games, we require that constraints chosen at the second and fourth steps are of the form $x_u - x_v = s_{uv} (\mathrm{mod}\ k)$. Note that in the "random edges, adversarial constraints" model, the condition that constraints are of the form $x_u - x_v = s_{uv} (\mathrm{mod}\ k)$ only restricts the adversary (and does not change how the random edges are selected). Therefore, our algorithm for semi-random general instances still applies to this case. However, in the "adversarial edges, random constraints" case, we need to sample constraints from a different distribution of permutations at the fourth step: for every edge $(u,v)$ we now choose a random shift permutation $x_v = x_u - s_{uv}$, where $s_{uv} \in_U \mathbb{Z}/k\mathbb{Z}$. We show that our algorithm still works in this case; the analysis however is different. We believe that it is of independent interest. We do not consider the case where only the initial satisfying assignment is chosen at random, since for Linear Unique Games, the initial assignment uniquely determines the constraints between edges (specifically, $s_{uv} = x_u - x_v (\mathrm{mod}\ k)$). Thus the case when only the second step is random is completely adversarial.

It is interesting that systems of linear equations affected by noise with more than two variables per equations are believed to be much harder. Suppose we have a consistent system of linear equations $Ax = b$ over $\mathbb{Z}/2\mathbb{Z}$. Then we randomly change an $\varepsilon$ fraction of entries of $b$. Alekhnovich [1] conjectured that no polynomial-time algorithm can distinguish the obtained instance from a completely random instance even if $\varepsilon \approx n^{-c}$, for some constant $c$ (Alekhnovich stated his conjecture both for systems with 3 variables per equation and for systems with an arbitrary number of variables per equation).

Our results can be easily generalized to Unique Games in arbitrary Abelian groups. We omit the details in the conference version of this paper.

### 1.1. *Brief Overview of Techniques*

We use different techniques to analyze different models. First, we outline how we solve semi-random unique games in the "adversarial constraints, random edges" model (see Section 3 for details). As we explained above, we cannot use the standard SDP relaxation to solve semi-random instances in this model. Instead, we consider a very unusual SDP program for the problem, which we call "Crude SDP" (C-SDP). This SDP is not even a relaxation for Unique Games and its value can be large when the instance is satisfiable. The C-SDP assigns a unit vector $u_i$ to every vertex $(u,i)$ of the label–extended graph (for a description of the label–extended graph we refer the reader to Section 2). We use vectors $u_i$ to define the length of edges of the label–extended graph: the length of $((u,i),(v,j))$ equals $\|u_i - v_j\|^2$. Then we find *super short edges* w.r.t. the C-SDP solution, i.e., those edges that have length $O(1/\log k)$. One may expect that there are very few short edges since for a given C-SDP most edges will be long if we choose the unique games instance at random. We prove, however, that for every C-SDP solution $\{u_i\}$, with high probability (over the semi-random instance) either

1) there are many super short edges w.r.t. $\{u_i\}$ in the satisfiable layer of the semi-random game,
2) or there is another C-SDP solution of value less than the value of the solution $\{u_i\}$.

Here, as we describe later on in Section 2, the "satisfiable layer" corresponds to the representation of the satisfying assignment in the label–extended graph. The proof shows how to combine the C-SDP solution with an integral solution so that the C-SDP value goes down unless almost all edges in the satisfiable layer are super short. We then show that this claim holds with high probability not only for one C-SDP solution but also for all C-SDP solutions simultaneously. The idea behind this step is to find a family $\mathcal{F}$ of representative C-SDP solutions and then use the union bound. One of the challenges is to choose a very small family $\mathcal{F}$, so that we can prove our result under the assumption that the average degree is only $\tilde{\Omega}(\log k)$. The result implies that w.h.p. there are many super short edges w.r.t. the optimal SDP solution.

Now given the set of super short edges, we need to find which of them lie in the satisfiable layer. We write and solve an LP relaxation for Unique Games, whose objective function depends on the set of super short edges. Then we run a rounding algorithm that rounds the C-SDP and LP solutions to a combinatorial solution using a variant of the "orthogonal separators" technique developed in [9].

Our algorithm for the "adversarial edges, random constraints" model is quite different. First, we solve the standard SDP relaxation for Unique Games. Now, however, we cannot claim that many edges of the label–extended graph are short. We instead find the "long" edges of the graph $G$ w.r.t. the SDP solution. We prove that most corrupted edges are long, and there are at most $O(\varepsilon)$ long edges in total (Theorem 4.4). We remove all long edges, and then solve the obtained highly–satisfiable instance using the algorithm of Charikar, Makarychev and Makarychev [8].

We also present algorithms for two more cases: the case of "adversarial edges, random constraints" where the constraints are of the special form MAX-Γ-LIN and the case of "random initial constraints".

Due to space limitations we focus on the "random edges, adversarial constraints" model in the conference version of this paper. Moreover, to simplify exposition we do not present the main result in full generality. We give more general results and full proofs in the full version of the paper [22].

## 2. Notation and Preliminaries

### 2.1. Label-Extended Graph and SDP Relaxation

**Label-Extended Graph** For a given instance of Unique Games on a constraint graph $G = (V, E)$, with alphabet size $k$ and constraints $\{\pi_{uv}\}_{(u,v)\in E}$ we define the *Label-Extended* graph $M(V' = V \times [k], E')$ associated with that instance as follows: $M$ has $k$ vertices $B_v = \{v_0, \cdots, v_{k-1}\}$ for every vertex $v \in V$. We refer to this set of vertices as the block corresponding to $v$. $M$ has a total of $|V|$ blocks, one for each vertex of $G$. Two vertices $u_i, v_j \in V'$ are connected by an edge if $(u, v) \in E$ and $\pi_{uv}(i) = j$.

We refer to a set of nodes $L = \{u^{(z)}{}_{i(z)}\}_{z=1}^{|V|}$ as a "layer" if $L$ contains exactly one node from each block $B_{u^{(z)}}$. We note that a layer $L$ can be seen as an assignment of labels to each vertex of $G$. If a layer $L$ consists of vertices with the same index $i$, i.e. $L = \{u_i^{(z)}\}_{z=1}^{|V|}$, we will call $L$ the $i$-th layer.

**Standard SDP for Unique Games** Our algorithms use the following standard SDP relaxation for Unique Games (see also [17], [20], [8], [9]).

$$\min \quad \frac{1}{2|E|} \sum_{(u,v)\in E} \sum_{i\in[k]} \|u_i - v_{\pi_{uv}(i)}\|^2$$

subject to:

$$\begin{aligned}
\sum_{i=1}^{k} \|u_i\|^2 &= 1 && \forall i \in [k];\\
\langle u_i, u_j\rangle &= 0 && \forall i, j \in [k] : i \neq j;\\
\langle u_i, v_j\rangle &\geq 0 && \forall u, v \in V,\ i, j \in [k];\\
\|u_i - w_l\|^2 + \|w_l - v_j\|^2 &\geq \|u_i - v_j\|^2
\end{aligned}$$

for all $u, v, w \in V,\ i, j, l \in [k]$.

In this relaxation, we have a vector variable $u_i$ for every vertex $u$ and label $i$. In the intended solution, $u_i$ is an indicator variable for the event "$x_u = i$". That is, if $x_u = i$ then $u_i = e$; otherwise, $u_i = 0$; where $e$ is a fixed unit vector. The objective function measures the fraction of *unsatisfied* constraints: if the unique game has value $1 - \varepsilon$, then the value of the intended SDP solution equals $\varepsilon$ (and, therefore, the value of the optimal SDP solution is at most $\varepsilon$). Given an SDP solution of value $\varepsilon$, the approximation algorithm of Charikar, Makarychev, and Makarychev [8] finds a solution of value $1 - O(\sqrt{\varepsilon \log k})$. We will use this approximation algorithm as a subroutine (we will refer to it as CMMa). We will also use the following fact.

**Lemma 2.1** (see e.g. Lemmas A.1 and A.2 in [9])**.** *Suppose, we are given two random Gaussian variables $\gamma_1$ and $\gamma_2$ with mean $0$ and variance $1$ (not necessarily independent), and a parameter $k \geq 2$. Let $\eta = 1/1000$, $\alpha = \eta/k$. Consider a threshold $t$ s.t. $\Pr(\gamma_1 \geq t) = \Pr(\gamma_2 \geq t) = \alpha$. Then $\Pr(\gamma_1 \geq t$ and $\gamma_2 \geq t) \geq \alpha\left(1 - \sqrt{\frac{1}{c^*}\operatorname{Var}(\gamma_1 - \gamma_2)\log k}\right)$ for some absolute constant $c^*$.*

### 2.2. Models

In what follows, we will use several models for generating semi-random $(1-\varepsilon)$–satisfiable instances of Unique Games:

1) **"Random Edges, Adversarial Constraints" Model.** The adversary selects a graph $G(V, E)$ on $n$ vertices and $m$ edges and an initial set of constraints

$\{\pi_{uv}\}_{(u,v)\in E}$ so that the instance is completely satisfiable. Then she adds every edge of $E$ to a set $E_\varepsilon$ with probability $\varepsilon$ (the choices for different edges are independent). Finally, the adversary replaces the constraint for every edge in $E_\varepsilon$ with a new constraint of her choice. Note that this model also captures the case where at the last step the constraints for every edge in $E_\varepsilon$ are replaced with a new random constraint (random adversary).

2) **"Adversarial Edges, Random Constraints" Model.** The adversary selects a graph $G(V,E)$ on $n$ vertices and $m$ edges and an initial set of constraints $\{\pi_{uv}\}_{(u,v)\in E}$ so that the instance is completely satisfiable. Then she chooses a set $E_\varepsilon$ of $\varepsilon|E|$ edges. Finally, the constraint for every edge in $E_\varepsilon$ is randomly replaced with a new constraint. We will also consider some variations of this model, where at all steps the constraints are MAX $\Gamma$-LIN. In particular, at the last step, choosing a random constraint of the form MAX $\Gamma$-LIN, corresponds to choosing a random value $s \in [|\Gamma|]$.
3) **"Random Initial Constraints" Model.** The adversary chooses the constraint graph $G = (V,E)$ and a "planted solution" $\{x_u\}$. Then for every edge $(u,v) \in E$, she randomly chooses a permutation (constraint) $\pi_{uv}$ such that $\pi_{uv}(x_u) = x_v$ (among $(k-1)!$ possible permutations). Then the adversary chooses an arbitrary set $E_\varepsilon$ of edges of size at most $\varepsilon|E|$ and replaces constraint $\pi_{uv}$ with a constraint $\pi'_{uv}$ of her choice for $(u,v) \in E_\varepsilon$.

Without loss of generality, we will assume, when we analyze the algorithms, that the initial completely satisfying assignment corresponds to the "zero" layer. I.e. for every edge $(u,v)$, $\pi_{uv}(0) = 0$. Note that in reality, the real satisfying assignment is hidden from us.

## 3. Random Edges, Adversarial Constraints

In this section, we study the "random edges, adversarial constraints" model and prove the following result.

**Theorem 3.1.** *There exists a polynomial-time approximation algorithm, that given an instance of unique games from the "random edges, adversarial constraints" model on graph $G$ with $Cn\log k(\log\log k)^2$ edges ($C$ is a sufficiently large absolute constant) and $\varepsilon \le 1/4$, finds a solution of value $1/2$.*

The main challenge in solving "random edges, adversarial constraints" unique games is that the standard SPD relaxation may assign zero vectors to layers corresponding to the optimal solution (as well as to some other layers) and assign non-zero vectors to layers, where every integral solution satisfies very few constraints. To address this issue, we introduce a new slightly modified SDP. As usual the SDP has a vector $u_i$ for every vertex–label pair $(u,i) \in V \times [k]$. We require that vectors $u_i$, $u_{i'}$ corresponding to the same vertex $u$ are orthogonal: $\langle u_i, u_{i'}\rangle = 0$ for all $u \in V$ and $i, i' \in [k]$, $i \ne i'$. We also impose triangle inequality constraints:

$$\frac{1}{2}\|u_i - v_j\|^2 + \frac{1}{2}\|u_{i'} - v_j\|^2 \ge 1,$$

for all $(u,v) \in E$ and $i, i', j \in [k]$, $i \ne i'$; and require that all vectors have unit length: $\|u_i\| = 1$ for all $u \in V$ and $i \in [k]$. Observe, that our SDP is not a relaxation[1], since the integer solution does not satisfy the last constraint. The objective function is

$$\mathbf{min} \quad \sum_{(u,v)\in E} \sum_{\substack{i\in[k]\\ j=\pi_{uv}(i)}} \frac{\|u_i - v_j\|^2}{2}.$$

Usually, this objective function measures the number of unsatisfied unique games constraints. However, in our case it does not. In fact, it does not measure any meaningful quantity. Note, that the value of the SDP can be arbitrary large even if the unique games instance is satisfiable. We call this SDP the Crude SDP or C-SDP. Given a C-SDP solution, we define the set of *super short* edges, which play the central role in our algorithm.

**Definition 3.2.** *We say that an edge $((u,i),(v,j))$ in the label–extended graph is super short, if $\|u_i - v_j\|^2 \le c^*\eta^2/\log k$, here $c^*$ is an absolute constant defined in Lemma 2.1; $\eta = 1/1000$. We denote the set of all super short edges by $\Gamma$.*

In Section 3.2, we prove the following surprising result (Theorem 3.3), which states that all but very few edges in the zero level of the label-extended graph are super short.

**Theorem 3.3.** *Let $k \in \mathbb{N}$ ($k \ge 2$), $\varepsilon \in [0, 1/4]$, $\eta = 1/1000$, and let $G = (V,E)$ be an arbitrary graph with at least $Cn\log k(\log\log k)^2$ edges. Consider a semi-random instance of Unique Games in the "random edges, adversarial constraints" model. Let $\{u_i\}$ be the optimal solution of the C-SDP. Then with probability tending to 1 (uniformly as $n \to \infty$) the set*

$$\Gamma^0 = \Gamma \cap \{((u,0),(v,0)) : (u,v) \in E\}$$

*contains at least $(1 - \varepsilon - \eta)|E|$ edges.*

We proceed as follows. First, we solve the C-SDP. Then, given the C-SDP solution, we write and solve an LP to obtain weights $x(u,i) \in [0,1]$ for every $(u,i) \in V \times [k]$. These weights are in some sense substitutes for lengths of vectors in the standard SDP relaxation. In the LP, for every vertex $u \in V$, we require that

$$\sum_{i\in[k]} x(u,i) = 1.$$

[1]Unless, the unique game is from a special family like Linear Unique Games.

The objective function is

$$\max \sum_{((u,i),(v,j))\in\Gamma} \min(x(u,i), x(v,j))$$

(note that the objective function depends on the C-SDP solution). Denote the value of the LP by $LP$. The intended solution of this LP is $x(u,0) = 1$ and $x(u,i) = 0$ for $i \neq 0$. Since the LP contribution of every edge in $\Gamma^0$ is 1, the value of the intended solution is at least $|\Gamma^0|$. Applying Theorem 3.3, we get $LP \geq |\Gamma^0| \geq (1-\varepsilon-\eta)|E| > 5/7\ |E|$. In the next section, we present an approximation algorithm (which rounds C-SDP and LP solutions) and its analysis. We prove the approximation guarantee in Lemma 3.5, which implies Theorem 3.1.

### *3.1. SDP and LP Rounding*

We now present an algorithm that given a C-SDP solution $\{u_i\}$ and an LP solution $\{x(u,i)\}$ finds an integer solution. This algorithm uses methods developed in CMMa [8] and CMMb [9]. We first present a procedure for sampling subsets of vertex–label pairs, which is an analog of the algorithm for finding orthogonal separators.

---

**LP Weighted Orthogonal Separators**
**Input:** An SDP solution $\{u_i\}$, an LP solution $\{x(u,i)\}$.
**Output:** A set $S$ of label vertex pairs $(u,i)$.

1) Set a parameter $\alpha = \eta/k$, which we call the probability scale.
2) Generate a random Gaussian vector $g$ with independent components distributed as $\mathcal{N}(0,1)$.
3) Fix a threshold $t$ s.t. $\Pr(\xi \geq t) = \alpha$, where $\xi \sim \mathcal{N}(0,1)$.
4) Pick a random uniform value $r$ in the interval $(0,1)$.
5) Find set
$$S = \{(u,i) \in V \times [k] : \langle u_i, g\rangle \geq t \text{ and } x(u,i) \geq r\}.$$
6) Return $S$.

---

The rounding algorithm is given below.

---

**LP and SDP Based Rounding Algorithm**
**Input:** An instance of unique games.
**Output:** An assignment of labels to the vertices.

1) Solve the SDP.
2) Find the set of all super short edges $\Gamma$.
3) Solve the LP.
4) Mark all vertices unprocessed.
5) while (there are unprocessed vertices)
   - Sample a set $S$ of vertex–label pairs using LP weighted orthogonal separators.
   - For all unprocessed vertices $u$:
     - Let $S_u = \{i : (u,i) \in S\}$
     - If $S_u$ contains exactly one element $i$, assign label $i$ to $u$ and mark $u$ as processed.

If after $nk/\alpha$ iterations, there are unprocessed vertices, the algorithm assigns arbitrary labels to them and terminates.

---

**Lemma 3.4.** *Let $S$ be an LP weighted orthogonal separator. For every $((u,i),(v,j)) \in \Gamma$ and $(u,i') \in V \times [k]$ ($i' \neq i$),*

1) $\Pr((u,i) \in S) = \alpha x(u,i)$.
2) $\Pr((u,i) \in S$ *and* $(v,j) \in S) \geq$
$\alpha(1-\eta)\min(x(u,i), x(v,j))$.
3) $\Pr((u,i) \in S; (v,j) \in S; (u,i') \in S) \leq$
$\alpha\eta/k \ \cdot \min(x(u,i), x(v,j))$.

*Proof:* We have

$$\begin{aligned}\Pr((u,i) \in S) &= \Pr(\langle u_i, g\rangle \geq t \text{ and } x(u,i) \geq r)\\ &= \Pr(\langle u_i, g\rangle \geq t)\Pr(x(u,i) \geq r)\\ &= \alpha x(u,i).\end{aligned}$$

Then, by Lemma 2.1 (using $\mathrm{Var}(\langle u_i, g\rangle - \langle v_j, g\rangle) = \|u_i - v_j\|^2 \leq c^*\eta^2/\log k$),

$$\begin{aligned}&\Pr((u,i) \in S \text{ and } (v,j) \in S)\\ &\qquad= \Pr(\langle u_i, g\rangle \geq t \text{ and } \langle v_j, g\rangle \geq t)\\ &\qquad\quad\times \Pr(\min(x(u,i), x(v,j)) \geq r)\\ &\qquad\geq \alpha(1-\eta)\min(x(u,i), x(v,j)).\end{aligned}$$

Finally, we have (below we use that $\langle u_i, g\rangle$ and $\langle u_{i'}, g\rangle$ are independent random variables)

$$\begin{aligned}&\Pr((u,i) \in S; (v,j) \in S; (u,i') \in S)\\ &\qquad\leq \Pr(\langle u_i, g\rangle \geq t)\Pr(\langle u_{i'}, g\rangle \geq t)\\ &\qquad\quad \Pr(\min(x(u,i), x(v,j)) \geq r)\\ &\qquad= \alpha^2 \min(x(u,i), x(v,j))\\ &\qquad= \alpha(\eta/k)\ \min(x(u,i), x(v,j)).\end{aligned}$$

■

**Lemma 3.5.** *Given a C-SDP solution $\{u_i\}$ and an LP solution $\{x(u,i)\}$ of value at least $LP \geq 5/7\ |E|$, the algorithm finds a solution to the unique games instance that satisfies at least a $1/2$ fraction of all constraints in the expectation.*

*Proof:* Consider an arbitrary edge $(u,v) \in E$. We estimate the probability that the algorithm assigns labels that satisfy the constraint $\pi_{uv}$. For simplicity of presentation, assume that $\pi_{uv}(i) = i$ (we may always assume this by renaming the labels of $v$). Let $\delta_i(u,v) = \min(x(u,i), x(v,i))$ if $((u,i),(v,i)) \in \Gamma$; and $\delta_i(u,v) = 0$, otherwise. Let $\delta(u,v) = \sum_i \delta_i(u,v)$. Consider an iteration at which both

$u$ and $v$ have not yet been processed. By Lemma 3.4 (item 2), if $((u,i),(v,i)) \in \Gamma$, then $\Pr(i \in S_u \text{ and } i \in S_v) \geq \alpha(1-\eta)\min(x(u,i),x(v,i))$. Then, by Lemma 3.4 (item 3) and the union bound, the probability that $S_u$ or $S_v$ contains more than one element and $i \in S_u$, $i \in S_v$ is at most $2\alpha\eta\min(x(u,i),x(v,i))$. Hence, the algorithm assigns $i$ to both $u$ and $v$ with probability at least

$$\alpha(1-3\eta)\min(x(u,i),x(v,i)) = \alpha(1-3\eta)\delta_i(u,v).$$

The probability that the algorithm assigns the same label to $u$ and $v$ is at least

$$\sum_{i:((u,i),(v,i))\in\Gamma} \alpha(1-3\eta)\delta_i(u,v) = \alpha(1-3\eta)\delta(u,v).$$

The probability that the algorithm assigns a label to $u$ is at most $\alpha$ and similarly the probability that the algorithm assigns a label to $v$ is at most $\alpha$. Thus the probability that it assigns a label to either $u$ or $v$ is at most $\alpha(2-(1-3\eta)\delta(u,v))$.

The probability that the algorithm assigns the same label to $u$ and $v$ at one of the iterations is at least (note that the probability that there are unlabeled vertices when the algorithm stops after $nk/\alpha$ iterations is exponentially small, therefore, for simplicity we may assume that the number of iterations is infinite)

$$\sum_{t=0}^{\infty}(1-\alpha(2-(1-3\eta)\delta(u,v)))^t\alpha(1-3\eta)\delta(u,v) = \frac{\alpha(1-3\eta)\delta(u,v)}{\alpha(2-(1-3\eta)\delta(u,v))} = \frac{(1-3\eta)\delta(u,v)}{2-(1-3\eta)\delta(u,v)}.$$

The function $t \mapsto t/(2-t)$ is convex on $(0,2)$ and

$$\frac{1}{|E|}\sum_{(u,v)\in E}(1-3\eta)\delta(u,v) \geq 5/7\,(1-3\eta) > 2/3.$$

Thus, by Jensen's inequality, the expected fraction of satisfied constraints is at least

$$\frac{1}{|E|}\sum_{(u,v)\in E}\frac{(1-3\eta)\delta(u,v)}{2-(1-3\eta)\delta(u,v)} \geq \frac{2/3}{2-2/3} = \frac{1}{2}.$$

■

### *3.2. Lower Bound on the Number of Super Short Edges: Proof of Theorem 3.3*

We need the following lemma.

**Lemma 3.6.** *Let $\gamma \in (0,1/2)$, $\varepsilon \in (0,1/4)$. Let $G=(V,E)$ be a graph on $n$ vertices with $|E| \geq Cn\gamma^{-1}\log(\gamma^{-1})$ for some significantly large absolute constant $C$. Suppose, that $\{Z_{uv}\}_{(u,v)\in E}$ are i.i.d. Bernoulli random variables taking values $1$ with probability $\varepsilon$ and $0$ with probability $(1-\varepsilon)$. Define the payoff function $p:\{0,1\}\times\mathbb{R}\to\mathbb{R}$ as follows*

$$p(z,\alpha) = \begin{cases} -2\alpha, & \text{if } z=1; \\ \alpha, & \text{if } z=0. \end{cases}$$

*Then, with probability at least $1-e^{-n}$ for every set of unit vectors $\{u_0\}_{u\in V}$ satisfying*

$$\frac{1}{2}\sum_{(u,v)\in E}\|u_0-v_0\|^2 \geq \gamma|E| \tag{1}$$

*the following inequality holds*

$$\sum_{(u,v)\in E} p(Z_{uv},\|u_0-v_0\|^2) > 0. \tag{2}$$

We need the following dimension reduction lemma, which is based on the Johnson–Lindenstrauss Lemma and is fairly standard (see the full version of the paper for a proof).

**Lemma 3.7.** *For every $\nu \in (0,1/2)$ and $\eta = 1/1000$, there exists a set $N$ of unit vectors of size $e^{O(\log^2(1/\nu))}$, such that for every set of unit vectors $\mathcal{U}$ and every set of pairs $E \subset \mathcal{U}\times\mathcal{U}$, there exists a set $E^* \subset E$ of size at least $(1-\nu)|E|$ and a map $\varphi:\mathcal{U}\to N$ satisfying the following property: for every $(u,v)\in E^*$,*

$$(1-\eta)\|u-v\|^2-\nu \leq \|\varphi(u)-\varphi(v)\|^2 \leq (1+\eta)\|u-v\|^2+\nu.$$

*We call the set $N$ a net.*

*Proof of Lemma 3.6:* Let $\nu = \eta\gamma$ and $N$ be the net of size at most $e^{C'(\log^2(1/\gamma))}$ from Lemma 3.7 ($C'$ is a constant). Suppose, that for a given realization $\{Z^*_{uv}\}_{(u,v)\in E}$ of $\{Z_{uv}\}_{(u,v)\in E}$ there exists a set of unit vectors $\{u_0\}_{u\in V}$ satisfying condition (1) and violating (2). Embed vectors $\{u_0\}_{u\in V}$ into the net $N$ using Lemma 3.7 so that for all edges $(u,v)\in E^*$,

$$(1-\eta)\|u_0-v_0\|^2-\eta\gamma \leq \|u^*-v^*\|^2 \leq (1+\eta)\|u_0-v_0\|^2+\eta\gamma,$$

here $u^*$ is the image of $u_0$; $v^*$ is the image of $v_0$; and $|E^*| \geq (1-\eta\gamma)|E|$.

We now derive inequalities similar to (1) and (2) for vectors $\{u^*\}_{u\in V}$. Write,

$$\sum_{(u,v)\in E\setminus E^*}\|u_0-v_0\|^2 \leq \max_{u,v\in V}\{\|u_0-v_0\|^2\}\cdot|E\setminus E^*| \leq 4\eta\gamma|E|;$$

$$\sum_{(u,v)\in E^*}\|u_0-v_0\|^2 \geq 2\gamma|E|-4\eta\gamma|E| = 2(1-2\eta)\gamma|E|.$$

Hence,

$$\sum_{(u,v)\in E}\|u^*-v^*\|^2 \geq \sum_{(u,v)\in E}\big((1-\eta)\|u_0-v_0\|^2-\eta\gamma\big) \geq 2(1-\eta)(1-2\eta)\gamma|E|-\eta\gamma|E| \geq \gamma|E|. \tag{3}$$

Define a new payoff function (for $z\in\{0,1\}$, $\alpha\geq 0$),

$$p_\eta(z,\alpha) = \begin{cases} -2\,(1+2\eta)\,\alpha, & \text{if } z=1; \\ (1-2\eta)\,\alpha, & \text{if } z=0. \end{cases}$$

Observe, that $p_\eta(z,\alpha) \leq p(z,\alpha)$ for every $\alpha\geq 0$ and $z\in\{0,1\}$. Moreover, for $(u,v)\in E^*$,

$$p_\eta(z,\|u^*-v^*\|^2) \leq p(z,\|u_0-v_0\|^2)+3\eta\gamma.$$

Thus, we have,

$$\sum_{(u,v)\in E^*} p_\eta(Z^*_{uv}, \|u^*-v^*\|^2)$$
$$\le \Big(\sum_{(u,v)\in E^*} p(Z^*_{uv}, \|u^*-v^*\|^2)\Big) + 3\eta\gamma|E^*|$$
$$\le -\Big(\sum_{(u,v)\in E\setminus E^*} p(Z^*_{uv}, \|u^*-v^*\|^2)\Big) + 3\eta\gamma|E^*|$$
$$\le 2\max_{u,v\in V}\{\|u^*-v^*\|^2\}|E\setminus E^*| + 3\eta\gamma|E|$$
$$\le 11\eta\gamma|E|.$$

Then,

$$\sum_{(u,v)\in E\setminus E^*} p_\eta(Z^*_{uv}, \|u^*-v^*\|^2) \le$$
$$\le 2(1+2\eta)\max_{u,v\in V}\{\|u^*-v^*\|^2\}|E\setminus E^*| \le 9\eta\gamma|E|.$$

We get

$$\sum_{(u,v)\in E} p_\eta(Z^*_{uv}, \|u^*-v^*\|^2) \ \le\ 20\eta\gamma|E|. \tag{4}$$

Thus, the existence of unit vectors $\{u_0\}_{u\in V}$ satisfying condition (1) and violating (2) implies the existence of vectors $\{u^*\}_{u\in V}$ satisfying (3) and (4). We now show that for a random $\{Z_{uv}\}$ such vectors $\{u^*\}$ exist with exponentially small probability.

Fix a sequence $\{u^*\}_{u\in V}$ with elements $u^*\in N$ satisfying $\sum_{(u,v)\in E}\|u^*-v^*\| \ge \gamma|E|$ (see (3)). Denote $Y_{uv} = p_\eta(Z_{uv}, \|u^*-v^*\|^2)$. We would like to find an upper bound on $\Pr(\sum_{(u,v)\in E} Y_{uv} < 20\eta\gamma|E|)$. Let $\mu = \mathbb{E}\sum_{(u,v)\in E} Y_{uv}$, $\mu_{uv} = \mathbb{E}Y_{uv}$. Then,

$$\mu_{uv} \equiv \mathbb{E}p_\eta(Z_{uv}, \|u^*-v^*\|^2)$$
$$= ((1-2\eta)(1-\varepsilon) - 2(1+2\eta)\varepsilon)\,\|u^*-v^*\|^2$$
$$\ge \frac{1}{4}\|u^*-v^*\|^2.$$

Hence, $|Y_{uv}| \le 2(1+2\eta)\|u^*-v^*\|^2 \le 10\mu_{uv}$ a.s., and

$$\mu = \mathbb{E}\sum_{(u,v)\in E} Y_{uv} \ge \frac{1}{4}\sum_{(u,v)\in E}\|u^*-v^*\|^2 \ge \frac{\gamma}{4}|E|.$$

By Hoeffding's inequality,

$$\Pr\Big(\sum_{(u,v)\in E} Y_{uv} \le \frac{\mu}{2}\Big) \le \exp\Big(-\frac{\mu^2}{8\sum_{(u,v)\in E}(20\mu_{uv})^2}\Big)$$
$$\le \exp\Big(-\frac{\mu}{3200\max\mu_{uv}}\Big)$$
$$\le \exp\big(-\frac{\mu}{15000}\big) \le \exp\big(-C''\gamma|E|\big),$$

where $C'' = 1/60000$.

The number of all possible sequences $\{u^*\}_{u\in V}\subset N$ is at most

$$|N|^n = \exp\left(C'n\log^2(\gamma^{-1})\right).$$

By the union bound with probability at least

$$1-\exp(C'n\log(1/\gamma) - C''\gamma|E|) \ge 1-\exp(-n)$$

for random $\{Z_{uv}\}_{(u,v)\in E}$, there does not exist a set of unit vectors $\{u_0\}_{u\in V}$ satisfying condition (1) and violating (2). Here we used that $C$ is chosen to be sufficiently large and, consecutively, $|E| \ge Cn\gamma^{-1}\log(\gamma^{-1}) \ge ((C'+1)/C'')\,\gamma^{-1}\log(\gamma^{-1})n$. ∎

*Proof of Theorem 3.3:* Let $\{u_i^*\}$ be the optimal SDP solution. Pick a unit vector $e$ orthogonal to all vectors $u_i^*$. Define a new SDP solution $u_0^{int} = e$ and $u_i^{int} = u_i^*$ for $i\ne 0$ (for all $u\in V$). Note that restricted to $\{u_0^{int}\}_{u\in V}$ this solution is integral. Since $\{u_i^*\}$ is the optimal solution,

$$\sum_{(u,v)\in E}\sum_{\substack{i\in[k]\\ j=\pi_{uv}(i)}}\|u_i^*-v_j^*\|^2 \le \sum_{(u,v)\in E}\sum_{\substack{i\in[k]\\ j=\pi_{uv}(i)}}\|u_i^{int}-v_j^{int}\|^2.$$

Denote by $E_\varepsilon$ the set of corrupted edges. Let $Z_{uv}=1$, if $(u,v)\in E_\varepsilon$ and $Z_{uv}=0$, otherwise. Let $\tilde E_\varepsilon = \{(u,v)\in E : \pi_{uv}(0)\ne 0\}$. Clearly, $\tilde E_\varepsilon \subset E_\varepsilon$. Write,

$$\sum_{(u,v)\in E}\sum_{\substack{i\in[k]\\ j=\pi_{uv}(i)}}\|u_i^*-v_j^*\|^2 - \|u_i^{int}-v_j^{int}\|^2 =$$
$$= \sum_{(u,v)\in E\setminus\tilde E_\varepsilon}\|u_0^*-v_0^*\|^2 +$$
$$+ \sum_{(u,v)\in\tilde E_\varepsilon}\|u_0^*-v^*_{\pi_{uv}(0)}\|^2 + \|u^*_{\pi_{vu}(0)}-v_0^*\|^2$$
$$- \sum_{(u,v)\in\tilde E_\varepsilon}\|u_0^{int}-v^{int}_{\pi_{uv}(0)}\|^2 + \|u^{int}_{\pi_{vu}(0)}-v_0^{int}\|^2.$$

For $(u,v)\in\tilde E_\varepsilon$, we have $\|u_0^{int}-v^{int}_{\pi_{uv}(0)}\|^2 = \|u^{int}_{\pi_{vu}(0)} - v_0^{int}\|^2 = 2$ and $\|u_0^*-v^*_{\pi_{uv}(0)}\|^2 \le 2-\|u_0^*-v_0^*\|^2$. Thus,

$$\sum_{(u,v)\in E}\sum_{\substack{i\in[k]\\ j=\pi_{uv}(i)}}\|u_i^*-v_j^*\|^2 - \|u_i^{int}-v_j^{int}\|^2$$
$$\ge \sum_{(u,v)\in E\setminus\tilde E_\varepsilon}\|u_0^*-v_0^*\|^2 - 2\sum_{(u,v)\in\tilde E_\varepsilon}\|u_0^*-v_0^*\|^2$$
$$\ge \sum_{(u,v)\in E\setminus E_\varepsilon}\|u_0^*-v_0^*\|^2 - 2\sum_{(u,v)\in E_\varepsilon}\|u_0^*-v_0^*\|^2$$
$$= \sum_{(u,v)\in E} p(Z_{uv}, \|u_0^*-v_0^*\|^2),$$

where $p(\cdot,\cdot)$ is the function from Lemma 3.6. Therefore,

$$\sum_{(u,v)\in E} p(Z_{uv}, \|u_0^*-v_0^*\|^2) \ \le$$
$$\le \sum_{(u,v)\in E}\sum_{\substack{i\in[k]\\ j=\pi_{uv}(i)}}\|u_i^*-v_j^*\|^2 - \|u_i^{int}-v_j^{int}\|^2 \le 0.$$

Since the average degree of $G$ is at least $C \log k \log^2 \log k$ we can apply Lemma 3.6 with $\gamma = c^* \eta^3/(4 \log k)$. We get

$$\Pr\Big(\frac{1}{2|E|} \sum_{(u,v)\in E} \|u_0^* - v_0^*\|^2 < \gamma\Big) \ \ge\ 1 - e^{-n}.$$

If $\frac{1}{2|E|} \sum_{(u,v)\in E} \|u_0^* - v_0^*\|^2 < \gamma$, then by the Markov inequality, for all but $\eta/2$ fraction of edges $(u,v) \in E$, $\|u_0^* - v_0^*\|^2 \le c^* \eta^2/\log k$.

Finally, we lower bound the size of $\Gamma_0$. By definition, $((u,0),(v,0)) \in \Gamma_0$ if $\pi_{uv}(0) = 0$ (i.e., $(u_0, v_0)$ is an edge of the label–extended graph) and $\|u_0^* - v_0^*\|^2 \le c^* \eta^2/\log k$. By the Chernoff bound, $|E_\varepsilon| \le (\varepsilon + \eta/2)|E|$ with probability $(1 - o(1))$; therefore, $\pi_{uv}(0) \ne 0$ for at most an $(\varepsilon + \eta/2)$ fraction of edges. Thus, with probability $(1 - o(1))$, there are at least $(1 - \varepsilon - \eta)|E|$ super short edges. ■

### 3.3. Hardness: Semi-Random Instances for $\varepsilon \ge 1/2$

The problem becomes hard when $\varepsilon \ge 1/2$. In the full version of the paper [22], we prove the following theorem.

**Theorem 3.8.** *For every $\varepsilon \ge 1/2$ and $\delta > 0$, no polynomial-time algorithm can distinguish with probability greater than $o(1)$ between the following two cases:*

1) *Yes Case: the instance is a $(1-\varepsilon)$–satisfiable semi-random instance (in the "random edges, adversarial constraints" model),*
2) *No Case: the instance is at most $\delta$–satisfiable.*

*This result holds if the 2–to–2 conjecture holds.*

The 2–to–2 conjecture follows from Khot's 2–to–1 conjecture (see the full version of this paper [22] for details).

**Definition 3.9.** *In a 2–to–2 game, we are given a graph $G = (V, E)$, a set of labels $[k] = \{0, \dots, k-1\}$ ($k$ is even) and set of constraints, one constraint for every edge $(u, v)$. Each constraint is defined by a 2–to–2 predicate $\Pi_{uv}$: for every label $i$ there are exactly two labels $j$ such that $\Pi_{uv}(i, j) = 1$ (the predicate is satisfied); similarly, for every $j$ there are exactly two labels $i$ such that $\Pi_{uv}(i, j) = 1$. Our goal is to assign a label $x_u \in [k]$ to every vertex $u$ so as to maximize the number of satisfied constraints $\Pi_{uv}(x_u, x_v) = 1$. The value of the solution is the number of satisfied constraints.*

**Definition 3.10.** *The 2–to–2 conjecture states that for every $\delta > 0$ and sufficiently large $k$, there is no polynomial time algorithm that distinguishes between the following two cases (i) the instance is completely satisfiable and (ii) the instance is at most $\delta$–satisfiable.*

## 4. Adversarial Edges, Random Constraints

**Theorem 4.1.** *There exists a polynomial-time approximation algorithm that given $k \in \mathbb{N}$ ($k \ge k_0$), $\varepsilon \le \varepsilon_0$, and a semi-random instance of unique games from the "adversarial edges, random constraints" model on graph $G = (V, E)$ with at least $Cn \log k$ edges ($C$, $k_0 \ge 2$ and $\varepsilon_0 \in (0, 1/2)$ are absolute constants) finds a solution of value $1/2$ with probability $1 - o(1)$.*

**Definition 4.2.** *We say that an edge $(u, v) \in E$ is $\zeta$-long with respect to an SDP solution $\{u_i\}$, if*

$$\frac{1}{2} \sum_{i\in[k]} \|u_i - v_{\pi_{uv}(i)}\|^2 > \zeta.$$

Our algorithm proceeds in several steps. First, it solves the standard SDP relaxation for Unique Games. Then it removes "long edges" with respect to the SDP solution, and finally it runs the CMMa [8] algorithm to solve the unique game on the remaining graph (the CMMa algorithm will again solve the SDP relaxation for Unique Games — it cannot reuse our SDP solution).

Now we formally present the algorithm.

---

**Input:** An instance of unique games.
**Output:** An assignment of labels to the vertices.

1) Solve the SDP and obtain an SDP solution $\{u_i^*\}$.
2) Remove all $1/16$–long (with respect to $\{u_i^*\}$) edges $(u, v) \in E$ from the graph $G$. Denote the new graph $G^*$.
3) Solve the SDP on the graph $G^*$ and run the CMMa algorithm.

---

In the full version of the paper we prove Theorem 4.4 that shows that after removing all $1/16$–long edges from the graph $G$, the unique games instance contains at most $c|E|/\log k$ corrupted constraints w.h.p. Since the value of the optimal SDP is at most $\varepsilon$, the algorithm removes at most $16\varepsilon \le 16\varepsilon_0$ edges at step 2. In the remaining graph, $G'$, the CMMa algorithm finds an assignment satisfying $1 - O(\sqrt{c}) > 3/4$ fraction of all constraints. This assignment satisfies at least $3/4 - 16\varepsilon_0 \ge 1/2$ fraction of all constraints in $G$.

**Remark 4.3.** *In the previous section we proved that a typical instance of unique games in the "random edges, adversarial constraints" model contains many "super short" edges of the label-extended graph. Then we showed how we can find an assignment satisfying many super short edges. Note, that edges in the set $E_\varepsilon$ are not necessarily short or long. In this section, we use a very different property: in the typical instance of unique games in the "adversarial edges, random constraints" model, most edges in the set $E_\varepsilon$ are long. However, note that the label-extended graph does not have to have any super short edges at all.*

**Theorem 4.4.** *Let $k \in \mathbb{N}$ ($k \ge k_0$), $\varepsilon \in [0, 1]$, $c \in (0, 1)$. Consider a graph $G = (V, E)$ with at least $(C/c)\, n \log k$ edges and a unique game instance on $G$ ($C$ and $k_0$ are absolute constants). Suppose that all constraints for edges* *in $E_\varepsilon$ are chosen at random; where $E_\varepsilon \subset E$ is a set of edges of size $\varepsilon|E|$. Then, the set $E_\varepsilon$ contains less than $c|E|/\log k$ $1/16$–short edges w.r.t. every SDP solution $\{u_i\}$ with probability $1 - o(1)$.*

We give the proof in the full version of the paper [22].

## REFERENCES


[1] M. Alekhnovich. *More on Average Case vs Approximation Complexity.* Proceedings of the 44st IEEE Symposium on Foundations of Computer Science: pp, 298–307, 2003.

[2] G. Andersson, L. Engebretsen, and J. Håstad. *A new way to use semidefinite programming with applications to linear equations mod p.* Journal of Algorithms, Vol. 39, 2001, pp. 162–204.

[3] S. Arora, B. Barak, and D. Steurer. *Subexponential Algorithms for Unique Games and Related problems.* Proceedings of the 51st IEEE Symposium on Foundations of Computer Science, 2010.

[4] S. Arora, R. Impagliazzo, W. Matthews, and D. Steurer. *Improved algorithms for unique games via divide and conquer.* ECCC Report TR10-041, 2010.

[5] S. Arora, S. Khot, A. Kolla, D. Steurer, M. Tulsiani, and N. Vishnoi. *Unique games on expanding constraint graphs are easy.* Proceedings of the 40th ACM Symposium on Theory of Computing, pp. 21–28, 2008.

[6] B. Barak, M. Hardt, I. Haviv, A. Rao, O. Regev and D. Steurer. *Rounding Parallel Repetitions of Unique Games*, Proceedings of the 49th IEEE Symposium on Foundations of Computer Science, pp. 374–383, 2008.

[7] A. Blum and J. Spencer. *Coloring Random and Semi-Random k-Colorable Graphs.* J. Algorithms, vol. 19, no. 2, pp. 204–234, 1995.

[8] M. Charikar, K. Makarychev, and Y. Makarychev. *Near-Optimal Algorithms for Unique Games.* Proceedings of the 38th ACM Symposium on Theory of Computing, pp. 205–214, 2006.

[9] E. Chlamtac, K. Makarychev, and Y. Makarychev. *How to Play Unique Games Using Embeddings.* Proceedings of the 47th IEEE Symposium on Foundations of Computer Science, pp. 687–696, 2006.

[10] A. Gupta and K. Talwar. *Approximating Unique Games.* Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms, pp. 99–106, 2006.

[11] U. Feige. *Relations Between Average Case Complexity and Approximation Complexity.* Proceedings of the 34th Annual ACM Symposium on Theory of Computing, pp. 534–543, 2002.

[12] U. Feige and J. Kilian. *Heuristics for Semirandom Graph Problems.* Journal of Computing and System Sciences, vol. 63, pp. 639–671, 2001.

[13] U. Feige and R. Krauthgamer. *Finding and Certifying a Large Hidden Clique in a Semi-Random Graph.* Random Structures and Algorithms, vol. 16(2), pp. 195–208, 2000.

[14] V. Guruswami, R. Manokaran, and P. Raghavendra. *Beating the Random Ordering is Hard: Inapproximability of Maximum Acyclic Subgraph.* Proceedings of the 49th IEEE Symposium on Foundations of Computer Science, pp. 573–582, 2008.

[15] V. Guruswami and P. Raghavendra. *Constraint satisfaction over a non-boolean domain: Approximation algorithms and unique-games hardness.* Proceedings of APPROX-RANDOM, 77–90, 2008.

[16] M. Jerrum. *Large Cliques Elude the Metropolis Process.* Random Structures and Algorithm, vol. 3 (4), pp. 347–359, 1992.

[17] S. Khot. *On the power of unique 2-prover 1-round games.* Proceedings of the 34th ACM Symposium on Theory of Computing, pp. 767–775, 2002.

[18] S. Khot, G. Kindler, E. Mossel, and R. O'Donnell. *Optimal inapproximability results for MAX-CUT and other 2-variable CSPs?* SIAM Journal on Computing, vol. 37(1), pp. 319–357, 2007.

[19] S. Khot and O. Regev. *Vertex cover might be hard to approximate to within* $2 - \varepsilon$. Proceedings of the 18th IEEE Annual Conference on Computational Complexity, 2003.

[20] S. Khot and N. Vishnoi. *The Unique Games Conjecture, Integrality Gap for Cut Problems and Embeddability of Negative Type Metrics into* $\ell_1$. Proceedings of IEEE Symposium on Foundations of Computer Science, pp. 53–62, 2005.

[21] A. Kolla. *Spectral Algorithms for Unique Games* Proceedings of the Conference on Computational Complexity, pp. 122–130, 2010.

[22] A. Kolla, K. Makarychev, Y. Makarychev. *How to Play Unique Games against a Semi-Random Adversary.* Preprint. ArXiv:1104.3806v1 [cs.DS].

[23] K. Makarychev and Y. Makarycev. *How to Play Unique Games on Expanders.* WAOA 2010. Lecture Notes in Computer Science, vol. 6534/2011, pp. 190–200, 2011.

[24] P. Raghavendra. *Optimal algorithms and inapproximability results for every CSP?* Proceedings of the 40th Annual ACM Symposium on Theory of Computing, pp. 245–254, 2008.

[25] P. Raghavendra and D. Steurer, *How to Round Any CSP.* Proceedings of the 50th IEEE Symposium on Foundations of Computer Science, pp. 586–594, 2009.

[26] A. Samorodnitsky and L. Trevisan. *Gowers uniformity, influence of variables, and PCPs.* Proceedings of the 38th annual ACM Symposium on Theory of Computing, pp. 11–20, 2006.

[27] L. Trevisan. *Approximation Algorithms for Unique Games.* Proceedings of the 46th IEEE Symposium on Foundations of Computer Science, pp. 197–205, 2005.