# Approximating Unique Games Using Low Diameter Graph Decomposition

Vedat Levi Alev[*]
University of Waterloo

Lap Chi Lau[†]
University of Waterloo

**Abstract**

We design approximation algorithms for Unique Games when the constraint graph admits good low diameter graph decomposition. For the $\mathsf{Max\text{-}2Lin}_k$ problem in $K_r$-minor free graphs, when there is an assignment satisfying $1 - \varepsilon$ fraction of constraints, we present an algorithm that produces an assignment satisfying $1 - O(r\varepsilon)$ fraction of constraints, with the approximation ratio independent of the alphabet size. A corollary is an improved approximation algorithm for the $\mathsf{Min\text{-}UnCut}$ problem for $K_r$-minor free graphs. For general Unique Games in $K_r$-minor free graphs, we provide another algorithm that produces an assignment satisfying $1 - O(r\sqrt{\varepsilon})$ fraction of constraints.

Our approach is to round a linear programming relaxation to find a minimum subset of edges that intersects all the inconsistent cycles. We show that it is possible to apply the low diameter graph decomposition technique on the constraint graph directly, rather than to work on the label extended graph as in previous algorithms for Unique Games. The same approach applies when the constraint graph is of genus $g$, and we get similar results with $r$ replaced by $\log g$ in the $\mathsf{Max\text{-}2Lin}_k$ problem and by $\sqrt{\log g}$ in the general problem. The former result generalizes the result of Gupta-Talwar for Unique Games in the $\mathsf{Max\text{-}2Lin}_k$ case, and the latter result generalizes the result of Trevisan for general Unique Games.

## 1 Introduction

For a given integer $k \geq 1$, an undirected graph $G = (V, E)$ and a set $\Pi = \{\pi_{uv} : uv \in E\}$ of permutations on $[k]$ satisfying $\pi_{uv} = \pi_{vu}^{-1}$, the Unique Games problem with alphabet size $k$ (denoted by $\mathsf{UG}_k$) is the problem of finding an assignment $x : V \to [k]$ to the vertices such that the number of edges $e = uv \in E$ satisfying the constraint $\pi_{uv}(x(u)) = x(v)$ is maximized. The value $\mathsf{SAT}(\mathfrak{I})$ of a Unique Games instance $\mathfrak{I} = (G, \Pi)$ is defined as,

$$\mathsf{SAT}(\mathfrak{I}) = \max_{x : V \to [k]} \frac{1}{|E|} \sum_{uv \in E} \mathbf{1}[\pi_{uv}(x(u)) = x(v)]$$

i.e. the maximum fraction of satisfiable constraints over all assignments $x$. We define $\mathsf{UNSAT}(\mathfrak{I}) = 1 - \mathsf{SAT}(\mathfrak{I})$ as the minimum fraction of unsatisfied constraints.

The Unique Games Conjecture of Khot [24] postulates that it is **NP**-hard to distinguish whether a given instance $\mathfrak{I} = (G, \Pi)$ of the Unique Games problem is almost satisfiable or almost unsatisfiable, and the problem becomes harder as the alphabet size $k$ increases.

**Conjecture 1.1** (The Unique Games Conjecture, [24]). *For every $\varepsilon > 0$, there exists an integer $k := k(\varepsilon)$, such that the decision problem of whether an instance $\mathfrak{I}$ of $\mathsf{UG}_k$ satisfies $\mathsf{SAT}(\mathfrak{I}) \geq 1 - \varepsilon$ or $\mathsf{SAT}(\mathfrak{I}) \leq \varepsilon$ is* **NP***-hard.*

The Unique Games Conjecture has attracted much attention over the years, due to its implications regarding the hardness of approximation for many **NP**-hard problems [26, 25, 32]. An important case of Unique Games is the $\mathsf{Max\text{-}2Lin}_k$ problem when the constraints are of the form $x_u - x_v \equiv c_{uv} \pmod{k}$ for $uv \in E$. This problem is shown to be as hard as the general case of the Unique Games problem by Khot et al. [25]. The $\mathsf{Max\text{-}Cut}$ problem is a well-studied special case of $\mathsf{Max\text{-}2Lin}_2$ where $x_u - x_v \equiv 1 \pmod{2}$ for $uv \in E$. Assuming the Unique Games Conjecture, Khot et al. [25] proved that it is **NP**-hard to distinguish $\mathsf{Max\text{-}Cut}$ instances where the optimal value is at least $1 - \varepsilon$ from instances where the optimal value is at most $1 - \Theta(\sqrt{\varepsilon})$.

There have been several efforts in designing polynomial time approximation algorithms for Unique Games [24, 37, 19, 12, 13], where the objective is to minimize the number of unsatisfied constraints. Let $\mathfrak{I}$ be the given instance of $\mathsf{UG}_k$ with $n$ variables and $\mathsf{UNSAT}(\mathfrak{I}) = \varepsilon$. Trevisan [37] gave an SDP-based algorithm that provides an assignment which violates at most an $\mathcal{O}(\sqrt{\varepsilon \log n})$ fraction of the constraints. Gupta and Talwar [19] gave an LP-based algorithm that provides an assignment which violates at most an $\mathcal{O}(\varepsilon \cdot \log n)$ fraction of the constraints. Charikar, Makarychev, and Makarychev [12] gave an SDP-based algorithm which finds an assignment violating at most a $\mathcal{O}(\sqrt{\varepsilon \log k})$ fraction of constraints, where $k$ is the alphabet size. Chlamtac, Makarychev, and Makarychev [13] gave another SDP-based algorithm which finds an assignment violating at most an $\mathcal{O}(\varepsilon \cdot \sqrt{\log k \log n})$-fraction of the constraints.

There are also some previous works exploiting the structures of the constraint graphs. Arora, Barak and Steurer [5] presented a subexponential time algorithm to distinguish the two cases in the Unique Games Conjecture. Their approach uses the spectral information of the constraint graph. If the Laplacian matrix of the constraint graph has only a few small eigenvalues, then they extend the subspace enumeration approach of Kolla [28] to search over this eigenspace for a good assignment. On the other hand, if there are many small eigenvalues, they give a graph decomposition procedure to delete a small fraction of edges so that each component in the remaining graph has only a few small eigenvalues. Combining these two steps carefully gives their subexponential time algorithm. There is also an SDP-based propagation rounding approach to find a good assignment when the constraint graph is an expander [6] and more generally when the Laplacian matrix of the constraint graph has only a few small eigenvalues [9, 20]. These gave an alternative SDP-based subexponential time algorithm for the Unique Games Conjecture.

Our initial motivation is to study the Unique Games problem when the Laplacian matrix of the constraint graph has many small eigenvalues, as there are no known good approximation algorithms for Unique Games in these graphs. The most natural graph family possessing this property is the class of graphs without a $K_r$ minor, where a graph $H$ is a minor of $G$ if $H$ can be obtained from $G$ by deleting and contracting edges, and $K_r$ is the complete graph with $r$ vertices. Kelner et al. [23], after a sequence of works [11, 35, 22], proved that the $k$-th smallest eigenvalue of the Laplacian matrix of a bounded degree $K_r$-minor free graph is $\mathcal{O}(\mathrm{poly}(r) \cdot k/n)$, showing that there are many small eigenvalues. The class of $K_r$-minor free graphs is well studied and is known to contain the class of planar graphs and the class of bounded genus graphs, where a graph is of genus $g$ if the graph can be embedded into a surface having at most $g$ handles without edge crossings. There are different (non-spectral) techniques in designing approximation algorithms for

various problems in $K_r$-minor free graphs (see e.g. [15, 14]), including problems that are known to be harder than Unique Games. This leads us to the question of whether we can extract those ideas to design better algorithms for Unique Games.

## 1.1 Our Results

In this paper, we consider the problem of approximately minimizing the number of unsatisfied constraints in an $\mathsf{UG}_k$ instance $\mathfrak{I} = (G, \Pi)$, when the constraint graph $G$ is $K_r$-minor free. Our first theorem is for the $\mathsf{Max\text{-}2Lin}_k$ problem.

**Theorem 1.2.** *Given a $\mathsf{Max\text{-}2Lin}_k$ instance $\mathfrak{I} = (G, \Pi)$ where $G$ is a $K_r$-minor free graph and $\mathsf{UNSAT}(\mathfrak{I}) = \varepsilon$ (respectively where $G$ is of genus at most $g$), there is an LP-based polynomial time algorithm which outputs an assignment that violates at most an $\mathcal{O}(r \cdot \varepsilon)$ fraction of constraints (respectively at most a $\mathcal{O}(\log g \cdot \varepsilon)$ fraction of constraints).*

For $\mathsf{Max\text{-}2Lin}$, Theorem 1.2 on bounded genus graphs is a refinement of the $\mathcal{O}(\log n \cdot \varepsilon)$ bound of Gupta and Talwar [19] as $g = \mathcal{O}(n)$. Theorem 1.2 also implies an improved approximation algorithm for the $\mathsf{Min\text{-}Uncut}$ problem (the complement of the $\mathsf{Max\text{-}Cut}$ problem), where the objective is to delete a minimum subset of edges so that the resulting graph is bipartite.

**Corollary 1.3.** *There is an LP-based polynomial time $O(r)$-approximation algorithm (respectively a $O(\log g)$-approximation algorithm) for the $\mathsf{Min\text{-}Uncut}$ problem for $K_r$-minor free graphs (respectively for graphs of genus $g$).*

The best known approximation algorithm for $\mathsf{Min\text{-}Uncut}$ is an SDP-based $\mathcal{O}(\sqrt{\log n})$-approximation algorithm [2, 13]. We are not aware of any improvement of this bound for $K_r$-minor free graphs and bounded genus graphs. The above algorithms crucially used the symmetry of the linear constraints in $\mathsf{Max\text{-}2Lin}$. For general Unique Games, we present a different algorithm with weaker guarantees. The following theorem on bounded genus graphs is a refinement of the $\mathcal{O}(\sqrt{\varepsilon \cdot \log n})$ bound of Trevisan [37] (see the discussion in [19, Section 4]).

**Theorem 1.4.** *Given a $\mathsf{UG}_k$ instance $\mathfrak{I} = (G, \Pi)$ where $G$ is a $K_r$-minor free graph and $\mathsf{UNSAT}(\mathfrak{I}) = \varepsilon$ (respectively where $G$ is of genus at most $g$), there is an LP-based polynomial time algorithm which outputs an assignment that violates at most an $\mathcal{O}(r \cdot \sqrt{\varepsilon})$ fraction of constraints (respectively at most a $\mathcal{O}(\sqrt{\log g \cdot \varepsilon})$ fraction of constraints).*

The main tool in our algorithms is the low diameter graph decomposition for $K_r$-minor free graphs and bounded genus graphs (see Section 2). Both of our algorithms are LP-based. The $\mathsf{Max\text{-}2Lin}_k$ algorithm is based on cutting inconsistent cycles, which is different from most existing algorithms for Unique Games that are based on finding good assignments. The $\mathsf{UG}_k$ algorithm is based on the propagation rounding method in Gupta and Talwar [19]. We defer the technical overviews to Section 3.2 and Section 5.2, after the preliminaries are defined.

## 1.2 Related Work

There are polynomial time approximation schemes for many problems in $K_r$-minor free graphs (see [14, 15]). For example, there is a $(1 - \varepsilon)$-approximation algorithm for $\mathsf{Max\text{-}Cut}$ with running time $2^{1/\varepsilon} \cdot n^{\mathcal{O}_r(1)}$ for

$K_r$-minor free graphs. The approach is a generalization of Baker's approach for planar graphs [7]: removing a small fraction of edges so that the remaining graph is of bounded treewidth, and then using dynamic programming to solve the problem on each bounded treewidth component. This approach can be used to distinguish the two cases in the Unique Games Conjecture for $K_r$-minor free graphs for any fixed $r$. However, this approach is not applicable to obtain multiplicative approximation algorithms for minimizing the number of unsatisfied constraints for Unique Games, since it requires to remove a constant fraction of edges while the optimal value could be very small. As mentioned previously, we are not aware of any polynomial time approximation algorithms with performance ratio better than $\mathcal{O}(\sqrt{\log n})$ for the Min-Uncut problem for $K_r$-minor free graphs.

The low diameter graph decomposition technique is very useful in designing approximation algorithms for $K_r$-minor free graphs. It was first developed by Bartal in [10] for probabilistically approximating general metric spaces by tree metrics. The special case of planar and minor-free graphs was studied by Klein, Plotkin and Rao [27] to establish the multicommodity flow-cut gap of $K_r$-minor free graphs, and since then this technique has found numerous applications. A recent result using this technique is a $(\mathcal{O}_\varepsilon(r^2), 1 + \varepsilon)$ bicriteria approximation algorithm [8] for the small set expansion problem, which is shown to be closely related to the Unique Games problem [33, 34].

It is a well-known result of Hadlock [21] that the maximum cut problem can be solved exactly in polynomial time on planar graphs. In Agarwal's thesis [3], he showed that an SDP relaxation (with triangle inequalities) for $\mathsf{UG}_2$ is exact for planar graphs, using a multicommodity flow-cut type argument introduced in Agarwal et al. [4]. It is mentioned in [3] that this approach of bounding the integrality gap (even approximately) is only known to work for $K_5$-minor free graphs.

Steurer and Vishnoi [36] showed that the Unique Games problem can be reduced to the Multicut problem and used it to recover Gupta and Talwar's result in the case of $\mathsf{Max\text{-}2Lin}_k$. The approach of Steurer and Vishnoi is similar to ours; see Section 3.2 for some discussion.

## 1.3 Organization

In Section 2, we describe the low diameter graph decomposition results that we will apply. In Section 3, we first present the proof for the Min-Uncut problem, as it is simpler and illustrates all the main ideas. Then we generalize the proof to the $\mathsf{Max\text{-}2Lin}_k$ problem in Section 4. In Section 5, we show the result for general Unique Games. The proof overviews for Theorem 1.2 and Theorem 1.4 will be presented in the corresponding sections, Section 3.2 and Section 5.2, after the preliminaries are defined.

# 2 Low Diameter Graph Decompositions

Let $G = (V, E)$ be a graph with non-negative edge weights $w : E \to \mathbb{R}_+$. A collection $P = \{C_1, \ldots, C_k\}$ of disjoint subsets $C_j \subseteq V$ (called clusters) is a partition if they satisfy $V = \cup_{i=1}^k C_j$. We call a partition $P$ weakly $\Delta$-bounded if each of the clusters has weak diameter $\Delta$, i.e.

$$d_G(u, v) \leq \Delta \quad \forall u, v \in C_j; \forall j \in [k]$$

where $d_G$ denotes the shortest path distance on $G$ (induced by the edge weights $w$). We say that the partition $P$ is strongly $\Delta$-bounded if each cluster has strong diameter $\Delta$, i.e.

$$d_{G[C_j]}(u,v) \leq \Delta \quad \forall u,v \in C_j; \forall j \in [k]$$

where $d_{G[C_j]}$ denotes the shortest path distance in the induced subgraph $G[C_j]$. We write $P(u)$ for the unique cluster $C_j$ containing the vertex $u \in V$. We call a distribution $\mathcal{A}$ of partitions $\Delta$-bounded $D$-separating if each cluster is of diameter $\Delta$ and for each edge $uv \in E$ we have

$$\mathbb{P}_{P \sim \mathcal{A}}[P(u) \neq P(v)] \leq \frac{D}{\Delta} \cdot w(u,v). \tag{2.1}$$

This implies that we can cut a graph into clusters with diameter at most $\Delta$ by deleting all the inter-cluster edges, while only losing a $D/\Delta$ fraction of the total edge weight. We call a $\Delta$-bounded $D$-separating partitioning scheme efficient, if we can sample it in polynomial time.

The seminal work of Klein, Plotkin and Rao [27] showed the first low diameter graph decomposition scheme for planar graphs and more generally for $K_r$-minor free graphs. We will use the latest result of this line of work [27, 17, 30, 1], as it gives the best known quantitative bound and also it guarantees the clusters have strong diameter $\Delta$ which will be important in our algorithm for general Unique Games.

**Theorem 2.1** ([1]). *Every weighted $K_r$-minor free graph admits an efficient weakly $\Delta$-bounded $\mathcal{O}(r)$-separating partitioning scheme for any $\Delta \geq 0$.*

**Theorem 2.2** ([1]). *Every weighted $K_r$-minor free graph admits an efficient strongly $\Delta$-bounded $\mathcal{O}(r^2)$-separating partitioning scheme for any $\Delta \geq 0$.*

We will also use the optimal bounds for bounded genus graphs, to derive better results for Unique Games in these graphs.

**Theorem 2.3** ([1, 30]). *Every weighted graph of genus $g$ admits an efficient strongly $\Delta$-bounded $\mathcal{O}(\log g)$-separating partitioning scheme for any $\Delta \geq 0$.*

The results in [1] are stated using the language of padded decompositions, but it is easy to see that the results we stated are corollaries of the theorems in [1].

# 3  Minimum Uncut

Given an undirected graph $G = (V, E)$ with a non-negative cost $c_e$ for each edge $e \in E$, the Min-Uncut problem is to find a subset $S \subseteq V$ to minimize the total cost of the uncut edges (the edges with both endpoints in $S$ or both endpoints in $V - S$). Alternatively, the problem is equivalent to finding a subset $F \subseteq E$ of minimum total cost so that $G - F$ is a bipartite graph (so $F$ is the uncut edges). As a graph is bipartite if and only if it has no odd cycles, the problem is equivalent to finding a subset of edges of minimum total cost that intersects all the odd cycles in the graph, which is also known as the Odd Cycle Transversal problem. We will tackle the Min-Uncut problem using this perspective, by writing a linear program for the Odd Cycle Transversal problem.

As mentioned already, the Min-Uncut problem is a special case of Max-2Lin$_2$. We will see in Section 4 that the ideas in this section can be readily generalized to design an approximation algorithm for the Max-2Lin$_k$ problem.

## 3.1 Linear Programming Relaxation

We consider the following well-known linear programming relaxation for the Odd Cycle Transversal problem, which is known to be exact when the input is a planar graph [18]. We note that this is similar to the LP formulation used by Gupta and Talwar [19] when specialized to the Min-Uncut problem, but their LP formulation is on the "label extended graph" that we will explain soon.

$$\mathsf{LP}^\star = \min \sum_{e \in E} c_e x_e \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\text{(LP-MinUncut)}$$

subject to

$$\sum_{e \in C} x_e \geq 1 \quad C \in \mathcal{C}$$

$$x_e \geq 0 \quad e \in E$$

where $\mathcal{C}$ is the set of odd cycles of $G$.

This LP has exponentially many constraints. To solve it in polynomial time using the ellipsoid method [31], we require a polynomial time separation oracle to check whether a solution $x$ is feasible or not, and if not provide a violating constraint. For this LP, it is well known that the separation oracle can be implemented in polynomial time using shortest path computations (e.g. see [19]). Since this will be relevant to our discussion, we describe the separation oracle in the following.

The idea is to construct the "label extended graph" $H = (V', E')$ of $G = (V, E)$ (to use the Unique Games terminology). For each vertex $v$ in $V$, we create two vertices $v^+$ and $v^-$ in $V'$. For each edge $uv$ in $E$, we add two edges $u^+v^-$ and $u^-v^+$ in $E'$, and we set the weight of $u^+v^-$ and $u^-v^+$ to be $x_{uv}$. By construction, there is an odd cycle in $G$ containing $v$ if and only if there is a path from $v^+$ to $v^-$ in the label extended graph $H$. So, to check that $x$ is feasible, we just need to check that the weight of the shortest path from $v^+$ to $v^-$ is at least one for every $v$.

## 3.2 Proof Overview

One natural approach to do the rounding is to consider the label extended graph $H$ of $G$. From the above discussion, destroying all the odd cycles in $G$ is equivalent to destroying all the $v^+$-$v^-$ paths in $H$ for all $v$. Since $x$ is feasible, we know that the shortest path distance between $v^+$ and $v^-$ is at least 1 for every $v$. Therefore, we can apply the low diameter graph decomposition result in the label extended graph, by setting $\Delta < 1$ to ensure that all $v^+$ and $v^-$ are disconnected, and hope to delete edges with weight at most $\sum_{e \in E} O(r/\Delta) \cdot c_e x_e = O(r) \cdot \mathsf{LP}^\star$ by Theorem 2.1. This is similar to the approach used in [36] to reduce Unique Games to Multicut. The problem of this approach is that the label extended graph $H$ could have arbitrarily large clique minor, even though the original constraint graph $G$ is $K_r$-minor free: even if the constraint graph $G$ is grid-like and planar, the label-extended graph $H$ can contain a $K_{\Omega(n)}$ minor, even when the alphabet size is just two. This means that applying the theorems in Section 2 blindly does not give better than a $\mathcal{O}(\log n)$-factor approximation.

This is often a technical issue in analyzing algorithms for Unique Games: It is most natural to work on

(a) The shortest path distance between any two pairs of vertices is 0. The bold edges correspond to an optimal integral solution to LP-MinUncut.

(b) After removing edges with weight at least $1/2$ (the bold edges), all remaining subgraphs are of diameter at most $1/4$ and they are bi-partite. The dashed edges are the inter-cluster edges.

Figure 1: Applying low diameter graph decomposition in a feasible solution to LP-MinUncut.

the label extended graph but the label extended graph does not necessarily share the nice properties in the original graph [28]. It is not obvious how to apply low diameter graph decomposition directly in the original constraint graph to do the rounding. For example, in the graph shown in Figure 1a, $x$ is an integral solution but the shortest path distance (using $x_e$ as the edge weight of $e$) is 0 for all pairs of vertices, providing no useful information about which pairs of vertices we need to separate.

Our main observation is that the shortest path distances are not useful only when there are edges with large $x_e$. In Lemma 3.2, we prove that if $x_e < 1/2$ for every $e$, then every odd cycle contains a pair of vertices $u, v$ with shortest path distance greater than $1/4$ (using $x_e$ as the edge weight of $e$). Therefore, if we apply low diameter graph decomposition with $\Delta = 1/4$, then we can ensure that no odd cycle will remain in any cluster, and the above calculation shows that the total weight of the deleted edges is $\mathcal{O}(r) \cdot \mathsf{LP}^\star$. To reduce to the case where there are no edges with $x_e \geq 1/2$, we can simply delete all such edges as their total weight is at most $2\mathsf{LP}^\star$. This preprocessing step is remotely similar to some iterative rounding algorithms (see [29]). See Figure 1b for an illustration.

## 3.3 Rounding Algorithm

---

**Algorithm 3.1** (Min-Uncut).

  **Intput**    A feasible solution $x$ to LP-MinUncut with value $\mathsf{LP}^\star$ on a $K_r$-minor free graph.

  **Output**    An integral solution to LP-MinUncut with total cost $O(r) \cdot \mathsf{LP}^\star$

1. Let $F_1$ be the subset of edges with $x_e \geq 1/2$. Delete all edges in $F_1$ from the graph.

2. Set the weight $w_e$ of each edge $e$ in the remaining graph to be $x_e$.
   Sample a weakly $(1/4)$-bounded $\mathcal{O}(r)$-separating partition $P$ guaranteed by Theorem 2.1 in the remaining graph.

3. Let $F_2$ be the set of inter-cluster edges in $P$, i.e. edges $uv$ with $P(u) \neq P(v)$.
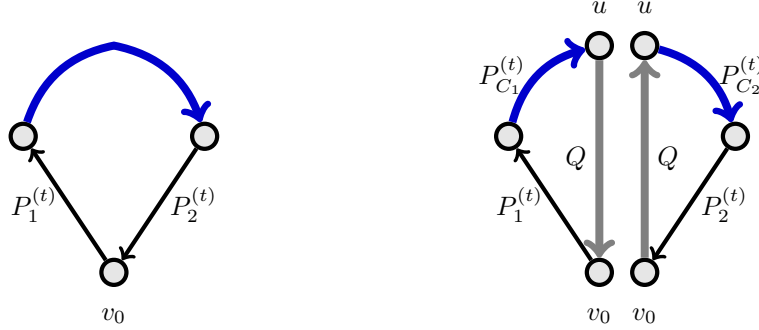   Return $F_1 \cup F_2$ as the output.

---

Figure 2: The paths involved in the proof of Lemma 3.2. The shortcut $Q$ is highlighted gray, and the cycle segments $P_{C_j}^{(t)}$ are highlighted blue. Since the walk we maintain is odd, one of the two walks we consider in the induction step (right figure) should be odd.

## 3.4 Main Lemma

The following lemma allows us to apply low diameter graph decomposition in the original constraint graph. The proof uses the simple but crucial fact that if we "shortcut" an odd cycle, one of the two cycles created is an odd cycle.

**Lemma 3.2.** *Let $G'$ be a graph with edge weight $x_e$ for each edge $e$. Suppose every odd cycle $C$ has total weight at least 1, i.e. $\sum_{e \in C} x_e \geq 1$. If $0 \leq x_e < \delta \leq 1$ for every edge $e \in G'$, then every odd cycle $C$ in $G'$ contains a pair of vertices $u, v$ satisfying $d_x(u, v) > (1 - \delta)/2$, where $d_x(u, v)$ denotes the shortest path distance from $u$ to $v$ induced by the edge weights $x_e$.*

*Proof.* Let $C$ be an arbitrary odd cycle and let $v_0$ be an arbitrary vertex in $C$. We will prove the stronger statement that if $d_x(v_0, v) \leq (1 - \delta)/2$ for every $v \in C$, then there is an edge $e \in C$ with $x_e \geq \delta$. Note that the contrapositive of this stronger statement clearly implies the lemma.

Since all odd cycles have total weight at least 1, any nontrivial odd walk (may visit some vertices multiple times) from $v_0$ to $v_0$ has total weight at least 1. This is because any odd walk can be decomposed into edge-disjoint simple cycles, with at least one of which is odd.

We will prove the statement by an inductive argument. In a general inductive step $t \geq 0$, we maintain a walk $C^{(t)}$ from $v_0$ to $v_0$ satisfying the following properties (see Figure 2):

1. $C^{(t)}$ is a nontrivial odd walk from $v_0$ to $v_0$, consisting of three paths $P_1^{(t)}$-$P_C^{(t)}$-$P_2^{(t)}$.

2. $P_1^{(t)}$ and $P_2^{(t)}$ contain $v_0$, with $v_0$ being the first vertex of $P_1^{(t)}$ and $v_0$ being the last vertex of $P_2^{(t)}$.

3. Both $P_1^{(t)}$ and $P_2^{(t)}$ have total weight at most $(1 - \delta)/2$.

4. $P_C^{(t)}$ is a continuous segment of $C$, i.e. if $C = (v_0, v_1, \ldots, v_k = v_0)$, then $P_C^{(t)} = (v_i, \ldots, v_j)$ for some $0 \leq i < j \leq k$. In particular, $P_C^{(t)} \neq \emptyset$.

Initially, $C^{(0)}$ is just the cycle $C$, with $P_1^{(0)} = P_2^{(0)} = \emptyset$ and $P_C^{(0)} = C$.

8

Let $w(P)$ denote the total weight of a path $P$, and let $|P|$ denote the number of edges in $P$. Since $w(P_1^{(t)}), w(P_2^{(t)}) \leq (1 - \delta)/2$, we must have $w(P_C^{(t)}) \geq \delta$, as $C^{(t)}$ is a nontrivial odd walk and thus the total weight is at least one. The inductive step is to show that if $d_x(v_0, v) \leq (1 - \delta)/2$ for all $v \in C$, then we can construct $C^{(t+1)}$ from $C^{(t)}$ so that $C^{(t+1)}$ still satisfies the properties but $|P_C^{(t+1)}| < |P_C^{(t)}|$. By applying this inductively, we will eventually construct a walk $C^{(T)}$ that satisfies the properties and $|P_C^{(T)}| = 1$, and so $P_C^{(T)}$ is an edge of $C$ with weight $w(P_C^{(t)}) \geq \delta$, and this will complete the proof.

It remains to prove the inductive step (see Figure 2). Let $C^{(t)}$ be a walk that satisfies the properties but $|P_C^{(t)}| \geq 2$. Let $u$ be an internal vertex of $P_C^{(t)}$, which splits $P_C^{(t)}$ into $P_{C_1}^{(t)}$ and $P_{C_2}^{(t)}$, so that the walk $C^{(t)}$ consists of $P_1^{(t)}$-$P_{C_1}^{(t)}$-$P_{C_2}^{(t)}$-$P_2^{(t)}$. Since $d_x(v_0, u) \leq (1 - \delta)/2$, there is a path $Q$ from $v_0$ to $u$ with $w(Q) \leq (1 - \delta)/2$. The path $Q$ splits the walk $C^{(t)}$ into two walks, $P_1^{(t)}$-$P_{C_1}^{(t)}$-$Q$ and $Q$-$P_{C_2}^{(t)}$-$P_2^{(t)}$ As $C^{(t)}$ is an odd walk, a simple parity argument implies that exactly one of these two walks must be odd, say $P_1^{(t)}$-$P_{C_1}^{(t)}$-$Q$ (the other case is similar). Then we let $C^{(t+1)} := P_1^{(t)}$-$P_{C_1}^{(t)}$-$Q$, with $P_1^{(t+1)} := P_1^{(t)}$, $P_C^{(t+1)} := P_{C_1}^{(t)}$, and $P_1^{(t+1)} := Q$. It is straightforward to check that $C^{(t+1)}$ still satisfy all the properties and furthermore $|P_C^{(t+1)}| < |P_C^{(t)}|$, completing the proof of the induction step. $\square$

## 3.5 Proof of Corollary 1.3

We are now ready to prove that the algorithm in Section 3.3 is an $O(r)$-approximation algorithm for Min-Uncut. In step 1, since each edge $e$ in $F_1$ has $x_e \geq 1/2$, the total cost of edges in $F_1$ is

$$\sum_{e \in F_1} c_e \leq 2 \sum_{e \in F_1} c_e x_e \leq 2\mathsf{LP}^\star.$$

Let $G' := G - F_1$ be the remaining graph. By Lemma 3.2, every odd cycle of $G'$ contains a pair of vertices $u, v$ with shortest path distance greater than $1/4$. Let $\Delta = 1/4$. In a $(1/4)$-bounded partition $P$, no cluster can contain an odd cycle $C$ as otherwise the pair of vertices $u, v \in C$ with $d_x(u, v) > 1/4$ guaranteed by Lemma 3.2 would contradict that the cluster has weak diameter at most $1/4$. So, each cluster induces a bipartite graph, and thus $G' - F_2$ is a bipartite graph where $F_2$ is the set of inter-cluster edges. Therefore, $F_1 \cup F_2$ is an integral solution to the Odd Cycle Transversal problem, and hence an integral solution to the Min-Uncut problem.

To complete the proof, it remains to bound the cost of the edges in $F_2$. We use Theorem 2.1 to sample from a distribution of partitions which is $\Delta$-bounded and $\mathcal{O}(r)$-separating, and by definition (2.1) the probability of an edge $e$ being an inter-cluster edge is at most $\mathcal{O}(r) \cdot x_e/\Delta = \mathcal{O}(r) \cdot x_e$. Therefore, the expected cost of $F_2$ is

$$\mathbb{E}\left[\sum_{e \in F_2} c_e\right] = \sum_{e = uv \in G'} c_e \cdot \mathbb{P}_{P \sim \mathcal{A}}[P(u) \neq P(v)] = \sum_{e \in G'} c_e \cdot \mathcal{O}(r) \cdot x_e = \mathcal{O}(r) \sum_{e \in G'} c_e x_e \leq \mathcal{O}(r) \cdot \mathsf{LP}^\star.$$

Hence, the expected total cost of edges in $F_1 \cup F_2$ is $\mathcal{O}(r) \cdot \mathsf{LP}^\star$, and this concludes the proof of Corollary 1.3 about $K_r$-minor free graphs. For bounded genus graphs, the proof is the same except that we use Theorem 2.3 which guarantees the partitioning scheme is $\mathcal{O}(\log g)$-separating.

# 4 Max-2Lin$_k$

In this section, we show that the Min-Uncut algorithm can be readily generalized to the Max-2Lin$_k$ problem. The proofs will be almost identical, so we just highlight the subtle differences.

One important feature of Theorem 1.2 is that the approximation ratio does not depend on the alphabet size. The reason is that the symmetry of the linear constraints allows us to define inconsistent cycles in the original constraint graph, which will play the same role as the odd cycles in the Min-Uncut problem. This allows us to reduce Max-2Lin$_k$ to the Inconsistent Cycle Transversal problem.

## 4.1 Problem Formulation

Consider the Max-2Lin$_k$ problem where each constraint is of the form $x_u - x_v = c_{uv} \pmod{k}$ where $c_{uv} \in \mathbb{Z}_k$. The symmetry property that we will exploit is that every permutation constraint $\pi_{uv}$ satisfies: $\pi_{uv}(i + c) = \pi_{uv}(i) + c$ for all $i, c \in \mathbb{Z}_k$. Note that there are "directions" in the constraints, as $\pi_{uv} = (\pi_{vu})^{-1}$ and they are in general different. In the Max-Cut (or Min-Uncut) problems, we have $\pi_{uv} = \pi_{vu}$ as the alphabet set is of size two, and so the concept of direction was not discussed.

**Definition 4.1** (Inconsistent cycles for Max-2Lin$_k$)**.** Let $\mathfrak{I} = (G, \Pi)$ be an instance of Max-2Lin$_k$. A cycle $(v_0, v_1, \ldots, v_l = v_0)$ of length $l$ in $G$ is called inconsistent if

$$\pi_{v_l v_{l-1}} \circ \pi_{v_{l-1} v_{l-2}} \circ \cdots \circ \pi_{v_1 v_0} \neq \mathrm{Id} \tag{4.2}$$

where Id is the identity permutation. By the aforementioned symmetry property of Max-2Lin$_k$, if the product $\pi$ of permutation constraints along a cycle is not the identity permutation, then $\pi(i) \neq i$ for all $i \in \mathbb{Z}_k$. This is the crucial property that we will use.

The following lemma shows that Max-2Lin$_k$ is equivalent to the Inconsistent Cycle Transversal problem. The reason is that whether a cycle is satisfiable is independent of which label to assign to the starting vertex because of the symmetry property. Note that this does not hold for general Unique Games.

**Lemma 4.2.** *A* Max-2Lin$_k$ *instance* $\mathfrak{I} = (G, \Pi)$ *is satisfiable if and only if* $G$ *contains no inconsistent cycles.*

*Proof.* Suppose $\mathfrak{I}$ is satisfiable. Let $x$ be a satisfying assignment. Consider an arbitrary cycle $C = (v_0, v_1, \ldots, v_l = v_0)$. The permutation constraints on $C$ enforce that $\pi_{v_l v_{l-1}} \circ \pi_{v_{l-1} v_{l-2}} \circ \cdots \circ \pi_{v_1 v_0}(x(v_0)) = x(v_0)$ where $x(v_0)$ is the value of $v_0$ in the assignment $x$. By the symmetry property of the constraints, this implies that $\pi_{v_l v_{l-1}} \circ \pi_{v_{l-1} v_{l-2}} \circ \cdots \circ \pi_{v_1 v_0}$ is the identity permutation, and thus it is consistent.

Suppose $G$ has no inconsistent cycles. Then we show that $G$ is satisfiable by the following trivial algorithm. Pick an arbitrary vertex $v_0 \in G$, and set $x(v_0)$ an arbitrary value. Then we propagate this assignment to every other vertex $v$ by using an arbitrary path $P = (v_0, v_1, \ldots, v_l = v)$ from $v_0$ to $v$ and set $x(v) = \pi_{v_l v_{l-1}} \circ \pi_{v_{l-1} v_{l-2}} \circ \cdots \circ \pi_{v_1 v_0}(v_0)$. In particular, we can use a breadth first search tree to propagate the assignment. Since $G$ has no inconsistent cycles, any two paths $P_1, P_2$ from $v_0$ to $v$ will define the same value $x(v)$, as otherwise following $P_1$ from $v_0$ to $v$ and following $P_2$ from $v$ to $v_0$ will give us an inconsistent cycle. This implies that any non-tree constraint $uv$ is also satisfied by the assignment, as otherwise it means that there are two paths from $v_0$ to $v$ defining different values from $x(v)$, one path being the tree path from $v_0$ to $u$ plus the edge $uv$, and the other path being the tree path from $v_0$ to $v$. $\qquad\square$

## 4.2 Linear Programming Relaxation

Given Lemma 4.2, we can formulate the minimization version of the Max-2Lin$_k$ problem, the Min-2Lin$_k$ problem, as the Inconsistent Cycle Transversal problem, where the objective is to find a subset of edges of minimum cost that intersects all the inconsistent cycles. We can then use the same linear programming relaxation for the Min-Uncut problem, with $\mathcal{C}$ being the set of inconsistent cycles in the constraint graph. Again, we can design a polynomial time separation oracle to check whether a solution $x$ is feasible, by constructing the label extended graph and using shortest path computations as in Section 3.1 (see [19]).

## 4.3 Rounding Algorithm and Analysis

The rounding algorithm is exactly the same as in Section 3.3, and so we do not repeat it here. The analysis is also the same, which relies on a generalization of Lemma 3.2.

**Lemma 4.3.** *Let $G'$ be a graph with edge weight $x_e$ for each edge $e$. Suppose every inconsistent cycle $C$ has total weight at least $1$, i.e. $\sum_{e \in C} x_e \geq 1$. If $0 \leq x_e < \delta \leq 1$ for every edge $e \in G'$, then every inconsistent cycle $C$ in $G'$ contains a pair of vertices $u, v$ satisfying $d_x(u, v) > (1 - \delta)/2$, where $d_x(u, v)$ denotes the shortest path distance from $u$ to $v$ induced by the edge weights $x_e$.*

*Proof.* The proof is essentially identical, by replacing every occurrence of "odd" by "inconsistent". The only place that needs explanation is in the last paragraph of Lemma 3.2, when we split an inconsistent walk using a path $Q$ from $v_0$ to $u$ into two walks $P_1^{(t)}$-$P_{C_1}^{(t)}$-$Q$ and $Q$-$P_{C_2}^{(t)}$-$P_2^{(t)}$, and we need to argue that at least one of these two walks is inconsistent. Suppose both walks are consistent. Let $\pi_{P_1}$ be the composition of the permutation constraints from $v_0$ to $u$ following the path $P_1^{(t)}$-$P_{C_1}^{(t)}$, $\pi_Q$ be the composition of the permutation constraints from $u$ to $v_0$ following the path $Q$, and $\pi_{P_2}$ be the composition of the permutation constraints from $u$ to $v_0$ following the path $P_{C_2}^{(t)}$-$P_2^{(t)}$. The first walk is consistent means that $\pi_Q \circ \pi_{P_1} = \mathrm{Id}$, and the second walk is consistent means that $\pi_{P_2} \circ (\pi_Q)^{-1} = \mathrm{Id}$. But this implies that following the first walk and then the second walk is consistent, and thus the original walk is also consistent as $\mathrm{Id} = (\pi_{P_2} \circ (\pi_Q)^{-1}) \circ (\pi_Q \circ \pi_{P_1}) = \pi_{P_2} \circ \pi_{P_1}$, contradicting that the original walk is inconsistent. The rest of the proof is identical. □

With Lemma 4.3, using exactly the same argument as in Section 3.5 gives us the proof of Theorem 1.2.

# 5 General Unique Games

For general Unique Games, we could not reduce the problem to some cycle cutting problem in the original constraint graph. Instead, we modify the LP-based algorithm of Gupta and Talwar [19] to prove Theorem 1.4.

## 5.1 Linear Programming Relaxation

Gupta and Talwar [19] use the following linear programming relaxation for the Unique Games problem.

$$\min \mathsf{LP}^\star = \sum_{uv \in E} \frac{c_{uv}}{2} \sum_{l=1}^{k} d(u,v,l) \tag{LP-UG}$$

subject to

$$\sum_{l=1}^{k} x(u,l) = 1 \qquad\qquad \forall u \in V$$

$$d(u,v,l) \geq |x(u,l) - x(v, \pi_{uv}(l))| \qquad\qquad \forall uv \in E,\ l \in [k]$$

$$\sum_{i=1}^{t} d(v_{i-1}, v_i, l_{i-1}) \geq x(u, l_0) \qquad\qquad \forall C,\ \forall u \in C,\ \forall l_0 \in B_{u,C}$$

$$1 \geq x(u,l) \geq 0 \qquad\qquad \forall u \in V,\ \forall l \in [k]$$

The intended value of $x(u,l)$ is 1 if we assign the label $l$ to vertex $u$ and 0 otherwise, and so the first constraint enforces that we assign exactly one label to each vertex. The intended value of $d(u,v,l)$ is 1 if we assign $u$ to $l$ but not assign $v$ to $\pi_{uv}(l)$ or vice versa and is 0 otherwise. So $\sum_{l=1}^{k} d(u,v,l)$ is two if the constraint $\pi_{uv}$ is not satisfied and is 0 if the constraint is satisfied, and therefore the objective function is to minimize the total cost of the violated constraints. The third constraint is the inconsistent cycle constraint in the label extended graph: $B_{u,C}$ is defined as the set of "bad" labels at $u$, so that if $u$ is assigned some label in $B_{u,C}$, then propagating this label along the cycle must violate some permutation constraint in $C$. So, the intention of the third constraint is that if we assign some label in $B_{u,C}$ to vertex $u$, then the number of violated constraint along the cycle $C$ must be at least 1. This is similar to our inconsistent cycle constraint, but defined on the label extended graph.

## 5.2 Proof Overview

Gupta and Talwar [19] gave a polynomial time randomized algorithm to return an integral solution of cost $\mathcal{O}(\log n) \cdot \mathsf{LP}^\star$ from a feasible solution to the LP with objective value $\mathsf{LP}^\star$.

The main technique in their rounding algorithm is the use of a low average distortion tree to propagate an assignment from a vertex. Their propagation rounding algorithm picks an arbitrary vertex $u \in V$ and assigns it a random label $l_u$ according to the probability distribution defined by $x(u,l)$. Then they design a correlated sampling scheme to sample a label $l_v$ for a neighbor $v$ of $u$ satisfying the properties that $\mathbb{P}[l_v = l] = x(v,l)$ and $\mathbb{P}[l_v \neq \pi_{uv}(l_u)] \leq \sum_{l=1}^{k} d(u,v,l)$. They use this correlated sampling to propagate the assignment from the starting vertex to every vertex in the graph using the low average distortion tree. Their approximation ratio comes from the average distortion $\mathcal{O}(\log n)$ of the tree given by the FRT embedding [16], which can not be improved even for planar graphs.

We will still use the propagation rounding method of Gupta and Talwar, but we apply it to different trees. In [19], the tree $T$ needs not be a spanning tree in the constraint graph (i.e. some edges in the tree may not exist in the graph), and this adds some complication to the analysis. In our application, all tree edges will be graph edges and we can use a simpler lemma in their proof. For an edge $uv \in E$, we let $d_G(u,v) := \sum_{l=1}^{k} d(u,v,l)$, and let $d_T(u,v) := \sum_{xy \in P} d_G(x,y)$ where $P$ is the unique path from $u$ to $v$ in the tree $T$.

**Lemma 5.1** (Lemma 3.1 in [19])**.** *Let $x$ be the assignment produced by the propagation rounding algorithm using correlated sampling along a tree $T$. For every edge $uv \in G$, we have*

$$\mathbb{P}[x(v) \neq \pi_{uv}(x(u))] \leq d_G(u, v) + 2d_T(u, v).$$

The idea of our algorithm is very simple. We use the strongly $\Delta$-bounded $\mathcal{O}(r^2)$-separating partitioning scheme to decompose the graph, using $d_G(u, v)$ as the weight of edge $uv \in E(G)$. As each cluster is of strong diameter $\Delta$, we simply use a shortest path tree in each cluster to do the propagation rounding and apply Lemma 5.1 to prove Theorem 1.4. We will choose $\Delta$ to balance the losses in the two steps.

## 5.3 Rounding Algorithm

---

**Algorithm 5.2** ($\mathsf{UG}_k$)**.**

   **Intput**   A feasible solution $x, d$ to LP-UG with value $\mathsf{LP}^\star$ on a $K_r$-minor free graph.

   **Output**   An integral solution to LP-UG with total cost $O(r) \cdot \sqrt{\mathsf{LP}^\star}$.

1. Set the weight $w_{uv}$ of each edge $uv$ to be $d_G(u, v)$.
   Sample a strongly $\Delta$-bounded $\mathcal{O}(r^2)$-separating partition $P$ guaranteed by Theorem 2.2.

2. Let $F$ be the set of inter-cluster edges in $P$, i.e. edges $uv$ with $P(u) \neq P(v)$.
   Delete $F$ from $G$.

3. In each cluster $C_j$ in the remaining graph, compute a shortest path tree $T_j$.

4. Run Gupta-Talwar propagation rounding on each cluster $C_j$ using tree $T_j$.

5. Return the solution $x, d$ as the union of the solution in each cluster.

---

## 5.4 Proof of Theorem 1.4

Since the partitioning scheme is $\mathcal{O}(r^2)$-separating, by definition (2.1), each edge $e$ is deleted with probability

$$\mathbb{P}[\text{edge } uv \text{ is deleted}] = \mathcal{O}(r^2) \cdot \frac{d_G(u, v)}{\Delta}.$$

Hence, the expected total cost of the deleted edges in Step 2 is

$$\sum_{uv \in E} c_{uv} \cdot \mathbb{P}[\text{edge } uv \text{ is deleted}] = \mathcal{O}(r^2/\Delta) \sum_{uv \in E} c_{uv} \cdot d_G(u, v) = \mathcal{O}(r^2/\Delta) \cdot \mathsf{LP}^\star.$$

We just assume that all of these edges will be violated by the assignment we produce at the end. Since each cluster $C_j$ has strong diameter $\Delta$, the shortest path tree $T_j$ satisfies

$$d_{T_j}(u, v) \leq \Delta \quad \forall u, v \in C_j.$$

Using the Gupta-Talwar propagation rounding, by Lemma 5.1, each edge in cluster $C_j$ is violated with probability $O(\Delta)$, and therefore the total cost of the violating constraints in the Step 4 is at most $O(\Delta) \sum_{e \in E} c_e$. By choosing $\Delta = r \cdot \sqrt{\mathsf{LP}^\star / \sum_{e \in E} c_e}$, the total cost of the violating constraints is at most $r \cdot \sqrt{\mathsf{LP}^\star \cdot \sum_{e \in E} c_e}$. When $\mathsf{LP}^\star = \varepsilon \cdot \sum_{e \in E} c_e$, the total cost of the violating constraint is at most $r\sqrt{\varepsilon} \sum_{e \in E} c_e$, proving Theorem 1.4 for $K_r$-minor free graphs. For bounded genus graphs, we just use the bound in Theorem 2.3 to replace $r^2$ by $\log g$, and the same proof gives Theorem 1.4 for bounded genus graphs.

# 6 Discussions and Open Problems

The algorithm for general Unique Games has a similar structure to the subexponential time algorithm [5]. Both algorithms first deletes a small fraction of edges so that each remaining component has some nice properties, and then solve the problem in each component using a propagation rounding method. The nice property in [5] is that each component has few small eigenvalues (which qualitatively means that the components have good expansion property), and the decomposition result is based on random walks. The nice property in this paper is that each component has small diameter, and the decomposition result is based on some combinatorial methods. The key to these algorithms is some graph decomposition result. Is there some property that captures both good expansion and small diameter so that graph decomposition is still possible? Is there some property that captures both good expansion and small diameter so that propagation rounding still works?

Another open question is whether the ideas in this paper can be generalized to handle graphs with many small eigenvalues.

# References

[1] Ittai Abraham, Cyril Gavoille, Anupam Gupta, Ofer Neiman, and Kunal Talwar. Cops, robbers, and threatening skeletons: Padded decomposition for minor-free graphs. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*. ACM, 2014.

[2] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. O(sqrt(log n)) approximation algorithms for min uncut, min 2cnf deletion, and directed cut problems. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*. ACM, 2005.

[3] Naman Agarwal. Unique games conjecture: The boolean hypercube and connections to graph lifts. Master's thesis, University of Illinois at Urbana-Champaign, 2014.

[4] Naman Agarwal, Guy Kindler, Alexandra Kolla, and Luca Trevisan. Unique games on the hypercube. *Chicago Journal of Theoretical Computer Science*, 2015, 2015.

[5] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2010.

[6] Sanjeev Arora, Subhash A Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K Vishnoi. Unique games on expanding constraint graphs are easy. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*. ACM, 2008.

[7] Brenda S Baker. Approximation algorithms for np-complete problems on planar graphs. *Journal of the ACM*, 41(1), 1994.

[8] Nikhil Bansal, Uriel Feige, Robert Krauthgamer, Konstantin Makarychev, Viswanath Nagarajan, Joseph Naor, and Roy Schwartz. Min-max graph partitioning and small set expansion. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science.* IEEE, 2011.

[9] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science.* IEEE, 2011.

[10] Yair Bartal. Probablistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37-th Annual IEEE Symposium on Foundations of Computer Science.* IEEE, 1996.

[11] Punyashloka Biswal, James R Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *Journal of the ACM*, 57(3), 2010.

[12] 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.* ACM, 2006.

[13] Eden Chlamtac, Konstantin Makarychev, and Yury Makarychev. How to play unique games using embeddings. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science.* IEEE, 2006.

[14] Erik D Demaine, Mohammad Taghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic graph minor theory: Decomposition, approximation, and coloring. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science.* IEEE, 2005.

[15] Erik D Demaine and MohammadTaghi Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, 51(3), 2008.

[16] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing.* ACM, 2003.

[17] Jittat Fakcharoenphol and Kunal Talwar. An improved decomposition theorem for graphs excluding a fixed minor. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques.* Springer Berlin Heidelberg, 2003.

[18] Jean Fonlupt, Ali Ridha Mahjoub, and JP Uhry. Compositions in the bipartite subgraph polytope. *Discrete mathematics*, 105(1-3), 1992.

[19] Anupam Gupta and Kunal Talwar. Approximating unique games. In *Proceedings of the 70th Annual ACM-SIAM Symposium on Discrete Algorithms.* SIAM, 2006.

[20] Venkatesan Guruswami and Ali Kemal Sinop. Lasserre hierarchy, higher eigenvalues, and approximation schemes for graph partitioning and quadratic integer programming with psd objectives. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science.* IEEE, 2011.

[21] Frank Hadlock. Finding a maximum cut of a planar graph in polynomial time. *SIAM Journal on Computing*, 4(3), 1975.

[22] Jonathan A Kelner. Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus. *SIAM Journal on Computing*, 35(4), 2006.

[23] Jonathan A Kelner, James R Lee, Gregory N Price, and Shang-Hua Teng. Metric uniformization and spectral bounds for graphs. *Geometric and Functional Analysis*, 21(5), 2011.

[24] Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings of the 34th Annual ACM Symposium on Theory of computing*. ACM, 2002.

[25] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for max-cut and other 2-variable csps? *SIAM Journal on Computing*, 37(1), 2007.

[26] Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-$\varepsilon$. *Journal of Computer and System Sciences*, 74(3), 2008.

[27] Philip Klein, Serge A Plotkin, and Satish Rao. Excluded minors, network decomposition, and multi-commodity flow. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*. ACM, 1993.

[28] Alexandra Kolla. Spectral algorithms for unique games. *Computational Complexity*, 20(2), 2011.

[29] Lap Chi Lau, Ramamoorthi Ravi, and Mohit Singh. *Iterative methods in combinatorial optimization*, volume 46. Cambridge University Press, 2011.

[30] James R Lee and Anastasios Sidiropoulos. Genus and the geometry of the cut graph In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2010.

[31] László Lovász, Martin Grötschel, and Alexander Schrijver. Geometric algorithms and combinatorial optimization. *Berlin: Springer-Verlag*, 33, 1988.

[32] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*. ACM, 2008.

[33] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*. ACM, 2010.

[34] Prasad Raghavendra, David Steurer, and Madhur Tulsiani. Reductions between expansion problems. In *Proceedings of the 27th IEEE Annual Conference on Computational Complexity*. IEEE, 2012.

[35] Daniel A Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421(2-3), 2007.

[36] David Steurer and Nisheeth K Vishnoi. Connections between unique games and multicut. In *Electronic Colloquium on Computational Complexity*, volume 16, 2009.

[37] Luca Trevisan. Approximation algorithms for unique games. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2005.