Bilu–Linial Stable Instances of Max Cut and Minimum Multiway Cut

Konstantin Makarychev
Microsoft Research

Yury Makarychev*
TTIC

Aravindan Vijayaraghavan†
CMU

Abstract

We investigate the notion of stability proposed by Bilu and Linial. We obtain an *exact* polynomial-time algorithm for γ-stable Max Cut instances with $\gamma \geq c\sqrt{\log n}\log\log n$ for some absolute constant $c > 0$. Our algorithm is robust: it never returns an incorrect answer; if the instance is γ-stable, it finds the maximum cut, otherwise, it either finds the maximum cut or certifies that the instance is not γ-stable. We prove that there is no robust polynomial-time algorithm for γ-stable instances of Max Cut when $\gamma < \alpha_{SC}(n/2)$, where α_{SC} is the best approximation factor for Sparsest Cut with non-uniform demands.

Our algorithm is based on semidefinite programming. We show that the standard SDP relaxation for Max Cut (with ℓ_2^2 triangle inequalities) is integral if $\gamma \geq D_{\ell_2^2 \to \ell_1}(n)$, where $D_{\ell_2^2 \to \ell_1}(n)$ is the least distortion with which every n point metric space of negative type embeds into ℓ_1. On the negative side, we show that the SDP relaxation is not integral when $\gamma < D_{\ell_2^2 \to \ell_1}(n/2)$. Moreover, there is no tractable convex relaxation for γ-stable instances of Max Cut when $\gamma < \alpha_{SC}(n/2)$. That suggests that solving γ-stable instances with $\gamma = o(\sqrt{\log n})$ might be difficult or impossible.

Our results significantly improve previously known results. The best previously known algorithm for γ-stable instances of Max Cut required that $\gamma \geq c\sqrt{n}$ (for some $c > 0$) [Bilu, Daniely, Linial, and Saks]. No hardness results were known for the problem.

Additionally, we present an exact robust polynomial-time algorithm for 4-stable instances of Minimum Multiway Cut.

We also study a relaxed notion of *weak stability* and present algorithms for weakly stable instances of Max Cut and Minimum Multiway Cut.

1 Introduction

Empirical evidence suggests that many discrete optimization problems like clustering and partitioning are much easier in practice than in the worst cast. So even though these problems are usually provably hard in the worst case, we can still try to design algorithms that work well on instances we encounter in practice. To do so, we need a good mathematical model for such instances.

There are several approaches to modeling real-life instances. Perhaps, a more classical approach dating back to early 1980's is to assume that real-life instances come from a random or "semi-random" distribution. To learn more about this approach, we refer the interested reader to our previous work [26] on semi-random instances of graph partitioning and references therein. An alternative approach, which we study here, is to identify certain structural properties that "interesting" instances (or "practically interesting instances" [8]) must satisfy, and then assume that instances arising in practice satisfy them. One such property was proposed by Bilu and Linial [9].

Bilu and Linial [9] introduced a notion of *stability of instances* for discrete optimization problems. They argue that interesting instances have stable solutions: the optimal solution does not change upon small perturbations. For example, a clustering instance that is meaningful should have a solution that stands out. This solution should remain optimal even if the edge weights are slightly inaccurate or noisy. As Balcan, Blum and Gupta [4] argue, the real goal of solving a clustering problem is often to obtain the correct "target" clustering, and so the objective function serves only as a proxy. In this case, if the edge weights, which may be rough estimates of how similar or dissimilar the endpoints are, are imprecise, then the solution is meaningful only if the instance is stable. Here is a formal definition of γ-stability [9].

Definition 1.1 (γ-Stability [9]). *Consider an instance of a graph optimization problem on n vertices, defined by the matrix of non-negative edge weights w. We say that the instance is γ-stable if there is an optimal solution which remains optimal, even when any subset of the edge weights are increased by a factor of at most γ.*

We note that prior to work of Bilu and Linial [9], Balcan, Blum and Gupta [4] introduced and studied a somewhat similar but different notion of *approximation–stability* for clustering problems like k-means and k-median; later Awasthi, Blum and Sheffet [3] and Balcan and Liang [5] studied a related notion of *perturbation resilience* for these center–based clustering problems.

As in Bilu and Linial [9] and Bilu, Daniely, Linial, and Saks [8], our primary focus is studying stable instances of Max Cut. In Max Cut, we are given a weighted graph $G(V, E, w)$ on n vertices with an adjacency matrix w. Our goal is to find a cut $(S, V \setminus S)$ in the graph with the maximum weight of edges crossing it

$$\text{Max Cut } (G) = \max_{S \subseteq V(G)} \sum_{e \in E(S, V \setminus S)} w_e.$$

Max Cut is one of the classic NP-hard problems [17]. It is NP-hard to approximate within a factor of 17/16 [19, 28]. Goemans and Williamson [18] gave a semidefinite programming based algorithm that achieves a 0.878 approximation ratio for Max Cut. Khot, Kindler, Mossel and O'Donnell [22] showed that this is the best possible approximation ratio in the worst-case, assuming the Unique Games Conjecture [21].

In the work that introduced γ-stability, Bilu and Linial [9] designed an algorithm for γ-stable instances of Max Cut with $\gamma \geq cn$ (for some absolute constant c). The aim of the algorithm is to

find the exact optimal solution. This solution is unique (because of stability) and corresponds to the "true" partitioning we want to find. Finding just a good approximation for γ-stable instances of Max Cut is easy, since γ-stable instances of Max Cut are almost bipartite, and for almost bipartite graphs, the algorithm of Goemans and Williamson [18] returns a solution in which almost all edges are cut (see [9] for more details). Bilu, Daniely, Linial, and Saks [8] gave an algorithm for γ-stable Max Cut instances with $\gamma \geq c\sqrt{n}$ (for some absolute constant c). Both papers also gave better algorithms for stable instances that satisfy some extra conditions (see [9] and [8]).

Our Results. In this work, we give an algorithm that solves γ-stable instances of Max Cut with $\gamma \geq c\sqrt{\log n} \log \log n$ (for some absolute constant c). We also study the classic partitioning problem of finding the Minimum Multiway Cut (see Section 4 for the definition) and give an algorithm for γ-stable instances of it, with $\gamma \geq 4$. Our result for Max Cut is an exponential improvement over previous results. Our algorithms are *robust* (in the notion of Raghavan and Spinrad [27]):

1. If the instance is γ-stable, the algorithm finds the unique optimal solution.

2. If the instance is not γ-stable, the algorithm either finds an optimal solution, or a (polynomial-time verifiable) certificate that proves that the instance is not γ-stable.

In other words, our algorithm is always correct: when we claim to output the maximum cut (minimum multiway cut), we can guarantee its optimality; else we identify that the given graph is not sufficiently stable. This is a very desirable property for algorithms we want to use in practice, since we may only assume that real-life or important instances are stable (or satisfy other properties), but we cannot be completely certain that they indeed are. When we use robust algorithms, we cannot get a suboptimal solution even if our assumptions are not quite correct. Note that previous algorithms for stable instances of Max Cut [9, 8], and Clustering [4, 5, 3] are not robust. That is, if the instance is not γ-stable, the previous algorithms can output a suboptimal solution without notifying us that the solution is suboptimal.

Our algorithms use the fact that our SDP and LP relaxations for Max Cut and Minimum Multiway Cut, respectively, are integral when γ is sufficiently large. For Max Cut, we prove that the standard SDP relaxation with triangle inequalities is integral for $(c\sqrt{\log n} \log \log n)$-stable instances. We remark that we are unaware of other natural settings when the semidefinite program becomes integral and the corresponding linear program does not!

We also present algorithms that work for the same values of γ with a more relaxed notion of stability which we call *weak stability*. The optimal solution of every perturbed instance of a weakly stable instance, is close to the optimal solution of the original instance, but may not be exactly the same (see Section 6 for details). We believe that γ-weak stability may be a more realistic assumption than γ-stability in practice. Bilu and Linial [9] mentioned weakly stable instances in the introduction to their paper (without formally defining them), and proposed to study them in the future. Moreover, instances that satisfy the notion of approximation–stability of Balcan, Blum and Gupta [4] are also weakly stable. Our algorithms for γ-weakly stable instances are not robust.

Our result for weakly stable instances of Max Cut uses an approximation algorithm for Sparsest Cut with non-uniform demands as a black box. In particular, the result implies that if there is an $\alpha(n)$-approximation algorithm for Sparsest Cut then there is an exact algorithm for $(1 + \varepsilon)\alpha(n)$-stable instances of Max Cut. (For simplicity of exposition, ε is fixed in our proof; in general, ε can be sub-constant, the running time is proportional to $1/\varepsilon$.)

Negative Results. We supplement our algorithmic results, by showing that any robust algorithm for Max Cut that works for better values of γ would result in a similar improvement in the worst-case approximation for non-uniform Sparsest Cut. Moreover, we also show that the SDP is not integral if $\gamma < D_{\ell_2^2 \to \ell_1}(n/2)$ (where $D_{\ell_2^2 \to \ell_1}(n)$ is the least distortion with which every n-point ℓ_2^2 space can be embedded into ℓ_1). While our algorithmic results give algorithms for sufficiently stable instances, our reduction from non-uniform Sparsest Cut suggests that it may be hard to obtain robust algorithms for Max Cut that work for $O(1)$-stable instances. Finally, we describe a very strong negative result for the Max k-Cut problem when $k \geq 3$. We show that for every function $\gamma(n)$ there is no exact polynomial-time algorithm for $\gamma(n)$-stable instances of the problem unless $NP = RP$.

We note that our positive results for Max Cut also apply to the problem of clustering points into two clusters (or, equivalently, to Max Cut with positive and negative weights). Our negative result for Max k-Cut, on the other hand, shows that there is no exact algorithm for the problem of clustering points into k clusters when $k \geq 3$ unless $RP = NP$ (see Appendix B for details).

1.1   Overview of Techniques

The main technical component of our work is showing that the standard SDP relaxation with triangle inequalities is *integral* for γ-stable instances of Max Cut, when $\gamma \geq D_{\ell_2^2 \to \ell_1}(n)$, where $D_{\ell_2^2 \to \ell_1}(n) = O(\sqrt{\log n} \log \log n)$ is the least distortion with which every n-point ℓ_2^2 space can be embedded into ℓ_1. Hence, when the instance is γ-stable (for sufficiently large γ), we can read off the optimal solution from the SDP; otherwise, the non-integrality of the SDP certifies that the instance is not γ-stable. This gives a *robust* algorithm for instances with $\gamma \geq c\sqrt{\log n} \log \log n$.

Loosely speaking, we prove that if the SDP solution is not integral, then the integral optimal solution can be improved, possibly after changing the weights of some edges. Given the SDP solution $\{\bar{u}\}$ (a vector $\bar{u}$ for each vertex u) and the optimal integral solution $(S, V \setminus S)$, we define a new configuration of vectors $\{\hat{u}\}$: we keep $\hat{u} = \bar{u}$ if $u \in S$; and replace $\bar{u}$ with $\hat{u} = -\bar{u}$ if $u \in V \setminus S$. If $\{\bar{u}\}$ is the integral solution corresponding to $(S, V \setminus S)$, then all vectors $\hat{u}$ are equal. Now, if the SDP is not integral, and the vectors $\hat{u}$ are not all equal, then we embed these vectors into ℓ_1, and obtain a distribution of cuts $(A', V \setminus A')$. It turns, out that if the distortion of the embedding was 1, then we would be able to improve the quality of the optimal solution by picking a random cut $(A', S \setminus A')$ and moving vertices in $S \cap A'$ to $V \setminus S$ and vertices in $(V \setminus S) \cap A'$ to S (see Figure 1). Of course, the distortion may be much larger than 1. Then, we show how to compensate for the distortion using the stability. We can first change the weights of some edges by a factor at most $D_{\ell_2^2 \to \ell_1}(n)$, and only then move the vertices to improve the quality of the solution. Hence, we get a contradiction if $\gamma \geq D_{\ell_2^2 \to \ell_1}(n)$.

Our algorithm for weakly stable instances starts with an approximate solution and then iteratively improves the quality of the solution using the algorithm for non-uniform Sparsest Cut by Arora, Lee, and Naor [1] as a subroutine.

1.2   Outline

In Section 2, we introduce the formal definitions of stability and some preliminaries including the semidefinite program (SDP) we use in our algorithm. Then, in Section 3, we describe our robust algorithm for γ-stable instances of Max Cut. In Section 5, we present evidence suggesting that obtaining algorithms for better values of γ may not be easy. We first give a reduction from

non-uniform Sparsest Cut (in Section 5.2), which shows that any robust algorithm with better guarantees would lead to a similar improvement for non-uniform Sparsest Cut. Then we show in Section 5.3 that the SDP is not integral for smaller values of γ. In Section 4, we present our algorithm for 4-stable instances of Minimum Multiway Cut. Finally, in Section 6, we introduce a more general notion of weak stability and obtain similar guarantees in this setting.

2 Preliminaries

We start with formally defining the notion of Bilu–Linial stability for Max Cut instances. Following [9], we give two equivalent definitions (see Proposition 2.1 in [9]).

Definition 2.1 (Bilu and Linial [9]). *Let $G = (V, E, w)$ be a weighted graph with edge weights $w(e)$ and let $\gamma > 1$. A weighted graph $G' = (V, E, w')$ is a γ-perturbation of G if for every $(u, v) \in V$,*

$$w(u, v) \leq w'(u, v) \leq \gamma \cdot w(u, v).$$

We say that G is a γ-stable instance of Max Cut if there is a unique cut which forms a maximal cut for every γ-perturbation G' of G.

Definition 2.2 (Bilu and Linial [9]). *Let $\gamma \geq 1$. A weighted graph G graph with maximal cut $(S, \bar{S})$ is γ-stable instance of Max Cut if for every vertex set $T \neq S$ and $T \neq \bar{S}$:*

$$w(E(S, \bar{S}) \setminus E(T, \bar{T})) > \gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S})).$$

We use the Goemans–Williamson SDP relaxation for Max Cut with additional ℓ_2^2-triangle inequalities [18]. In the SDP relaxation, we have an SDP vector variable $\bar{u}$ for every vertex $u \in V$. The intended SDP solution corresponding to an integral solution $(S, \bar{S})$ assigns a fixed unit vector $\bar{e}$ to variable $\bar{u}$ if $u \in S$, and assigns $-\bar{e}$ to $\bar{u}$, otherwise.

$$\text{maximize } \frac{1}{4} \sum_{(u,v) \in E} w_{uv} \|\bar{u} - \bar{v}\|^2$$

subject to

$$\|\bar{u}\|^2 = 1 \quad \text{for every } u \in V,$$
$$\|\bar{u} - \bar{v}\|^2 + \|\bar{v} - \bar{w}\|^2 \geq \|\bar{u} - \bar{w}\|^2 \quad \text{for every } u, v, w \in V,$$
$$\|\bar{u} - \bar{v}\|^2 + \|\bar{v} + \bar{w}\|^2 \geq \|\bar{u} + \bar{w}\|^2 \quad \text{for every } u, v, w \in V,$$
$$\|\bar{u} + \bar{v}\|^2 + \|\bar{v} + \bar{w}\|^2 \geq \|\bar{u} - \bar{w}\|^2 \quad \text{for every } u, v, w \in V.$$

The last three constraints are ℓ_2^2-triangle inequalities for the set of vectors $\{\pm \bar{u} : u \in V\}$. Note that the intended solution satisfies all constraints, and its value equals the cost of the cut $(S, \bar{S})$.

We show that γ-stable instances of Max Cut are integral. The formal definition of an integral SDP is as follows.

Definition 2.3. *Let G be an instance of Max Cut. We say that an SDP solution $\{\bar{u}\}$ is integral if there exists a vector $\bar{e}$ such that $\bar{u} = \bar{e}$ or $\bar{u} = -\bar{e}$ for every $u \in V$. We say that the SDP relaxation for G is integral if every optimal SDP solution for G is integral.*

4

Our algorithm for γ-stable instances is robust in the sense of Raghavan and Spinrad [27]: it always returns a correct output regardless of whether the input is γ-stable or not.

Definition 2.4. *An algorithm for γ-stable instances of Max Cut is robust if the following conditions hold.*

- *If the input instance is γ-stable, the algorithm must output a maximum cut.*

- *If the input instance is not γ-stable, the algorithm must either output a maximum cut or a special symbol $\perp$ (which certifies that the instance is not γ-stable).*

Metrics of Negative Type and Sparsest Cut with Non-Uniform Demands. In the proof, we use some standard definitions from metric geometry.

Definition 2.5. *The Lipschitz constant $\|\varphi\|_{Lip}$ of a map φ between two metric spaces (X, d_X) and (Y, d_Y) equals*

$$\|\varphi\|_{Lip} = \sup_{\substack{x,y \in X \\ x \neq y}} \frac{d_Y(\varphi(x), \varphi(y))}{d_X(x,y)}.$$

The distortion of an embedding $\varphi : X \hookrightarrow Y$ equals $\|\varphi\|_{Lip} \cdot \|\varphi^{-1}\|_{Lip}$ (the distortion is infinite if φ is not injective).

Definition 2.6. *A set $X \subset \ell_2$ is an ℓ_2^2 space, if the distance function $d(u,v) = \|u - v\|^2$ satisfies triangle inequalities:*

$$d(u,v) + d(v,w) \geq d(u,w) \quad \text{for every } u, v, w \in X.$$

A metric space is of negative type if it is isometric to an ℓ_2^2 space. We denote the least distortion with which every n point metric space of negative type embeds into ℓ_1 by $D_{\ell_2^2 \to \ell_1}(n)$ or just $D_{\ell_2^2 \to \ell_1}$.

Arora, Lee and Naor [1] proved that $D_{\ell_2^2 \to \ell_1}(n) = O(\sqrt{\log n} \log \log n)$; on the other hand, Cheeger, Kleiner and Naor [13] showed that $D_{\ell_2^2 \to \ell_1}(n) = \Omega(\log^c n)$ for some constant $c > 0$, and it is widely believed that $D_{\ell_2^2 \to \ell_1}(n) = \Omega(\sqrt{\log n})$ (that is, that the upper bound of Arora, Lee and Naor is almost optimal).

Note that since every ℓ_1 metric can be represented as a linear combination of cut metrics with non-negative coefficients, we have the following fact.

Fact 2.7. *Let $X \subset \ell_2$ be an n-point ℓ_2^2 space ($n > 1$). Then there exists a distribution of random cuts $(A, \bar{A} = X \setminus A)$ and a scale parameter $\sigma > 0$ such that for every pair $(x, y) \in X \times X$*

$$\sigma \cdot \Pr\left((x,y) \text{ is separated by } (A, \bar{A})\right) \leq \|x - y\|^2 \leq \sigma \cdot D_{\ell_2^2 \to \ell_1}(n) \cdot \Pr\left((x,y) \text{ is separated by } (A, \bar{A})\right).$$

Now recall the definition the Sparsest Cut problem (with non-uniform demands).

Definition 2.8 (Sparsest Cut problem with Non-uniform Demands)**.** *We are given a connected graph $G = (V, E_c)$, with positive edge capacities cap_{uv}, a set of demands pairs $E_d \subset V \times V$, and positive demands dem_{uv}. The sparsity of a cut $(A, \bar{A})$ is*

$$\phi(A) = \frac{\mathrm{cap}(E_c(A, \bar{A}))}{\mathrm{dem}(E_d(A, \bar{A}))},$$

5

where $E_c(A, \bar{A})$ is the set of capacity edges between A and $\bar{A}$, $\mathrm{cap}_{uv}(E_c(A, \bar{A}))$ is their total capacity, $E_d(A, \bar{A})$ is the set of demand pairs separated by $(A, \bar{A})$, and $\mathrm{dem}(E_d(A, \bar{A}))$ is their total demand. Our goal is to find a cut $(A, \bar{A})$ with minimum sparsity $\phi(A)$. We denote the best possible approximation factor for the problem by $\alpha_{SC}(n)$. (Strictly speaking, $\alpha_{SC}(n)$ is not uniquely defined. Formally, $\alpha_{SC}(n) \leq f(n)$ means that there is an algorithm with approximation guarantee $f(n)$; $\alpha_{SC}(n) > f(n)$ means there is no algorithm with approximation guarantee $\alpha_{SC}(n)$.)

Arora, Lee, and Naor showed that $\alpha_{SC}(n) \leq D_{\ell_2^2 \to \ell_1}(n) = O(\sqrt{\log n} \log \log n)$. Chawla, Krauthgamer, Kumar, Rabani, and Sivakumar [12] and independently Khot and Vishnoi [23] proved that $\alpha_{SC}(n) \to \infty$ as $n \to \infty$ assuming the Unique Games Conjecture (UGC); moreover, they showed that $\alpha_{SC}(n) = \Omega((\log \log n)^{1/2})$ assuming some strong version of UGC. Chuzhoy and Khanna [14] proved that there is no polynomial-time approximation scheme for Sparsest Cut; that is, that $\alpha_{SC} > 1 + \varepsilon$ for some absolute constant $\varepsilon > 0$.

3   Algorithm for Max Cut

In this section, we prove that the SDP relaxation for every γ-stable Max Cut instance with $\gamma \geq D_{\ell_2^2 \to \ell_1}$ is integral and as a corollary obtain a polynomial-time algorithm for γ-stable instances.

Theorem 3.1. *The SDP relaxation for every γ-stable Max Cut instance with $\gamma \geq D_{\ell_2^2 \to \ell_1}$ is integral.*

Proof. Let G be a γ-stable instance with $\gamma \geq D_{\ell_2^2 \to \ell_1}$ and $\{\bar{u}\}$ be an optimal SDP solution for G. Assume to the contrary that $\{\bar{u}\}$ is not integral.

Let $(S, \bar{S})$ be the maximum cut in G. Since $\{\bar{u}\}$ is an optimal SDP solution, we have that its SDP value is at least the cost of the maximum cut:

$$\frac{1}{4} \sum_{(u,v) \in E} w_{uv} \|\bar{u} - \bar{v}\|^2 \geq w(E(S, \bar{S})). \tag{1}$$

Define

$$\hat{u} = \begin{cases} \bar{u}, & \text{if } u \in S, \\ -\bar{u}, & \text{if } u \notin S. \end{cases}$$

If $(u, v) \in E(S, \bar{S})$ then

$$\|\bar{u} - \bar{v}\|^2 = 2 - 2\langle \bar{u}, \bar{v} \rangle = 2 + 2\langle \hat{u}, \hat{v} \rangle = 4 - \|\hat{u} - \hat{v}\|^2;$$

if $(u, v) \in E \setminus E(S, \bar{S})$ then $\|\bar{u} - \bar{v}\|^2 = \|\hat{u} - \hat{v}\|^2$. Therefore,

$$
\begin{aligned}
\frac{1}{4} \sum_{(u,v) \in E} w_{uv} \|\bar{u} - \bar{v}\|^2 &= \frac{1}{4} \sum_{(u,v) \in E(S,\bar{S})} w_{uv} \|\bar{u} - \bar{v}\|^2 + \frac{1}{4} \sum_{(u,v) \in E \setminus E(S,\bar{S})} w_{uv} \|\bar{u} - \bar{v}\|^2 \\
&= \frac{1}{4} \sum_{(u,v) \in E(S,\bar{S})} w_{uv} (4 - \|\hat{u} - \hat{v}\|^2) + \frac{1}{4} \sum_{(u,v) \in E \setminus E(S,\bar{S})} w_{uv} \|\hat{u} - \hat{v}\|^2 \\
&= w(E(S, \bar{S})) + \frac{1}{4} \sum_{(u,v) \in E \setminus E(S,\bar{S})} w_{uv} \|\hat{u} - \hat{v}\|^2 - \frac{1}{4} \sum_{(u,v) \in E(S,\bar{S})} w_{uv} \|\hat{u} - \hat{v}\|^2.
\end{aligned}
$$

6

From (1), we get

$$\sum_{(u,v)\in E(S,\bar{S})} w_{uv}\|\hat{u}-\hat{v}\|^2 \leq \sum_{(u,v)\in E\setminus E(S,\bar{S})} w_{uv}\|\hat{u}-\hat{v}\|^2.$$

The set $X = \{\hat{u} : u \in V\}$ is an ℓ_2^2 space since the SDP solution ensures that vectors in $\{\pm\bar{u} : u \in V\}$ satisfy ℓ_2^2-triangle inequalities and $X \subset \{\pm\bar{u} : u \in V\}$. This space embeds into ℓ_1 with distortion $D_{\ell_2^2\to\ell_1}(n)$, and, hence (see Fact 2.7) there is a distribution of random cuts $(A,\bar{A})$ of X and a parameter $\sigma > 0$ such that

$$\sigma \cdot \Pr\left(\text{pair } (\hat{u},\hat{v}) \text{ is separated by } (A,\bar{A})\right) \leq \|\hat{u}-\hat{v}\|^2$$
$$\leq \sigma \cdot D_{\ell_2^2\to\ell_1}(n) \cdot \Pr\left(\text{pair } (\hat{u},\hat{v}) \text{ is separated by } (A,\bar{A})\right).$$

Let $A' = \{u : \hat{u} \in A\}$ and $\bar{A}' = V \setminus A' = \{u : \hat{u} \notin A\}$. We get,

$$\sum_{(u,v)\in E(S,\bar{S})} w_{uv}\sigma \Pr\left((u,v) \in E(A',\bar{A}')\right) \leq \sum_{(u,v)\in E(S,\bar{S})} w_{uv}\|\hat{u}-\hat{v}\|^2$$
$$\leq \sum_{(u,v)\in E\setminus E(S,\bar{S})} w_{uv}\|\hat{u}-\hat{v}\|^2$$
$$\leq \sum_{(u,v)\in E\setminus E(S,\bar{S})} w_{uv}\sigma D_{\ell_2^2\to\ell_1}(n) \Pr\left((u,v) \in (A',\bar{A}')\right).$$

Therefore,

$$\mathbb{E}\left[w(E(S,\bar{S}) \cap E(A',\bar{A}'))\right] \leq D_{\ell_2^2\to\ell_1}(n) \cdot \mathbb{E}\left[w((E \setminus E(S,\bar{S})) \cap E(A',\bar{A}'))\right].$$

In particular, for some cut A'', we have

$$w(E(S,\bar{S}) \cap E(A'',\bar{A}'')) \leq D_{\ell_2^2\to\ell_1}(n) \cdot w((E \setminus E(S,\bar{S})) \cap E(A'',\bar{A}''))$$
$$\leq \gamma \cdot w((E \setminus E(S,\bar{S})) \cap E(A'',\bar{A}'')).$$

Let $T = (S \cap A'') \cup (\bar{S} \cap \bar{A}'')$ (see Figure 1). Note that $A'' \neq V$ and $A'' \neq \varnothing$, hence $T \neq S$ and $T \neq \bar{S}$. Write

$$E(S,\bar{S}) \cap E(A'',\bar{A}'') = E(S \cap A'', \bar{S} \cap \bar{A}'') \cup E(S \cap \bar{A}'', \bar{S} \cap A'')$$
$$= E(S \cap T, \bar{S} \cap T) \cup E(S \cap \bar{T}, \bar{S} \cap \bar{T}) = E(S,\bar{S}) \setminus E(T,\bar{T}),$$
$$(E \setminus E(S,\bar{S})) \cap E(A'',\bar{A}'') = E(S \cap A'', S \cap \bar{A}'') \cup E(\bar{S} \cap A'', \bar{S} \cap \bar{A}'')$$
$$= E(T \cap A'', \bar{T} \cap A'') \cup E(T \cap \bar{A}'', \bar{T} \cap \bar{A}'') = E(T,\bar{T}) \cap E(S,\bar{S}).$$

We get,

$$w(E(S,\bar{S}) \setminus E(T,\bar{T})) \leq \gamma \cdot w(E(T,\bar{T}) \setminus E(S,\bar{S})),$$

which contradicts to the fact that G is a γ-stable instance (see Definition 2.2). $\qquad\square$

From Theorem 3.1, we get the main algorithmic result of our paper.

Corollary 3.2. *There is a robust polynomial-time algorithm for γ-stable instance of Max Cut with $\gamma \geq D_{\ell_2^2\to\ell_1}(n)$.*
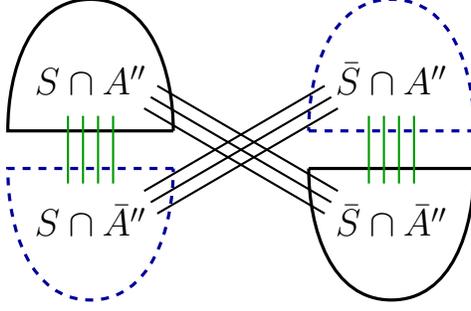
Figure 1: In the figure, the set $T = (S \cap A'') \cup (\bar{S} \cap \bar{A}'')$ is represented by semicircles in the upper-left and lower-right corners; the set $\bar{T} = (S \cap \bar{A}'') \cup (\bar{S} \cap A'')$ is represented by semicircles in the lower-left and upper-right corners. Edges in $E(S, \bar{S}) \setminus E(T, \bar{T})$ are drawn diagonally; edges in $E(T, \bar{T}) \setminus E(S, \bar{S})$ are drawn vertically.

Proof. The algorithm solves the SDP relaxation for the problem. If the solution is integral, the algorithm returns the cut corresponding to it. Otherwise, it returns $\bot$ (indicating that the instance is not γ-stable). Note that if the algorithm returns a cut, it must be a maximum cut (otherwise, the SDP solution would not be optimal). By Theorem 3.1, the algorithm always returns a solution if the instance is γ-stable. $\qquad\square$

4   Algorithm for Minimum Multiway Cut

In this section, we study stable instances of Minimum Multiway Cut. We prove that the linear programming relaxation of Călinescu, Karloff, and Rabani is integral for 4-stable instances of the problem. Thus there is a robust polynomial-time algorithm for 4-stable instances of Minimum Multiway Cut.

The Minimum Multiway Cut problem was introduced by Dahlhaus, Johnson, Papadimitriou, Seymour, and Yannakakis [16]. We refer the reader to [10] for the summary of known results for the problem.

Definition 4.1. *In the Minimum Multiway Cut problem, we are given a graph $G = (V, E, w)$ with positive edge weights w_e and a set of terminals $T = \{s_1, \ldots, s_k\} \subset V$. Our goal is to partition the graph into k pieces $S_1, \ldots, S_k$ such that $s_i \in S_i$ so as to minimize the total weight of cut edges.*

We give a definition of γ-stable instances of Minimum Multiway Cut (cf. Definition 2.1).

Definition 4.2. *Let $\gamma > 1$. An instance $\{G = (V, E, w), T\}$ of Minimum Multiway Cut is γ-stable if there is a multiway cut $\mathcal{S}$ which is the unique optimal solution for every γ-pertrubation of G.*

We also restate this definition as follows (cf. Definition 2.2).

Definition 4.3. *Consider an instance $\{G = (V, E, w), T\}$ of Minimum Multiway Cut. Let $\gamma > 1$. Denote the optimal multiway cut by $\mathcal{S}^*$, and let the set it cuts be E^*. We say that G is a γ-stable instance of Multiway Cut if for every multiway cut $\mathcal{S}' \neq \mathcal{S}^*$, we have*

$$w(E' \setminus E^*) > \gamma \cdot w(E^* \setminus E'),$$

where E' is the set of edges cut by $\mathcal{S}'$.

Consider the LP relaxation of Călinescu, Karloff, and Rabani [15]. In this relaxation, there is a variable $\bar{u} = (u_1, \ldots, u_k) \in \mathbb{R}^k$ for every vertex $u \in V$. Let $e_1, \ldots, e_k$ be the standard basis in $\mathbb{R}^k$ and $\Delta = \{x : \|x\|_1 = 1, x_1 \geq 0, \ldots, x_k \geq 0\}$ be the simplex with vertices $e_1, \ldots, e_k$.

$$\text{minimize } \frac{1}{2} \sum_{(u,v) \in E} w(u,v) \|\bar{u} - \bar{v}\|_1 \tag{2}$$

subject to

$$\bar{s}_i = e_i \qquad\qquad\qquad\qquad \text{for every } i,$$
$$\bar{u} \in \Delta \qquad\qquad\qquad\qquad \text{for every } u \in V.$$

Every feasible LP solution defines a metric on V: $d(u,v) = \|\bar{u} - \bar{v}\|_1$. We will need the following lemma.

Lemma 4.4. *Consider a feasible LP solution $\{\bar{u} : u \in V\}$. There is a distribution of multiway cuts (partitions) $S_1, \ldots, S_k$ such that*

- $s_i \in S_i$ *for every $i \in \{1, \ldots, k\}$ (always)*,

- $\Pr(u$ *and v are separated by the cut$) \leq \frac{2d(u,v)}{1+d(u,v)}$ for every u and v (u and v are separated if $u \in S_i$ and $v \in S_j$ with $i \neq j$). In particular, for every edge (u,v),*

$$\Pr((u,v) \text{ is cut}) \leq 2d(u,v) \quad and \quad \Pr((u,v) \text{ is not cut}) \geq \frac{1 - d(u,v)}{2}.$$

If the LP solution is not integral, the distribution is supported on at least two multiway cuts.

Proof. We round the LP solution to an integral solution using the dependent sampling technique of Holenstein (which he used in his proof of the parallel repetition theorem) [20]. We iteratively construct sets $S_1, \ldots, S_k$. We start with empty sets $S_1, \ldots, S_k$ and then in each iteration add vertices to one of the sets $S_1, \ldots, S_k$. We stop once each vertex u belongs to some set u. In each iteration, we choose independently and uniformly at random $r \in (0,1)$ and $i \in \{1, \ldots, k\}$. We add each vertex u to S_i if $r \leq \bar{u}_i$ and it was not added to any set S_j in previous iterations.

First, note that we add every vertex u to some S_j with probability $\sum_{i=1}^{k} \bar{u}_i/k = 1/k$ in each iteration (unless u already lies in some S_j). So eventually we will add every vertex to some set S_j. Also note that we cannot add s_i to S_j if $j \neq i$. Therefore, $s_i \in S_i$.

Now consider two vertices u and v. Consider an iteration of our partitioning algorithm. Suppose that neither u nor v is assigned to some S_j. The probability that at least one of them is assigned to S_i in this iteration is

$$\frac{1}{k} \sum_{i=1}^{k} \max(\bar{u}_i, \bar{v}_i) = \frac{1}{k} \sum_{i=1}^{k} \left(\frac{\bar{u}_i + \bar{v}_i}{2} + \frac{|\bar{u}_i - \bar{v}_i|}{2} \right) = \frac{1}{k} \left(1 + \frac{\|\bar{u} - \bar{v}\|_1}{2} \right) = \frac{1 + d(u,v)}{k}.$$

The probability that exactly one of them is assigned to S_i is

$$\frac{1}{k} \sum_{i=1}^{k} |\bar{u}_i - \bar{v}_i| = \frac{\|\bar{u} - \bar{v}\|_1}{k} = \frac{2d(u,v)}{k}.$$

Therefore, the probability that u and v are separated in one of iterations is at most $2d(u,v)/(1 + d(u,v))$. $\qquad\square$

9

Remark 4.5. *Note that in general this rounding scheme gives only a 2 approximation. Other known rounding schemes for Multiway Cut achieve a better approximation; e.g. the rounding scheme of Călinescu, Karloff, and Rabani [15] gives a 3/2 approximation. However, this rounding scheme has a property that other rounding schemes do not have: it does not cut an edge (u, v) with probability at least $(1 - d(u, v))/2 = \Omega(1 - d(u, v))$. This property is crucial for our proof.*

Now we prove the main result of this section.

Theorem 4.6. *The LP relaxation is integral if the instance is 4-stable.*

Proof. Consider a 4-stable instance of Multiway Cut. Let $\mathcal{S}^*$ be the minimum multiway cut and E^* be the set of edges cut by it. Let $\{\bar{u} : u \in V\}$ be the optimal solution of the LP relaxation. Assume to the contrary that it is not integral. Consider a random multiway $\mathcal{S}' = (S_1, \ldots, S_k)$ as in Lemma 4.4. Let E' be the set of edges cut by it (E' is a random variable). Note that since the input instance is 4-stable, we have $4w(E^* \setminus E') < w(E' \setminus E^*)$ unless $\mathcal{S}' = \mathcal{S}^*$. Since the LP solution is not integral, $\mathcal{S}' \neq \mathcal{S}^*$ with positive probability, and thus $4\mathbb{E}\left[w(E^* \setminus E')\right] < \mathbb{E}\left[w(E' \setminus E^*)\right]$.

Let

$$
\begin{aligned}
\mathsf{LP}_+ &= \sum_{(u,v) \in E^*} w(u, v)(1 - d(u, v)), \\
\mathsf{LP}_- &= \sum_{(u,v) \in E \setminus E^*} w(u, v)\, d(u, v).
\end{aligned}
$$

We have,

$$
\mathbb{E}\left[w(E^* \setminus E')\right] = \sum_{(u,v) \in E^*} w(u, v)\, \Pr((u, v) \text{ is not cut}) \geq \sum_{(u,v) \in E^*} w(u, v)(1 - d(u, v))/2 = \mathsf{LP}_+/2,
$$

$$
\mathbb{E}\left[w(E' \setminus E^*)\right] = \sum_{(u,v) \in E \setminus E^*} w(u, v)\, \Pr((u, v) \text{ is cut}) \leq 2 \sum_{(u,v) \in E^*} w(u, v) d(u, v) = 2\, \mathsf{LP}_-.
$$

We conclude that $\mathsf{LP}_+ < \mathsf{LP}_-$. On the other hand,

$$
\mathsf{LP}_+ - \mathsf{LP}_- = w(E^*) - \sum_{(u,v) \in E} w(u, v)\, d(u, v) = w(E^*) - \frac{1}{2} \sum_{(u,v) \in E} w(u, v)\, \|\bar{u} - \bar{v}\|_1 \geq 0
$$

since the value of the relaxation is at most the value of the integral solution. We get a contradiction.

Finally, note that if some $\bar{u}_j \in (0, 1)$ then with positive probability $u \notin S_i$, and with positive probability $u \notin S_i$ (specifically, the probabilities of these events are $\bar{u}_i$ and $1 - \bar{u}_i$, respectively). Therefore, if the LP solution is not integral, the distribution of multiway cuts is supported on at least two multiway cuts. $\qquad \square$

As an immediate corollary we get that there is a robust polynomial-time algorithm for 4-stable instances of Multiway Cut.

Corollary 4.7. *There is a robust polynomial-time algorithm for 4-stable instances of Multiway Cut.*

Proof. We solve the LP relaxation for Multiway Cut. If the LP solution is integral, we return the corresponding combinatorial solution. Otherwise, we return that the instance is not 4-stable. $\qquad \square$

5 Negative Results

In this section, we present our hardness results and prove that the SDP relaxation is not integral if $\gamma < D_{\ell_2^2 \to \ell_1}(n/2)$.

5.1 Reduction from Sparsest Cut to Max Cut

We first present a reduction from Sparsest Cut to Max Cut, which we use later to prove both our negative results.

Consider a Sparsest Cut instance. Denote the set of vertices by V_0, the set of capacity edges by E_c, the set of demand pairs by E_d, edge capacities by cap_{uv}, and demands by dem_{uv}. Define graph $G(V, E, w)$ as follows. Introduce two vertices u_1 and u_2 for every $u \in V_0$, and let $V = \{u_1, u_2 : u \in V_0\}$. Let

$$E = \{(u_1, v_2) : (u, v) \in E_c\} \cup \{(u_1, v_1), (u_2, v_2) : (u, v) \in E_d\} \cup \{(u_1, u_2) : u \in V\}.$$

Define edge weights by $w(u_1, v_2) = \mathrm{cap}_{uv}$, $w(u_1, v_1) = w(u_2, v_2) = \mathrm{dem}_{uv}$, and $w(u_1, u_2) = W_\infty$, where W_∞ is an arbitrary number larger than

$$\gamma \cdot w(\{(u_1, v_2) : (u, v) \in E_c\} \cup \{(u_1, v_1), (u_2, v_2) : (u, v) \in E_d\}).$$

Let $S = \{u_1 : u \in V_0\}$ and $\bar{S} = V \setminus S = \{u_2 : u \in V_0\}$.

Lemma 5.1. *If $\phi(A) > \gamma$ for every cut $(A, \bar{A})$ (see Definition 2.8), then the instance G is γ-stable with the maximum cut $(S, \bar{S})$.*

Proof. We need to show that for every cut $(T, \bar{T})$ different from $(S, \bar{S})$:

$$w(E(S, \bar{S}) \setminus E(T, \bar{T})) > \gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S})).$$

Here, we use Definition 2.2 of γ-stability. Note that if for some u, the edge (u_1, u_2) is not cut by $E(T, \bar{T})$ then $w(E(S, \bar{S}) \setminus E(T, \bar{T})) \geq w(u_1, u_2) = W_\infty$ and $\gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S})) < W_\infty$, and the desired inequality holds. So we assume below that every edge (u_1, u_2) is cut by $E(T, \bar{T})$. Then, for every u either $u_1 \in T$ and $u_2 \in \bar{T}$, or $u_1 \in \bar{T}$ and $u_2 \in T$. Let

$$A = \{u \in V_0 : u_1 \in T\} = \{u \in V_0 : u_2 \in \bar{T}\}; \tag{3}$$
$$\bar{A} = \{u \in V_0 : u_1 \in \bar{T}\} = \{u \in V_0 : u_2 \in T\}. \tag{4}$$

Observe, that $S \cap T = \{u_1 : u \in A\}$; $S \cap \bar{T} = \{u_1 : u \in \bar{A}\}$, similarly $\bar{S} \cap T = \{u_2 : u \in \bar{A}\}$; $\bar{S} \cap \bar{T} = \{u_2 : u \in A\}$. Since $\phi(A) > \gamma$, we have

$$
\begin{aligned}
w(E(S, \bar{S}) \setminus E(T, \bar{T})) &= w(E(S \cap T, \bar{S} \cap T)) + w(E(S \cap \bar{T}, \bar{S} \cap \bar{T})) \\
&= 2\,\mathrm{cap}(E_c(A, \bar{A})) > \gamma \cdot 2\,\mathrm{dem}(E_d(A, \bar{A})) \\
&= w(E(S \cap T, S \cap \bar{T})) + w(E(\bar{S} \cap T, \bar{S} \cap \bar{T})) = w(E(T, \bar{T}) \setminus E(S, \bar{S})),
\end{aligned}
$$

as required. We proved that the instance is γ-stable. $\qquad\square$

5.2 Hardness Result for Max Cut

We now prove that there is no robust polynomial-time algorithm for γ-stable instances of Max Cut when $\gamma < \alpha_{SC}(n/2)$.

Theorem 5.2. *Suppose that there is a robust polynomial-time algorithm $\mathcal{A}$ for γ-stable instances of Max Cut with $\gamma \geq \gamma(n)$. Then there is a polynomial-time algorithm for the decision version of Sparsest Cut with promise that either*

$$\phi^* = \min_{(A,\bar{A})} \phi(A) < 1 \ or \ \phi^* > \gamma(2n).$$

The algorithm given a Sparsest Cut instance decides whether $\phi^* < 1$ or $\phi^* > \gamma(2n)$.

Proof. We apply reduction from Section 5.1 and obtain a graph G on $2n$ vertices. Then we run $\mathcal{A}$ on G. If $\mathcal{A}$ returns the cut $(S, \bar{S})$ (where $S = \{u_1 : u \in V_0\}$), we decide that $\phi^* > \gamma(2n)$. Otherwise, we decide that $\phi^* < 1$.

We prove that we always decide correctly. Assume first that $\phi^* > \gamma(2n)$ then G is $\gamma(2n)$-stable and $(S, \bar{S})$ is the maximum cut by Lemma 5.1. Therefore, $\mathcal{A}$ returns $(S, \bar{S})$ and we correctly decide that $\phi^* > \gamma(2n)$. Now assume that $\phi^* < 1$. Denote the sparsest cut in G by A. Let $T = \{u_1 : u \in A\} \cup \{u_2 : u \notin A\}$. We have,

$$w(E(S,\bar{S})) - w(E(T,\bar{T})) = w(E(S \cap T, \bar{S} \cap T)) + w(E(S \cap \bar{T}, \bar{S} \cap \bar{T}))$$
$$- (w(E(S \cap T, S \cap \bar{T})) + w(E(\bar{S} \cap T, \bar{S} \cap \bar{T})))$$
$$= 2 \operatorname{cap}(E_c(A, \bar{A})) - 2 \operatorname{dem}(E_d(A, \bar{A})) < 0.$$

Hence $(S, \bar{S})$ is not a maximum cut. Since $\mathcal{A}$ is a robust algorithm it must either return a cut different from $(S, \bar{S})$ or $\perp$. Therefore, we decide that $\phi^* > \gamma(2n)$. $\qquad\square$

We note that Theorem 5.2 implies that there is no polynomially-time tractable relaxation for Max Cut that is integral on γ-stable instances if $\gamma < \alpha_{SC}(n/2)$. If there was such a relaxation, by solving it, we would get a robust algorithm as we do in Corollary 3.2.

Corollary 5.3. *There is no polynomial-time tractable relaxation for Max Cut that is integral on γ-stable instances if $\gamma < \alpha_{SC}(n/2)$.*

5.3 SDP Integrality Gap

In this section, we prove that the SDP relaxation for γ-stable instances is not integral in general when $\gamma < D_{\ell_2^2 \to \ell_1}(n)$. To this end, we show how to transform an integrality gap Sparsest Cut instance to a γ-stable Max Cut instance with a non-integral SDP solution.

We say that an instance of Sparsest Cut has integrality gap $D > 1$ if for some assignment of vectors $u \mapsto \bar{u}$ such that the set $\{\bar{u}\}$ is an ℓ_2^2 space, we have

$$\phi(A) = \frac{\operatorname{cap}(E_c(A, \bar{A}))}{\operatorname{dem}(E_d(A, \bar{A}))} \geq D \cdot \frac{\sum_{(u,v) \in E_c} \operatorname{cap}_{uv} \cdot \|\bar{u} - \bar{v}\|^2}{\sum_{(u,v) \in E_d} \operatorname{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2} \quad \text{for every cut}(A, \bar{A})$$

Fact 5.4 (Linial, London and Rabinovich [25], Aumann and Rabani [2]). *For every $n > 1$, there is an instance of Sparsest Cut on n vertices with integrality gap $D_{\ell_2^2 \to \ell_1}(n)$.*

We will need the following technical lemma.

Lemma 5.5. *For every $n > 1$ and $\varepsilon > 0$, there exists a Sparsest Cut instance on n vertices and unit vectors $\{\bar{u}\}$ such that vectors $\{\pm\bar{u}\}$ form an ℓ_2^2 space and*

$$\mathrm{cap}(E_c(A, \bar{A})) > (D_{\ell_2^2 \to \ell_1} - \varepsilon) \cdot \mathrm{dem}(E_d(A, \bar{A})) \qquad \text{for every cut } (A, \bar{A}) \tag{5}$$

$$\sum_{(u,v) \in E_c} \mathrm{cap}_{uv} \cdot \|\bar{u} - \bar{v}\|^2 < \sum_{(u,v) \in E_d} \mathrm{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2. \tag{6}$$

To prove the lemma, we first rescale demands so that conditions (5) and (6) hold. Then we transform vectors u so that all of them lie on the unit sphere. Specifically, if all vectors u_i lie on some sphere, we scale all vectors u_i to unit vectors and move the origin to the center of the sphere; these transformations preserve ratios of distances between vectors. In a degenerate case, when all vectors u_i do not lie on a sphere, we first slightly perturb all vectors and then apply the above argument. The formal proof is a bit technical, so we present it in Appendix A.

Theorem 5.6. *For every n and $\gamma \in [1, D_{\ell_2^2 \to \ell_1}(n/2))$, there is a γ-stable instance G of Max Cut on $2n$ vertices, such that that the SDP relaxation for G is not integral.*

Proof. Let $\varepsilon = (D_{\ell_2^2 \to \ell_1}(n) - \gamma)/2$. From Lemma 5.5, we get a Sparsest Cut instance on a set V_0 and vectors $\bar{u}$ such that $\{\pm\bar{u}\}$ is an ℓ_2^2 space and inequalities (5) and (6) hold. We apply reduction from Section 5.1 to this instance and obtain a graph $G(V, E, w)$. From Lemma 5.1 and inequality (5), we get that G is a γ-stable Max Cut instance.

We define an SDP solution for the SDP relaxation for G by $\bar{u}_1 = \bar{u}$ and $\bar{u}_2 = -\bar{u}$. Since all vectors $\bar{u}$ are unit vectors and $\{\pm\bar{u}\}$ is an ℓ_2^2 space, this is a feasible SDP solution. Its value equals

$$\begin{aligned}
SDP &\equiv \frac{1}{4} \sum_{(x,y) \in E} w(x, y) \|\bar{x} - \bar{y}\|^2 \\
&= nW_\infty + \frac{1}{4} \sum_{(u,v) \in E_d} \mathrm{dem}_{uv} \cdot (\|\bar{u}_1 - \bar{v}_1\|^2 + \|\bar{u}_2 - \bar{v}_2\|^2) \\
&\qquad + \frac{1}{4} \sum_{(u,v) \in E_c} \mathrm{cap}_{uv} \cdot (\|\bar{u}_1 - \bar{v}_2\|^2 + \|\bar{u}_2 - \bar{v}_1\|^2) \\
&= nW_\infty + \frac{1}{2} \sum_{(u,v) \in E_d} \mathrm{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2 + \frac{1}{2} \sum_{(u,v) \in E_c} \mathrm{cap}_{uv} \|\bar{u} + \bar{v}\|^2.
\end{aligned}$$

Using that $\|\bar{u} + \bar{v}\|^2 = 4 - \|\bar{u} - \bar{v}\|^2$, we get

$$SDP = (nW_\infty + 2\mathrm{cap}(E_c)) + \frac{1}{2} \sum_{(u,v) \in E_d} \mathrm{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2 - \frac{1}{2} \sum_{(u,v) \in E_c} \mathrm{dem}_{uv} \|\bar{u} - \bar{v}\|^2.$$

The first term $nW_\infty + 2\mathrm{cap}(E_c)$ equals $w(S, \bar{S})$ (where $(S, \bar{S})$ is maximum cut). From inequality (6), we get

$$SDP = w(S, \bar{S}) + \frac{1}{2} \sum_{(u,v) \in E_d} \mathrm{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2 - \frac{1}{2} \sum_{(u,v) \in E_c} \mathrm{dem}_{uv} \|\bar{u} - \bar{v}\|^2 > w(S, \bar{S}).$$

We conclude that the optimal SDP solution has value at least SDP, which is greater than $w(S, \bar{S})$. Therefore, the SDP relaxation is not integral. $\square$

5.4   Hardness Result for Max k-Cut

In this section, we prove a hardness result for Max k-Cut.

Definition 5.7. *The Max k-Cut problem is to partition a given weighted graph G into k pieces so as to maximize the total weight of cut edges.*

Definition 5.8. *Let us say that an instance $G = (V, E, w)$ of Max k-Cut is ∞-stable if it is γ-stable for every γ. That is, there is a partition $\mathcal{P}$ of V such that for every set of positive weights w', $\mathcal{P}$ is an optimal solution for Max Cut instance $G' = (V, E, w')$.*

Claim 5.9. *For every $k \geq 3$, there is no polynomial-time algorithm that solves ∞-stable instances of Max k-Cut unless $NP = RP$.*

Proof. The claim easily follows from the hardness result for the Unique k-Coloring problem by Barbanchon [7]. Recall that a graph G is uniquely k colorable if there exists exactly one proper coloring of G in k colors (up to permutation of the colors). Barbanchon [7] showed[1] that there is no polynomial algorithm that given a uniquely k-colorable graph finds its k coloring unless $NP = RP$.

   Let G be a uniquely k-colorable graph. We assign each edge of G weight 1 and obtain an instance of Max k-Cut. We show that the instance is ∞-stable. Let $\mathcal{P}$ be the partition corresponding to the unique k-coloring $\mathcal{C}$ of G. Note that no matter what positive weights we assign to edges, the value of $\mathcal{P}$ equals the total weight of all edges in the graph (since $\mathcal{P}$ cuts all edges). Thus $\mathcal{P}$ is an optimal k-partition. Moreover, $\mathcal{P}$ is the only optimal partition. Indeed if a k-partition $\mathcal{P}'$ cuts all edges, then the coloring that colors every piece in $\mathcal{P}'$ in its own color is a proper k-coloring, and thus it is equal to $\mathcal{C}$ (up to permutation of the colors). The result of Barbanchon implies that there is no polynomial-time algorithm that finds the optimal Max k-Cut in G unless $NP = RP$. □

6   Weakly Stable Instances

6.1   Weakly Stable Instances of Max Cut

In this section, we define a relaxed notion of stability, which we call weak stability, and give an algorithm for approximately solving weakly stable instances of Max Cut. We note that Awasthi, Blum and Sheffet [3] and Balcan and Liang [5] studied a very closely related notion of *perturbation resilience* for the k-Median Clustering problem.

Definition 6.1. *Consider a weighted graph G. Let $(S, \bar{S})$ be a maximum cut in G, N be a set of cuts that contains $(S, \bar{S})$, and $\gamma \geq 1$. We say that G is a (γ, N)-weakly stable instance of Max Cut if for every cut $(T, \bar{T}) \notin N$:*

$$w(E(S, \bar{S}) \setminus E(T, \bar{T})) > \gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S})).$$

   The notion of weak stability generalizes the notion of stability: an instance is γ-stable if and only if it is $(\gamma, \{(S, \bar{S})\})$-weakly stable. We think of the set N in the definition of weak stability as a neighborhood of the maximum cut $(S, \bar{S})$; it contains cuts that are "close enough" to $(S, \bar{S})$.

---

[1] Barbanchon [7] states his result only for $k = 3$. The result for $k > 3$ follows from his result as follows. For a graph G, let G' be the union of graphs G and K_{k-3} in which every vertex of G is connected with every vertex of K_{k-3}. Then G is uniquely 3-colorable if and only if G' is uniquely k-colorable.

Intuitively, the definition requires that every cut that is sufficiently different from $(S, \bar{S})$ is much smaller then $(S, \bar{S})$, but does not impose any restrictions on cuts that are close to $(S, \bar{S})$. One natural way to define the neighborhood of $(S, \bar{S})$ is captured in the following definition.

Definition 6.2. *Consider a weighted graph G. Let $(S, \bar{S})$ be a maximum cut in G, $\delta \geq 0$, and $\gamma \geq 1$. We say that G is a (γ, δ)-weakly stable instance of Max Cut if G is $(\gamma, \{(S', \bar{S}') : |S \Delta S'| \leq \delta n\})$-weakly stable. In other words, G is (γ, δ)-weakly stable if for every cut $(T, \bar{T})$ such that $|S \Delta T| > \delta n$ and $|S \Delta \bar{T}| > \delta n$, we have*

$$w(E(S, \bar{S}) \setminus E(T, \bar{T})) > \gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S})).$$

The main result of this section is the following theorem.

Theorem 6.3. *There is a polynomial-time algorithm that given a (γ, N)-stable instance of Max Cut, returns a cut from N if $\gamma \geq c\sqrt{\log n} \log \log n$ (for some absolute constant c). (The set N is not part of the input and is not known to the algorithm.)*

The algorithm starts with an arbitrary cut and then iteratively improves it. We now describe a subroutine that algorithm runs in each iteration.

Lemma 6.4. *There is a polynomial-time algorithm $\mathcal{A}$ for the following task. Let $G = (V, E, w)$ be a (γ, N)-weakly stable instance of Max Cut for some N and $\gamma \geq c\sqrt{\log n} \log \log n$ (where c is an absolute constant), $(S, \bar{S})$ be the optimal cut, and $(T, \bar{T}) \notin N$ be a cut in G. The algorithm $\mathcal{A}$ given the graph G, the cut $(T, \bar{T})$ and a parameter $\omega \in \mathbb{R}^+$ either returns a cut $(T', \bar{T}')$ such that $w(T', \bar{T}') \geq w(T, \bar{T}) + \omega$, or returns $\bot$. The algorithm always returns a cut $(T', \bar{T}')$ if $w(E(S, \bar{S}) \setminus E(T, \bar{T})) \geq 4m\omega$ (where $m = |E|$).*

Proof. We construct an auxiliary Sparsest Cut instance $\mathcal{I}$ on V defined by

$$E_c = E(T, \bar{T}), \qquad\qquad\qquad \text{cap}_{uv} = w(u, v)$$
$$E_d = \left\{(u, v) \in E \setminus E(T, \bar{T}) : w(u, v) \geq 2\omega\right\}, \qquad \text{dem}_{uv} = w(u, v).$$

Then we run the approximation algorithm for Sparsest Cut by Arora, Lee and Naor and find an approximate cut $(A, \bar{A})$. We let $T' = (T \cap A) \cup (\bar{T} \cap \bar{A})$ (see Figure 2). If $w(T', \bar{T}') \geq w(T, \bar{T}) + \omega$ then we return $(T', \bar{T}')$, otherwise we return $\bot$.

Whenever the algorithm returns a cut, the cut satisfies the requirement $w(T', \bar{T}') \geq w(T, \bar{T}) + \omega$. Thus we only need to prove that if $w(E(S, \bar{S}) \setminus E(T, \bar{T})) \geq 4m\omega$ then the algorithm finds a cut.

First we show that there is a Sparsest Cut with sparsity at most $2/\gamma$ in $\mathcal{I}$. Let $A^* = (S \cap T) \cup (\bar{S} \cap \bar{T})$. Since $(T, \bar{T}) \notin N$, we have $w(E(S, \bar{S}) \setminus E(T, \bar{T})) > \gamma \cdot w(E(T, \bar{T}) \setminus E(S, \bar{S}))$, or equivalently $w((E \setminus E(T, \bar{T})) \cap E(A^*, \bar{A}^*)) > \gamma \cdot w(E(T, \bar{T}) \cap E(A^*, \bar{A}^*))$. Thus

$$\text{dem}(E_d \cap E(A^*, \bar{A}^*)) \geq w((E \setminus E(T, \bar{T})) \cap E(A^*, \bar{A}^*)) - m \cdot 2\omega \geq w((E \setminus E(T, \bar{T}))/2$$
$$> \gamma \cdot w(E(T, \bar{T}) \cap E(A^*, \bar{A}^*))/2 = (\gamma/2) \cdot \text{cap}(E_c).$$

We get that $\phi(A^*) < 2/\gamma$. Therefore, our algorithm finds a cut A with $\phi(A) \leq O(\sqrt{\log n} \log \log n) \cdot 2/\gamma < 1/2$. We have

$$w(T', \bar{T}') - w(T, \bar{T}) = w(E(T', \bar{T}') \setminus E(T, \bar{T})) - w(E(T, \bar{T}) \setminus E(T', \bar{T}'))$$
$$\geq \text{dem}(E_d \cap E(A, \bar{A})) - w(E_c \cap E(A, \bar{A})) \geq \text{dem}(E_d \cap E(A, \bar{A}))/2 \geq \omega.$$

We used that the demand of every pair in E_d is at least 2ω, and hence $\text{dem}(E_d \cap E(A, \bar{A}))/2 \geq \omega$. $\quad\square$
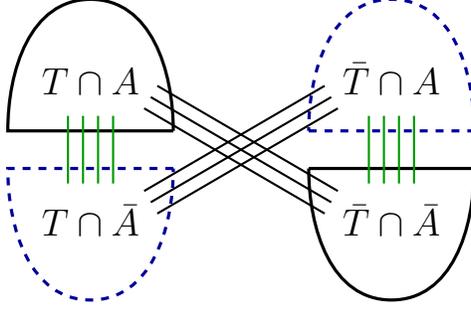
Figure 2: This figure shows the Sparsest Cut instance $\mathcal{I}$ and cut $(A, \bar{A})$. Demand pairs separated by $(A, \bar{A})$ are shown by vertical segments, capacity edges cut by $(A, \bar{A})$ are shown by diagonal segments. Set T' consists of vertices in semicircles in the upper-left and lower-right corners.

Now we are ready to prove Theorem 6.3.

Proof of Theorem 6.3.
Algorithm. We start with an arbitrary cut $(T, \bar{T})$. Then we iteratively run the algorithm $\mathcal{A}$ from Lemma 6.4. In each iteration, we go over all values of ω in $\Omega = \{w(u, v)/(4m) : (u, v) \in E\}$ in the descending order, and execute $\mathcal{A}$ on input G, $(T, \bar{T})$ and ω. If $\mathcal{A}$ finds a cut $(T', \bar{T}')$, we let $T = T'$ and start a new iteration. If $\mathcal{A}$ does not find any cut, we stop and output $(T, \bar{T})$.

Analysis. We first show that the algorithm always returns a cut from N. At every step of the algorithm when $(T, \bar{T}) \notin N$, we have $w(S, \bar{S}) > w(T, \bar{T})$, hence $E(S, \bar{S}) \setminus E(T, \bar{T}) \neq \varnothing$ and $w(E(S, \bar{S}) \setminus E(T, \bar{T})) \geq \min_{e \in E} w(e)$. Therefore, for some $\omega \in \Omega$ (in particular, for $\omega = \frac{\min_{e \in E} w(e)}{4m}$; see the statement of Lemma 6.4), the algorithm $\mathcal{A}$ finds a better cut $(T', \bar{T}')$, and the main algorithm does not terminate. It remains to check that the running time is polynomial.

Consider one iteration of the algorithm. Let $(T, \bar{T})$ be the current cut. Let (u, v) be the heaviest edge in $E(S, \bar{S}) \setminus E(T, \bar{T})$ and $\omega^* = w(u, v)/(4m)$. Note that in this iteration we find a cut when we run $\mathcal{A}$ with some $\omega \geq \omega^*$ (since if we do not find a cut $(T', \bar{T}')$ when $\omega > \omega^*$, we must find a cut $(T', \bar{T}')$ when $\omega = \omega^*$ by Lemma 6.4). We also have

$$w(E(S, \bar{S})) - w(E(T, \bar{T})) \leq w(E(S, \bar{S}) \setminus E(T, \bar{T})) \leq |E(S, \bar{S}) \setminus E(T, \bar{T})| \cdot (4m\omega^*) \leq 4m^2\omega.$$

We charge this iteration to "level" ω. We show that every $\omega \in \Omega$ pays for at most $4m^2$ iterations and therefore the number of iterations is $O(m^3)$.

Indeed, consider $\omega \in \Omega$. Let T_0 be the value of T just before we perform an iteration at level ω for the first time, and T_k be the value of T right after we perform k iterations at level ω (possibly we perform iterations at other levels in between). We have,

$$w(E(T_k, \bar{T}_k)) \geq w(E(T_0, \bar{T}_0)) + k\omega \geq w(E(S, \bar{S})) - 4m^2\omega + k\omega.$$

Since $w(E(T_k, \bar{T}_k)) \leq w(E(S, \bar{S}))$, we get that $k \leq 4m^2$. This concludes the proof. $\qquad\square$

6.2 Weakly Stable Instances of Minimum Multiway Cut

In this section, we give an algorithm for approximately solving weakly stable instances of Minimum Multiway Cut.

Definition 6.5. *Consider a weighted graph G. Let $\mathcal{S}^* = (S_1^*, \ldots, S_k^*)$ be a minimum multiway cut in G, N be a set of multiway cuts that contains $\mathcal{S}^*$, and $\gamma \geq 1$. We say that G is a (γ, N)-weakly stable instance of Minimum Multiway Cut if for every multiway cut $\mathcal{S}' = (S_1', \ldots, S_k') \notin N$:*

$$w(E' \setminus E^*) > \gamma \cdot w(E^* \setminus E'),$$

where E^* is the set of edges cut by $\mathcal{S}^*$ and E' is the set of edges cut by $\mathcal{S}'$.

The main result of this section is the following theorem.

Theorem 6.6. *There is a polynomial-time algorithm for the following task. Given a $(4, N)$-stable instance of Minimum Multiway Cut with integer edge weights in the range $[1, poly(n)]$, the algorithm returns a multiway cut from N. The set N is not part of the input and is not known to the algorithm.*

If the weights are not polynomially bounded, the following version of this theorem holds. (The proofs of Thereoms 6.6 and 6.7 are very similar. For simplicity of exposition, we only present the proof of Theorem 6.6.)

Theorem 6.7. *There is a polynomial-time algorithm that given a $(4 + \varepsilon, N)$-stable instance of Minimum Multiway Cut, returns a multiway cut from N. The running time of the algorithm is inversely proportional do ε. The set N is not part of the input and is not known to the algorithm.*

The algorithm starts with an arbitrary multiway cut $\mathcal{S}$ and then iteratively improves it.

Lemma 6.8. *There is a polynomial-time algorithm that given a $(4, N)$ weakly stable instance $G = (V, E, w)$ and a solution $\mathcal{S}^\circ$ either finds a solution $\mathcal{S}'$ of smaller cost or certifies that $\mathcal{S}^\circ \in N$.*

Proof. Let E° be the set of edges cut by $\mathcal{S}^\circ$. Define edge weights $w'(u, v)$ by

$$w'(u, v) = \begin{cases} w(u, v), & \text{if}(u, v) \in E^\circ, \\ 4\,w(u, v), & \text{otherwise.} \end{cases}$$

We solve the LP relaxation (2) for Multiway Cut with weights $w'(u, v)$. Let $\{\bar{u}\}$ be the LP solution. Consider the distribution of random cuts $\mathcal{S}' = (S_1', \ldots, S_k')$ from Lemma 4.4. Let E' be the set of edges cut by $\mathcal{S}'$. Similarly to the proof of Theorem 4.6, we define

$$\mathsf{LP}_+ = \sum_{(u,v) \in E^\circ} w'(u, v)(1 - d(u, v)) = \sum_{(u,v) \in E^\circ} w(u, v)(1 - d(u, v)),$$

$$\mathsf{LP}_- = \sum_{(u,v) \in E \setminus E^\circ} w'(u, v)\, d(u, v) = 4 \sum_{(u,v) \in E \setminus E^\circ} w(u, v)\, d(u, v).$$

We have,

$$\mathbb{E}\left[w(E^\circ \setminus E')\right] = \sum_{(u,v) \in E^\circ} w(u, v)\Pr((u, v) \text{ is not cut}) \geq \sum_{(u,v) \in E^\circ} w(u, v)(1 - d(u, v))/2 = \mathsf{LP}_+/2,$$

$$\mathbb{E}\left[w(E' \setminus E^\circ)\right] = \sum_{(u,v) \in E \setminus E^*} w(u, v)\,\Pr((u, v) \text{ is cut}) \leq 2 \sum_{(u,v) \in E^\circ} w(u, v)d(u, v) = \mathsf{LP}_-/2.$$

Therefore,

$$\mathbb{E}\left[w(E^\circ) - w(E')\right] = \mathbb{E}\left[w(E^\circ \setminus E')\right] - \mathbb{E}\left[w(E' \setminus E^\circ)\right] = \frac{\mathsf{LP}_+ - \mathsf{LP}_-}{2}.$$

We have,

$$\mathsf{LP}_+ - \mathsf{LP}_- = w'(E^\circ) - \sum_{(u,v)\in E} w'(u,v)\, d(u,v) \geq w'(E^\circ) - w'(E^*) = w'(E^\circ \setminus E^*) - w'(E^* \setminus E^\circ),$$

since the LP value of solution $\{\bar{u}\}$ is at most the value of solution E^* (of the multiway instance with weights w'). Now if $\mathcal{S}^\circ \notin N$ then

$$\mathsf{LP}_+ - \mathsf{LP}_- = w'(E^\circ \setminus E^*) - w'(E^* \setminus E^\circ) = w(E^\circ \setminus E^*) - 4w(E^* \setminus E^\circ) > 0.$$

Therefore, $\mathbb{E}\left[w(E') - w(E^\circ)\right] < 0$ and for some multiway cut in the distribution, we have $w(E') < w(E^\circ)$. Note that we can efficiently go over all multiway cuts in the support of the distribution. If we find a multiway cut with $w(E') < w(E^\circ)$, we return it; otherwise, we output that $\mathcal{S}^\circ \in N$. □

Proof of Theorem 6.6. We start with an arbitrary feasible multiway cut $\mathcal{S}^\circ$ and iteratively improve it using the algorithm from Lemma 6.8. Once the algorithm returns that the current cut $\mathcal{S}^\circ$ lies in N, we output it. Since the cost of the multiway cut decreases by at least 1 in each iteration, and the initial cost of $\mathcal{S}^\circ$ is polynomial in n, the algorithm terminates after polynomially many steps. □

7 Discussion

In this paper, we presented algorithms for stable instances of Max Cut and Minimum Multiway Cut. In conclusion, we briefly discuss what properties of these problems we used. We provide a sufficient condition under which there is an algorithm for stable instances of a graph partitioning problem.

Consider a graph partitioning problem. Our goal is to partition a graph into several pieces, subject to certain constraints, so as to minimize or maximize the weight of cut edges. Consider a metric relaxation for this problem. The relaxation defines a metric $d(\cdot, \cdot)$ on the set of vertices. A combinatorial solution to the problem corresponds to a multicut metric $d(u, v)$: the distance between vertices in one piece is 0, the distance between vertices in different pieces is 1. For Max Cut and Multiway Cut, we proved that *the metric relaxation is integral when the instance is sufficiently stable*; this, in turn, implied the existence of polynomial-time robust algorithms for stable instances of these problems. We summarize the properties that we used in the proof in the following meta-theorem.

Theorem 7.1 (Meta-theorem). *Consider a graph partitioning problem and a metric relaxation for it. Suppose that there is a rounding scheme that given a graph $G = (V, E, w)$ and a metric $d(\cdot, \cdot)$ returns a feasible partition such that for some $\alpha \geq 1$ and $\beta \geq 1$:*

> **For a cut minimization problem,**
> 1. $\Pr\left(u \text{ and } v \text{ are separated}\right) \leq \alpha d(u, v),$
> 2. $\Pr\left(u \text{ and } v \text{ are not separated}\right) \geq \beta^{-1}(1 - d(u, v)).$

For a cut maximization problem,

1′. $\Pr\left(u \text{ and } v \text{ are separated}\right) \geq \alpha^{-1}d(u,v)$

2′. $\Pr\left(u \text{ and } v \text{ are not separated}\right) \leq \beta(1 - d(u,v))$

Then the metric relaxation is integral for $(\alpha\beta)$-stable instances of the problem. Consequently, there is a robust polynomial-time algorithm for $(\alpha\beta)$-stable instances (if the relaxation is polynomial-time solvable). Moreover, there is an algorithm for $(\alpha\beta+\varepsilon)$-weakly stable instances of the problem. (The meta-theorem also holds for a cut maximization/minimization problem with positive and negative weights. Then we require that all four properties 1, 1′, 2 and 2′ hold.)

The proof of this meta-theorem repeats the proofs of Theorems 3.1 and 4.6. Note that if a rounding scheme just satisfies property 1 or 1′ then there is an α approximation algorithm for the problem. However, properties 1 and 1′ alone do not imply that the relaxation is integral. For example, there is a rounding scheme for Max k-Cut satisfying 1′, but there is no algorithm for stable instances of Max k-Cut (see Claim 5.9). Another example is Minimum Multicut. There is a rounding scheme for the standard LP relaxation of Minimum Multicut with $\alpha = O(\log n)$ [24]. However, this relaxation is not integral even for $(n-2-\varepsilon)$-stable instances of the problem (for every $\varepsilon > 0$). Indeed, consider an instance on $(n-1)$ terminals $s_1, \ldots, s_{n-1}$ and one extra vertex u; u is connected with s_1 by an edge of weight $n-2-\varepsilon/2$ and with all other terminals by edges of weight 1. We need to separate every pair of terminals s_i and s_j. This instance is $(n-2-\varepsilon)$-stable. However, the optimal LP solution is not integral: it assigns $d(u,s_i) = 1/2$ (for every i) and $d(s_i,s_j) = 1$ (for every $i \neq j$).

References

[1] Sanjeev Arora, James R. Lee, and Assaf Naor. Euclidean distortion and the sparsest cut. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 553–562, 2005.

[2] Yonatan Aumann and Yuval Rabani. An $O(\log k)$ approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[3] Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Inf. Process. Lett.*, 112(1-2):49–54, January 2012.

[4] Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Approximate clustering without the approximation. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1068–1077, 2009.

[5] Maria-Florina Balcan and Yingyu Liang. Clustering under perturbation resilience. In *Proceedings of the International Colloquium on Automata, Languages and Programming*, pages 63–74, 2012.

[6] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1):89–113, 2004.

[7] Régis Barbanchon. On unique graph 3-colorability and parsimonious reductions in the plane. *Theoretical computer science*, 319(1):455–482, 2004.

[8] Yonatan Bilu, Amit Daniely, Nati Linial, and Michael Saks. On the practically interesting instances of maxcut. In *Proceedings of the Symposium on Theoretical Aspects of Computer Science*, pages 526–537, 2013.

[9] Yonatan Bilu and Nathan Linial. Are stable instances easy? In *Innovations in Computer Science*, pages 332–341, 2010.

[10] Niv Buchbinder, Joseph Seffi Naor, and Roy Schwartz. Simplex partitioning via exponential clocks and the multiway cut problem. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing*, pages 535–544, 2013.

[11] Moses Charikar and Anthony Wirth. Maximizing quadratic programs: extending Grothendieck's inequality. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 54–60, 2004.

[12] Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating Multicut and Sparsest-Cut. *Comput. Complexity*, 15(2):94–114, 2006.

[13] Jeff Cheeger, Bruce Kleiner, and Assaf Naor. A $(\log n)^{\Omega(1)}$ integrality gap for the sparsest cut SDP. In *Proceedings of 50th Annual Symposium on Foundations of Computer Science*, pages 555–564, 2009.

[14] Julia Chuzhoy and Sanjeev Khanna. Hardness of cut problems in directed graphs. In *Proceedings of the Symposium on Theory of Computing*, pages 527–536, 2006.

[15] Gruia Călinescu, Howard Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. In *Proceedings of the Symposium on Theory of Computing*, pages 48–52, 1998.

[16] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23:864–894, 1994.

[17] M. R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.

[18] Michel X. Goemans and David P. Williamson. .879-approximation algorithms for MAX CUT and MAX 2SAT. In *Proceedings of the Twenty-Sixth Annual Symposium on Theory of Computing*, pages 422–431, 1994.

[19] Johan Håstad. Some optimal inapproximability results. *J. ACM*, 48(4):798–859, July 2001.

[20] Thomas Holenstein. Parallel repetition: Simplification and the no-signaling case. *Theory of Computing*, 5(1):141–172, 2009.

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

[22] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, April 2007.

[23] Subhash Khot and Nisheeth Vishnoi. The Unique Games Conjecture, integrality gap for cut problems and embeddability of negative type metrics into ℓ_1. In *Proceedings of 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 53–62, 2005.

[24] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, November 1999.

[25] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[26] Konstantin Makarychev, Yury Makarychev, and Aravindan Vijayaraghavan. Approximation algorithms for semi-random partitioning problems. In *Proceedings of the 44th Annual Symposium on Theory of Computing*, pages 367–384, 2012.

[27] Vijay Raghavan and Jeremy Spinrad. Robust algorithms for restricted domains. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms*, pages 460–467, 2001.

[28] L. Trevisan, G.B. Sorkin, M. Sudan, and D.P. Williamson. Gadgets, approximation, and linear programming. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science*, pages 617–626, 1996.

A   Proof of Corollary 5.5

Proof of Lemma 5.5. Consider an integrality gap instance with gap $D_{\ell_2^2 \to \ell_1}(n)$(see Fact 5.4). Denote the set of vertices by V_0, the set of capacity edges by E_c, and edge capacities by cap_{uv}. Similarly, denote the set of demand pairs by E_d, and demands by dem_{uv}. Let $\{\bar{u}\}$ be the optimal SDP solution for the standard SDP relaxation for $\mathcal{I}$. Since the integrality gap is $D_{\ell_2^2 \to \ell_1}(n)$, we have that for every cut $(A, \bar{A} = V_0 \setminus A)$

$$\frac{\mathrm{cap}(E_c(A, \bar{A}))}{\mathrm{dem}(E_d(A, \bar{A}))} \geq D_{\ell_2^2 \to \ell_1} \cdot \frac{\sum_{(u,v) \in E_c} \mathrm{cap}_{uv} \cdot \|\bar{u} - \bar{v}\|^2}{\sum_{(u,v) \in E_d} \mathrm{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2}.$$

By rescaling demands we may assume without loss of generality that

$$\frac{D_{\ell_2^2 \to \ell_1} - \varepsilon}{D_{\ell_2^2 \to \ell_1}} < \frac{\sum_{(u,v) \in E_c} \mathrm{cap}_{uv} \cdot \|\bar{u} - \bar{v}\|^2}{\sum_{(u,v) \in E_d} \mathrm{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2} < 1.$$

Then we have (5).

We now transform vectors $\bar{u}$ to unit vectors $\tilde{u}$ such that $\{\pm\tilde{u}\}$ is an ℓ_2^2 space and vectors $\tilde{u}$ satisfy inequality (6). Choose a sufficiently small $\delta > 0$ (which we specify later) and let $\hat{u}' = \hat{u} + \delta \cdot \bar{e}_u$, where vectors $\bar{e}_u$ are unit vectors orthogonal to each other and all vectors $\bar{v}$. All vectors $\bar{u}'$ are in general position (that is, no $r + 2$ vectors lie in an r dimensional affine subspace). Therefore all vectors $\bar{u}'$ lie on some sphere. Denote its center by $\bar{c}$ and radius by R. Let $\bar{z}$ be a unit vector orthogonal to all vectors $\bar{u}'$. Finally, define vectors $\tilde{u}$,

$$\tilde{u} = \frac{\sqrt{3}\,\bar{z}}{2} + \frac{\bar{u}' - \bar{c}}{2R}.$$

Note that vectors $(\bar{u}' - \bar{c})/R$ are unit vectors, and therefore vectors $\tilde{u}$ are also unit vectors. Now

$$\|\tilde{u} - \tilde{v}\|^2 = \left\| \frac{\bar{u}' - \bar{v}'}{2R} \right\|^2 = \frac{\|\bar{u} - \bar{v}'\|^2 + 2\delta^2}{4R^2}.$$

Therefore, when $\delta \to 0$,

$$\frac{\sum_{(u,v) \in E_c} \operatorname{cap}_{uv} \cdot \|\tilde{u} - \tilde{v}\|^2}{\sum_{(u,v) \in E_d} \operatorname{dem}_{uv} \cdot \|\tilde{u} - \tilde{v}\|^2} = \frac{\sum_{(u,v) \in E_c} \operatorname{cap}_{uv} \cdot \|\bar{u} - \bar{v}\|^2}{\sum_{(u,v) \in E_d} \operatorname{dem}_{uv} \cdot \|\bar{u} - \bar{v}\|^2} + O(\delta^2).$$

We choose $\delta > 0$ so that

$$\frac{\sum_{(u,v) \in E_c} \operatorname{cap}_{uv} \cdot \|\tilde{u} - \tilde{v}\|^2}{\sum_{(u,v) \in E_d} \operatorname{dem}_{uv} \cdot \|\tilde{u} - \tilde{v}\|^2} < 1.$$

We get that inequality (6) holds. Finally, we verify that vectors $\{\pm\tilde{u}\}$ form an ℓ_2^2 space. We have,

$$\|\tilde{u} - \tilde{v}\|^2 + \|\tilde{v} - \tilde{w}\|^2 = \frac{\|\bar{u} - \bar{v}\|^2 + \|\bar{v} - \bar{w}\|^2 + 4\delta^2}{4R^2} \geq \frac{\|\bar{u} - \bar{w}\|^2 + 4\delta^2}{4R^2} > \|\tilde{u} - \tilde{w}\|^2,$$

$$\|\tilde{u} - \tilde{v}\|^2 + \|\tilde{v} + \tilde{w}\|^2 = \|\tilde{u} - \tilde{v}\|^2 + (4 - \|\tilde{v} - \tilde{w}\|^2)$$
$$> (\|\tilde{u} - \tilde{w}\|^2 + \|\tilde{w} - \tilde{v}\|^2) + (4 - \|\tilde{v} - \tilde{w}\|^2) = \|\tilde{u} + \tilde{w}\|^2,$$

$$\|\tilde{u} + \tilde{v}\|^2 + \|\tilde{v} + \tilde{w}\|^2 = 4 - \|\tilde{u} - \tilde{v}\|^2 + 4 - \|\tilde{v} - \tilde{w}\|^2 \geq 8 - 2 > \|\tilde{u} - \tilde{w}\|^2.$$

In the the last line, we used that $\|\tilde{u} - \tilde{v}\|^2 = \left\| \frac{\bar{u}' - \bar{c}}{2R} - \frac{\bar{v}' - \bar{c}}{2R} \right\|^2 \leq 1$ for every u and v. $\qquad\square$

B   Remark on Correlation Clustering

In this section, we briefly describe how our results extend to stable instances of the Correlation Clustering problem. The Correlation Clustering problem was introduced by Bansal, Blum, and Chawla [6] and later studied by Charikar and Wirth [11], and others. Our positive results for stable and weakly stable instances of Max Cut also apply to stable and weakly stable instances of 2-Correlation Clustering. Our negative result when Max k-Cut shows that there is no exact polynomial-time algorithm for ∞-stable instances of k-Correlation Clustering for $k \geq 3$ unless $RP = NP$.

Definition B.1. *An instance of the k-Correlation Clustering problem is a weighted graph $G = (V, E, w)$ in which every edge is labeled with either "+" or "−". We denote the set of edges labeled with "+" by E^+ and the set of edges labeled with "−" by E^-. Consider a clustering $\mathcal{C}$ of V into k disjoint clusters. For every $u \in V$, let $\mathcal{C}(u)$ be the cluster that u belongs to. We define the total weight of agreements and disagreements as follows:*

$$\mathsf{Agree}_G(\mathcal{C}) = \sum_{\substack{(u,v) \in E^+ \\ \mathcal{C}(u) = \mathcal{C}(u)}} w_{(u,v)} + \sum_{\substack{(u,v) \in E^- \\ \mathcal{C}(u) \neq \mathcal{C}(u)}} w_{(u,v)}$$

$$\mathsf{DisAgree}_G(\mathcal{C}) = \sum_{\substack{(u,v) \in E^+ \\ \mathcal{C}(u) \neq \mathcal{C}(u)}} w_{(u,v)} + \sum_{\substack{(u,v) \in E^- \\ \mathcal{C}(u) = \mathcal{C}(u)}} w_{(u,v)}.$$

The Correlation Clustering problems asks to find a clustering of V into k clusters that maximizes the total weight of agreements, $\mathsf{Agree}_G(\mathcal{C})$.

We note that different variants of the problem have been studied in the literature.

variant	objective
Maximum Agreement	maximize $\mathsf{Agree}_G(\mathcal{C})$
Minimum Disagreement	minimize $\mathsf{DisAgree}_G(\mathcal{C})$
Maximum Correlation	maximize $\mathsf{Agree}_G(\mathcal{C}) - \mathsf{DisAgree}_G(\mathcal{C})$

A good *approximate* solution for one variant is not necessarily a good approximate solution for the other variants. However, an *optimal* solution for one variant is also an optimal solution for all other variants. Thus an instance of Correlation Clustering is γ-stable w.r.t. one objective if and only if it is γ-stable w.r.t. each of them. Since in this paper we study exact algorithms for γ-stable instances, all three variants of the problem are equivalent for our purposes. We will assume that our objective is to maximize $\mathsf{Agree}_G(\mathcal{C})$.

Positive Results. Our positive results for stable and weakly stable instances of Max Cut also apply to stable and weakly stable instances of 2-Correlation Clustering. The proofs of Theorem 3.1, Corollary 3.2, and Theorem 6.3 can very easily be modified to deal with the 2-Correlation Clustering problem. We do not describe the necessary modifications in this paper. Instead, we point out that there is a simple reduction that maps γ-stable instances of 2 Correlation Clustering to γ-stable instances of Max Cut, and weakly stable instances to weakly stable instances. Therefore, every algorithm for solving γ-stable or γ-weakly stable instances of Max Cut can be used to solve γ-stable or γ-weakly stable instances of 2-Correlation Clustering. We now briefly describe the reduction. Given a graph $G = (V, E^+ \cup E^-, w)$, the reduction constructs a graph $G' = (V', E', w')$ with

$$V' = \{u : u \in V\} \cup \{u' : u \in V\},$$
$$E' = \{(u,v) : (u,v) \in E^-\} \cup \{(u',u) : u \in V\} \cup \{(u',v), (u,v') : (u,v) \in E^+\},$$
$$w'(u,v) = w(u,v), \quad w'(u,v') = w(u,v)/2, \quad w(u,u') = W_\infty,$$

where W_∞ is large enough (e.g. $W_\infty = 2\gamma \sum_{e \in E} w_e$). Since the weight of edges (u,u') is very large, every maximum cut in G' cuts all edges (u,u'), even if we increase some edge weights by a factor at most γ. For every 2-clustering $(S, \bar{S})$ of G, consider the corresponding cut $(S', \bar{S}')$ in G'

$$S' = \{u : u \in S\} \cup \{u' : u \in \bar{S}\} \quad \bar{S}' = \{u : u \in \bar{S}\} \cup \{u' : u \in S\}.$$

Then $\mathsf{Agree}_G((S, \bar{S})) = w'(E'(S', \bar{S}')) - nW_\infty$ (note that the term nW_∞ does not depend on S). We get that $(S, \bar{S})$ is an optimal 2-clustering if and only if $(S', \bar{S}')$ is a maximum cut in G'.

Negative Results. The Max k-Cut problem is a special case of the k-Correlation Clustering problem, in which all edges are labeled with "-". Therefore, the result of Theorem 5.9 applies to the k-Correlation Clustering problem when $k \geq 3$.