CSA E0 249: Approximation Algorithms	March 27, 2015

Lecture 12

Instructor: Arnab Bhattacharyya | *Submitted by: Ajith S and Dheeraj Ram*

1 Graph Colouring

Chromatic number $\chi(G)$ of a graph denotes the smallest number of colors needed to color the vertices of a graph G so that no two neighboring vertices receive the same color. χ(G) is hard to approximate upto $n^{1-\varepsilon}$ $\forall\varepsilon > 0$.

1.1 Min-K Colouring Problem

We know that, to find the colouring, given G and $\chi(G)$ is an NP-Hard problem. So our aim is to find the least number of colors, with which we can color the graph. Given a graph G and $\chi(G) = k$, we have to find a proper colouring in polynomial time, with smallest number of colours (It is NP-Hard to colour a 3-colourable graph with 4 colours). Obviously, every n-vertex graph can be colored with n colors. But this is a simple way of using significantly fewer colors, namely, $O(\sqrt{n})$ (Wigdersons trick):

1. If G is 3-colorable, then the neighborhood of a vertex is bipartite and thus it can be 2-colored (fast).
2. So, given a parameter Δ, by repeatedly coloring maximum-degree vertices and their neighbors, we can use $O(n/\Delta)$ colors and reduce the maximum degree below Δ.
3. A graph with maximum degree Δ can easily be colored by $\Delta + 1$ colors (greedy algorithm give each vertex a color not used by its neighbors).
4. Combining these two methods and setting $\Delta := \sqrt{n}$, we can color every 3-colorable G with $O(\sqrt{n})$ colors.

This barrier ($O(\sqrt{n})$) was finally broken using SDP by Karger et al. [KMS 98]: $O(n^{0.25})$ colors suffice this is what we are now going to present. (Current best is $O(n^{0.2072})$ was obtained by Chlamtac [Chl07], who used an SDP relaxation at a higher level of the Lasserre hierarchy).

1.1.1 Lovasz θ function - $\theta(G)$

Consider the following SDP.

$$
\begin{aligned}
\text{minimize} \quad & t \\
\text{subject to} \quad & v_i.v_j \qquad , \forall (i,j) \in E \\
& v_i.v_i = 1, \forall i \\
& v_i \in \mathbb{R}^n
\end{aligned}
$$

Let $\theta(G)$ be the solution from the above SDP Problem.

Theorem 1 $\theta(G) \leq -\dfrac{1}{\chi(G) - 1}$

This immediately implies, $\chi(G) \leq 1 - \dfrac{1}{\theta(G)}$. So $W(G) \leq \theta(G) \leq \chi(G)$ where W(G) is the maximum clique number of the graph G.

Proof We use the method of proof by Mathematical Induction.

Basis step: Let G have three vertices. If all are connected (triangle), then the three vectors will be oriented in such a way that the angle between any two vectors is 120°. So $cos(120) \leq -\frac{1}{3-1}$, which is true. If only two are connected, we will get $\chi(G)$ as 2, and two vectors points to opposite directions (180°), still satisfying the theorem.

Inductive step: We assume the theorem holds good for k vertex graph G. So $v_1, v_2, \ldots, v_k$ such that $\forall i \neq j, v_i.v_j \leq -\frac{1}{k-1}$. We have to prove that for a k+1 vertex graph, $v'_1, v'_2, \ldots, v'_{k+1}$ s.t. $v'_i, v'_j \leq -\frac{1}{k}$.

Define $v'_{k+1} = (0, 0, .., 1)$ and $1 \leq i \leq k : v'_i = (\sqrt{1 - \frac{1}{k^2}} * v_i, -\frac{1}{k})$.

Here we can see that $|v'_i| = \sqrt{(1 - \frac{1}{k^2})v_i^2 + (\frac{-1}{k})^2} = 1$

Also,

$$
\begin{aligned}
v'_i.v'_j &= (1 - \frac{1}{k^2})v_i.v_j + \frac{1}{k^2} \\
&\leq (1 - \frac{1}{k^2})(\frac{-1}{k-1}) + \frac{1}{k^2} \\
&= \frac{-1}{k-1} + \frac{1}{k^2}(\frac{1}{k-1} + 1) \\
&= \frac{-1}{k-1} + \frac{1}{k^2}\frac{k}{k-1} \\
&= \frac{1}{k-1}(\frac{1}{k} - 1) \\
&= \frac{-1}{k}
\end{aligned}
$$

So in polynomial time, given 3 coloured graph G, we can find $v_1, ..v_n \in \mathbb{R}$ such that $v_i.v_j \leq -\frac{1}{2}\ \forall (i,j) \in E$. ■

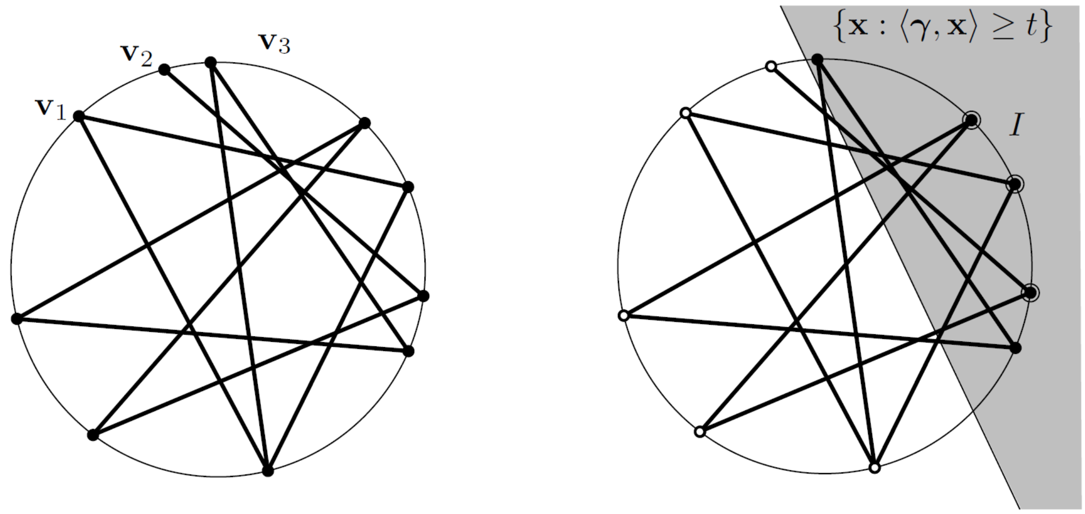

Figure 1: Spherical Clustering in KMS Algorithm [1]

1.2 KMS Algorithm and Analysis

We use a spherical cap to cluster vectors in this algorithm, as shown in the figure 1.

KMS Algorithm

First, we calculate the following.

- Choose vector $g = (g_1, ..g_n)$ where g_i are normal and independent.
- $I = \{v_i : v_i.g \geq s\}$ (s will be decided later)
- $I_0 = \{v \in I : \text{no neighbour of v is in } I\}$

Lemma 2 *If G has max-degree* $\leq \Delta$*, then* $\mathbb{E}[|I_0|] \geq \Omega(\frac{n}{\Delta^{\frac{1}{3}}})$

1. Set $\Delta = n^{\frac{3}{4}}$
2. Use Wigderson's trick to reduce max degree to $n^{\frac{3}{4}}$ using $n^{\frac{1}{4}}$ colours.
3. Repeatedly use above lemma $n^{\frac{1}{4}}$ times.

If the above claim is true, then it is clear that all the vertices will be covered after $n^{\frac{1}{4}}$ iterations of the algorithm. So now we come to the proof of the lemma.

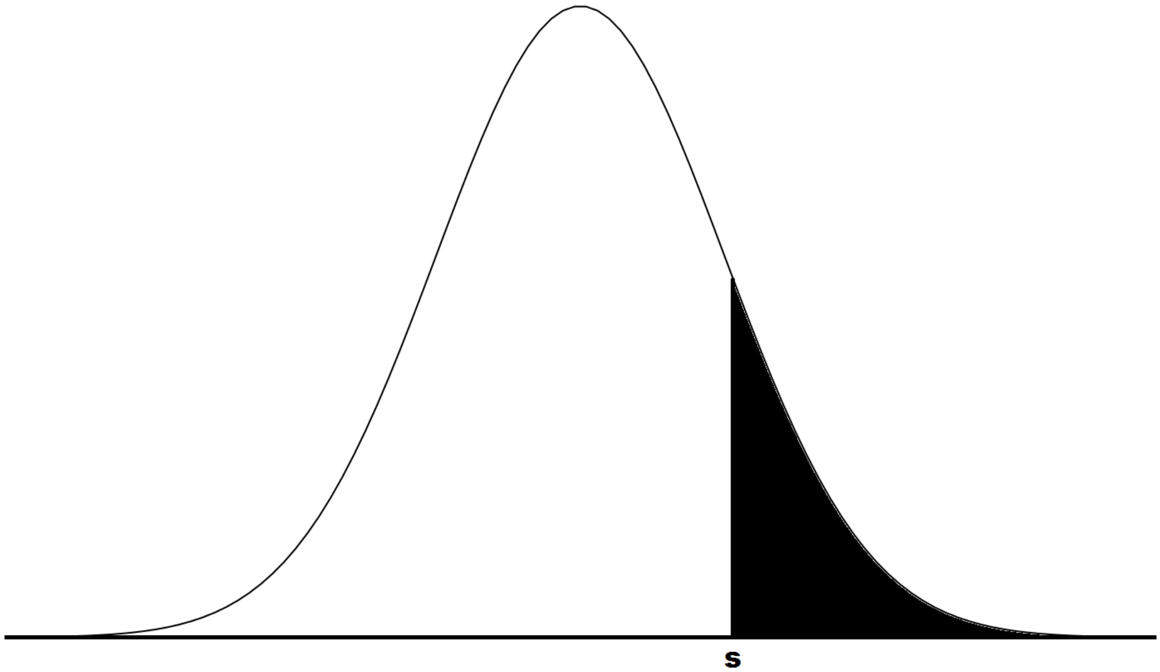

Figure 2: Normal Distribution

Proof We know that the Gaussian structure is spherically symmetric. So, without loss of generality, we can rotate the whole sphere in a way where $v_1 = (1, 0, 0 \ldots 0)$. Now, $\mathsf{Pr}[v_i.g \geq s] = \mathsf{Pr}[g_1 \geq s] = \mathcal{N}(\mathrm{s})$

where $\mathcal{N}(\mathrm{s})$ is the area of the Normal Distribution from $x = s$ to $x = \inf$. The density function $\varphi(x) = \dfrac{1}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$

$$\text{So } \mathcal{N}(s) = \int_s^\infty \varphi(x) dx$$

$$\approx \frac{1}{s}\ e^{\frac{-s^2}{2}}$$

We use the fact that $\mathcal{N}(ks) \approx (\mathcal{N}(s))^{k^2}$. Let u be any neighbour of v.

$$\mathsf{Pr}\,[\ v \notin I_0 \mid v \in I\ ] \ \leq\ \Delta.\mathsf{Pr}\,[\ u.g\ \geq\ s \mid v.g\ \geq\ s\ ]$$

From SDP $u.v \leq -\frac{1}{2}$. So if $v = (0, 1, 0, \ldots 0)$, u can be at most at an angle of $120°$. Let's assume $u = (\frac{\sqrt{3}}{2}, \frac{-1}{2}, 0, \ldots 0)$.

$$
\begin{aligned}
\Pr[\, u.g \geq s \mid v.g \geq s \,] \; &\leq \Pr\left[\; \frac{\sqrt{3}}{2} g_1 - \frac{1}{2} g_2 \;\geq\; s \mid g_2 \;\geq\; s \;\right] \\
&= \Pr\left[\; \frac{\sqrt{3}}{2} g_1 \;\geq\; \frac{3}{2} s \;\right] \\
&= \Pr\left[\; g_1 \;\geq\; \sqrt{3} s \;\right] \\
&\leq \mathcal{N}(s)^3.
\end{aligned}
$$

Now we got

$$\Pr[\, v \notin I_0 \mid v \in I \,] \;\leq\; \Delta \cdot \mathcal{N}(s)^3$$

and,

$$\Pr[\, v \in I \,] = \mathcal{N}(s).$$

So,

$$\Pr[\, v \in I_0 \,] \;\geq\; (1 - \Delta \cdot \mathcal{N}(s)^3) \cdot \mathcal{N}(s)$$

Choose s such that $\mathcal{N}(s)^3 = \dfrac{1}{2 \cdot \Delta}$. Now,

$$
\begin{aligned}
\Pr[\, v \in I_0 \,] \;&\geq\; \frac{1}{2} \cdot \mathcal{N}(s) \\
&=\; \Omega(\Delta^{\frac{-1}{3}})
\end{aligned}
$$

So we get $\mathbb{E}[|I_0|] \geq \Omega(\frac{n}{\Delta^{\frac{1}{3}}})$ by applying Union Bound. This completes the proof of the claim.

■

2 MaxCut - A New Rounding Approach

This section describes a "generic" rounding technique, which attains the integrality gap, up to any given additive constant $\varepsilon > 0$, for a wide class of semidefinite relaxations of constraint satisfaction problems. Assuming the Unique Games Conjecture, this rounding technique almost achieves the best approximation factor attainable by any polynomial time algorithm, for every class of constraint satisfaction problems (Raghavendra [Rag08], Raghavendra and Steurer [RS09b]).

2.1 Miniatures for MaxCut

Here we show Raghavendras rounding technique on the GoemansWilliamson semidefinite relaxation for MaxCut. Let G be an input graph for MaxCut with vertex set $\{1, 2, \dots, n\}$, and let $(\mathrm{v_1}, \dots, \mathrm{v_n})$ be an optimal SDP solution for it. This means that the $\mathrm{v_i}$ are unit vectors maximizing $\sum_{\{i,j\}\in E} \frac{1 - \mathrm{v_i^T v_j}}{2}$.

In the rounding algorithm, we build a *miniature* of the input instance, which is a small graph $\hat{G}$ plus a sequence $(\hat{\mathrm{v}}_1, \dots, \hat{\mathrm{v}}_{\hat{n}})$ of unit vectors. The graph in the miniature is *weighted*. Let $\mathsf{Opt}(G, w)$ denote the maximum weight of a cut in (G, w). The semidefinite relaxation is now:

$$\mathsf{SDP}(G, w) := \max \left\{ \sum_{\{i,j\}\in E} w_{ij} \frac{1 - \mathrm{v_i^T v_j}}{2} \; : ||\mathrm{v_1}|| = \dots = ||\mathrm{v_n}|| = 1 \right\}$$

Let

$$\mathsf{Gap} := \sup_{G,w} \frac{\mathsf{SDP}(G, w)}{\mathsf{Opt}(G, w)}$$

be the worst-case integrality gap of the semidefinite relaxation, over all weighted graphs. So the miniature is a weighted $\hat{n}$-vertex graph $(\hat{G}, \hat{w})$ plus a sequence $\hat{\mathrm{v}}_1, \dots, \hat{\mathrm{v}}_{\hat{n}}$ of unit vectors. The miniature is so small that the maximum cut of $(\hat{G}, \hat{w})$ can be computed exactly by brute force. The number $\hat{n}$ of vertices is a constant, depending on ε.

Claim 3 *SDP* $(\hat{G}, \hat{w}) \geq$ *SDP* $(G) \; - \; \varepsilon|E|$

Proof

$$\begin{aligned} \mathsf{Alg} = \mathsf{Opt}\ (\hat{G}, \hat{w}) &\geq \mathsf{SDP}\ (\hat{G}, \hat{w}) \cdot \mathsf{Gap} \\ &\geq \mathsf{Gap} \cdot (\mathsf{SDP}(G) \; - \; \varepsilon|E|) \\ &\geq \mathsf{Gap} \cdot (\mathsf{Opt}(G) \; - \; \varepsilon|E|) \\ &\geq \mathsf{Opt} \cdot \mathsf{Gap} \cdot (1 \; - \; 2\varepsilon) \end{aligned}$$

■

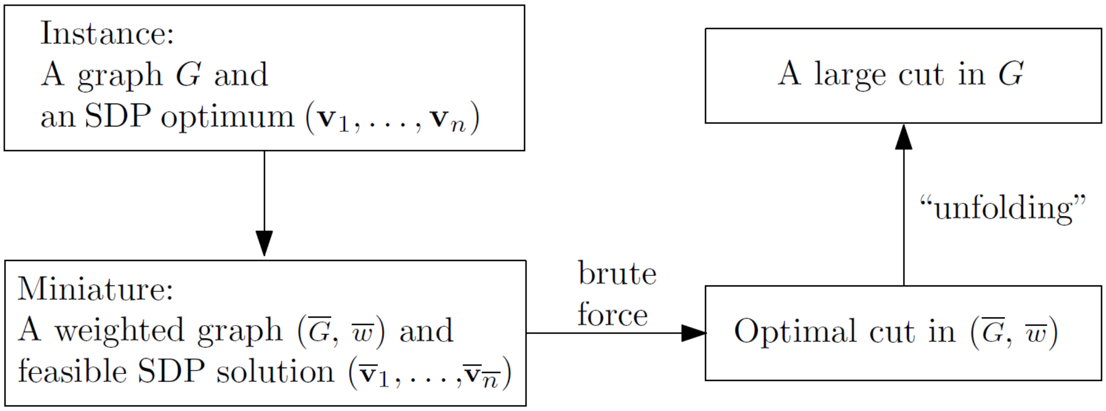

Figure 3: Rounding via a miniature [1]

Two Lemmas for Miniature Builders

The first lemma tells us that unit vectors in S^{d-1} can be "discretized" up to a given error $\delta > 0$.

Lemma 4 *For every d and every $\delta \in \{0, 1\}$, there exists a set $N \subseteq S^{d-1}$ that is δ-dense in S^{d-1} (that is, for every $\mathrm{x} \in \mathrm{S}^{\mathrm{d}-1}$ there exists $\mathrm{z} \in \mathrm{N}$ with $||\mathrm{x} - \mathrm{z}|| < \delta$), and $|N| \leq \left(\frac{3}{\delta}\right)^d$*

Proof We build $N = \{p_1, p_2, \ldots\}$ by a "greedy algorithm": We place p_1 to S^{d-1} arbitrarily, and having already chosen $p_1, \ldots, p_{i-1}$, we place p_i to S^{d-1} so that it has distance at least δ from $p_1, \ldots, p_{i-1}$. This process finishes as soon as we can no longer place the next point, i.e., the resulting set is δ-dense. To estimate the number of points produced in this way, we observe that the balls of radius $\frac{\delta}{2}$ centered at the p_i have disjoint interiors and are contained in the ball of radius $1 + \frac{\delta}{2} \leq \frac{3}{2}$ around 0. Thus, the sum of volumes of the small balls is at most the volume of the large ball, and this gives the lemma. ■

Let Φ be a "normalized random linear map" defined as $\Phi : \mathbb{R}^n \to \mathbb{R}^d$. We choose d independent standard n-dimensional Guassian vectors $\gamma_1, \ldots, \gamma_d \in \mathbb{R}^n$, and set $\Phi(v) := \frac{1}{\sqrt{d}} \left(\gamma_1^T \mathrm{v}, \ldots, \gamma_d^T \mathrm{v}\right)$.

Lemma 5 *(Dimension Reduction). Let $\mathrm{u}, \mathrm{v} \in \mathbb{R}^{\mathrm{n}}$ be unit vectors, let Φ be the random linear map as above, and let $t \geq 0$. Then, for a sufficiently large constant C,*

$$\mathsf{Prob}\left[|\mathrm{u}^{\mathrm{T}}\mathrm{v} - \Phi(\mathrm{u})^{\mathrm{T}}\Phi(\mathrm{v})| \geq \mathrm{t}\right] \leq \frac{C}{dt^2}$$

Proof It suffices to prove for u = v; the general case then follows using the equation $\mathrm{u}^{\mathrm{T}}\mathrm{v} = \frac{1}{2}\left(||\mathrm{u}||^2 + ||\mathrm{v}||^2 - ||\mathrm{u}-\mathrm{v}||^2\right)$. Let $X := \Phi(\mathrm{v})^{\mathrm{T}}\Phi(\mathrm{v}) = ||\Phi(\mathrm{v})||^2$. Then $X = \frac{1}{d}\sum_{i=1}^{d} Z_i^2$, with $Z_i := \gamma_i^T \mathrm{v}$. The Z_i are standard normal and independent, with $\mathsf{Var}\left[Z_i\right] = \mathbb{E}\left[Z_i^2\right] = 1$. So $\mathbb{E}\left[X\right] = 1$.

By using *Chebyshev inequality* : $\mathsf{Prob}\left[|X - \mathbb{E}\left[X\right]| \geq t\right] \leq \mathsf{Var}\left[X\right]/t^2$, where $\mathsf{Var} = \mathbb{E}\left[X^2\right] - \mathbb{E}\left[X\right]^2$ is the variance. By independence, $\mathsf{Var}\left[X\right] = \frac{1}{d^2}\sum_{i=1}^{d}\mathsf{Var}Z_i^2$. We want to see that each $\mathsf{Var}\left[Z_i^2\right]$ is bounded by a constant. A direct calculation (integration) gives $\mathsf{Var}\left[Z_i^2\right] = 2$. So $\mathsf{Var}\left[X\right] = \frac{2}{d}$ and the lemma follows. ■

Building the miniature

Fix $\delta > 0$ and choose $\hat{d}$ sufficiently large in terms of δ. the vectors $\hat{\mathrm{v}}_i$ of the miniature will live in $\mathbb{R}^{\hat{d}}$. Fix a δ-dense set $\hat{N}$ in $S^{\hat{d}-1}$ according to lemma 4 with size $\hat{n} := |\hat{N}|$, denoting the number of vertices in the miniature graph $\hat{G}$. Choose a random linear map $\Phi : \mathbb{R}^n \to \mathbb{R}^{\hat{d}}$, as in lemma 5 and set $\mathrm{v}_\mathrm{i}^* := \Phi(\mathrm{v}_\mathrm{i}), \mathrm{i} = 1, 2, \ldots, \mathrm{n}$.
Let us say that Φ *fails* on the vertex i if $||\mathrm{v}_\mathrm{i}^*|| \notin [1-\delta, 1+\delta]$. Similarly, Φ fails on an edge $\{i, j\}$ if it fails on i or j or if $|(\mathrm{v}_\mathrm{i}^*)^{\mathrm{T}}\mathrm{v}_\mathrm{j}^* - \mathrm{v}_\mathrm{i}^{\mathrm{T}}\mathrm{v}_\mathrm{j}| > \delta$. Let $F \subseteq E$ be the set of edges where Φ fails. By lemma 5 we have $\mathbb{E}\left[|F|\right] \leq \delta|E|$. So by Markov's inequality, $|F| \leq 2\delta|E|$ with probability at least $\frac{1}{2}$. We repeat the random choice of Φ until $|F| \leq 2\delta|E|$ holds; the expected number of repetitions is at most 2.

Let G^* be the graph obtained from G by discarding all vertices and edges where Φ fails. Next *discretize* the vectors v_i^*: For every $i \in V(G^*)$, we choose $\mathrm{v}_\mathrm{i}^{**} \in \hat{\mathrm{N}}$ at distance at most 2δ from v_i^*. (This is possible since each v_i^* is at most δ away from the sphere $S^{\hat{d}-1}$, and $\hat{N}$ is δ-dense in $S^{\hat{d}-1}$.) Finally, we $fold$ the graph G^* so that all vertices with the same vector $\mathrm{v}_\mathrm{i}^{**}$ are identified into the same vertex. The miniature graph $(\hat{G}, \hat{w})$ is the result of this folding.

Explicitly:

- The vertices of $\hat{G}$ are $1, 2, \ldots, \hat{n}$.
- Let $\hat{\mathrm{v}}_1, \ldots, \hat{\mathrm{v}}_{\hat{n}}$ be the points of $\hat{N}$ listed in some fixed order. These will form the feasibe SDP solution for the miniature.
- The edges of $\hat{G}$ are obtained from the edges of G^*; i.e., $\{\hat{\imath}, \hat{\jmath}\}$ is an edge if there is an edge $\{i, j\} \in E(G^*)$ such that $\mathrm{v}_\mathrm{i}^{**} = \hat{\mathrm{v}}_{\hat{\imath}}$ and $\mathrm{v}_\mathrm{j}^{**} = \hat{\mathrm{v}}_{\hat{\jmath}}$.
- The weight of such an edge $\{\hat{\imath}, \hat{\jmath}\}$ is the number of edges $\{i, j\}$ that got folded onto $\{\hat{\imath}, \hat{\jmath}\}$:
$$\hat{w}_{\hat{\imath}\hat{\jmath}} := \left|\left\{\{i, j\} \in E(G^*) : \mathrm{v}_\mathrm{i}^{**} = \hat{\mathrm{v}}_{\hat{\imath}}, \mathrm{v}_\mathrm{j}^{**} = \hat{\mathrm{v}}_{\hat{\jmath}}\right\}\right|$$

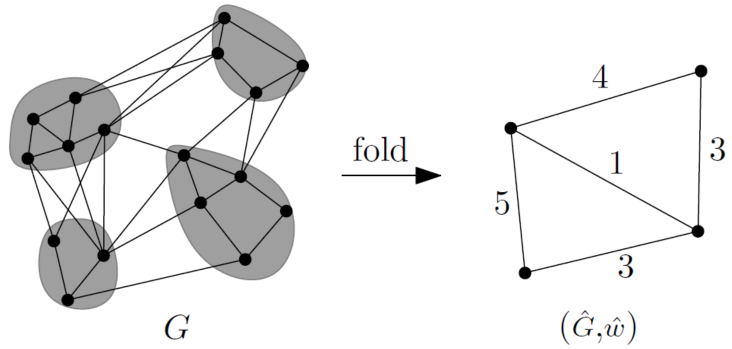

Figure 4: Folding Procedure [1]

Let us analyze the algorithm backwards. The folding loses nothing; the value equals $\sum_{\{i,j\}\in E(G^*)} \left(1 - (\mathrm{v}_\mathrm{i}^{**})^T \mathrm{v}_\mathrm{j}^{**}\right)/2$. (here we need to be slightly careful with vertices i, j that get folded to the same vertex $\hat{\imath}$ of $\hat{G}$, since there is no edge corresponding to $\{i, j\}$ in $\hat{G}$ - but for such i, j we have $(\mathrm{v}_\mathrm{i}^{**})^T \mathrm{v}_\mathrm{j}^{**} = 1$, and so they do not contribute to the SDP value). The discretization, replacing v_i^* by $\mathrm{v}_\mathrm{i}^{**}$, changes each of the relevant scalar products $(\mathrm{v}_\mathrm{i}^*)^T \mathrm{v}_\mathrm{j}^*$ by at most $O(\delta)$ (additive error), and thus the objective function changes by no more than $O(\delta \cdot |E|$.

Similarly, for each non-failed edge $\{i, j\}$, replacing the original v_i and v_j by the Φ-images $\mathrm{v}_\mathrm{i}^*, \mathrm{v}_\mathrm{j}^*$ changes the scalar product $\mathrm{v}_\mathrm{i}^T \mathrm{v}_\mathrm{j}$ by at most δ, and again, the total change in the objective function is $O(\delta \cdot |E|$. Finally, the number of failed edges is at most $2\delta|E|$, and discarding these may decrease the objective function by at most this much. Hence, $\mathsf{SDP}(\hat{G}, \hat{w}) \geq \mathsf{SDP}(G) - O(\delta) \cdot |E| \geq \mathsf{SDP}(G) - \varepsilon|E|$.

The miniature is finished and its size is bounded by a constant, and we can compute $\mathsf{Opt}(\hat{G}, \hat{w})$ by brute force. A cut in $(\hat{G}, \hat{w})$ of weight s can be *unfolded* to a cut in G^* with s edges, and hence to a cut with at least s edges in G. This finishes the rounding algorithm.

Theorem 6 *For every fixed $\varepsilon > 0$, and for every input graph G, the above rounding algorithm computes a cut of size at least*

$$\frac{1}{(1+\varepsilon)\mathit{Gap}} \cdot \mathit{Opt}(G),$$

in expected polynomial time.

Here is a summary of the rounding algorithm:

Algorithm 1: Rounding via miniatures for MaxCut

Input: G and an optimal SDP solution $v_1, \ldots, v_n$

1. (Dimension reduction) Map the vectors v_i into $\mathbb{R}^d$ using the random Gaussian map Φ. Repeat until only at most a small fraction of the edges fail.
2. Discard the failed vertices and edges.
3. Discretize the remaining vectors to vectors of a fixed δ-dense set.
4. Fold the graph, lumping together vertices with the same discretized vector.
5. Compute a maximum cut in the resulting constant-size graph.
6. Unfold back to a cut in the original graph

References

[1] Bernd Gartner and Jiri Matousek. *Approximation Algorithms and Semidefinite Programming*. Springer, 2012.