Convex Optimization and Applications

8 - Combinatorial Optimization

Guillaume Sagnol

Technische Universität Berlin

Outline

Lovász's Sandwich Theorem

For a simple graph $G = ([n], E)$, define

$$\begin{aligned}\vartheta(G) = \max_{X \in \mathbb{S}^n} \quad & \langle J, X \rangle \\ s.t. \quad & \langle I, X \rangle = 1 \\ & X_{ij} = 0, \quad \forall ij \in E \\ & X \succeq 0.\end{aligned}$$

Theorem (Lovász)

It holds

$$\alpha(G) \leq \vartheta(G) \leq \bar{\chi}(G),$$

where $\alpha(G)$ and $\bar{\xi}(G)$ are the stability and clique covering number of G, respectively.

Proof: cf. Handout

Corolary: A polytime algo to compute maximum stable sets and optimal colorings in perfect graphs.

Outline

Outline

Maximum cut

Definition

A cut of G is a partition of V in two node sets S and $\bar{S}$. The weight of a cut is the sum of the weights of the cut edges:

$$\operatorname{cut}(S,\bar{S}) = \sum_{\substack{ij\in E\\ i\in S, j\notin S}} w_{ij}$$

MAXCUT: find the cut maximizing $\operatorname{cut}(S,\bar{S})$.

- Many Applications: computer vision, chip design
- We can assume w.l.o.g. that G is the complete graph (set $w_{ij}=0$ whenever $ij \notin E$)

A binary QP formulation

Let $\boldsymbol{x} \in \{-1, 1\}^n$ be an indicator vector for the cut, i.e. $x_i = 1$ if $i \in S$, and $x_i = -1$ if $i \in \bar{S}$.

A binary QP formulation

Let $\boldsymbol{x} \in \{-1, 1\}^n$ be an indicator vector for the cut, i.e. $x_i = 1$ if $i \in S$, and $x_i = -1$ if $i \in \bar{S}$.

Then,

$$1 - x_i x_j = \begin{cases} 2 & \text{if } \{i, j\} \text{ is a cut-edge} \\ 0 & \text{otherwise.} \end{cases}$$

A binary QP formulation

Let $\boldsymbol{x} \in \{-1, 1\}^n$ be an indicator vector for the cut, i.e. $x_i = 1$ if $i \in S$, and $x_i = -1$ if $i \in \bar{S}$.

Then,

$$1 - x_i x_j = \begin{cases} 2 & \text{if } \{i, j\} \text{ is a cut-edge} \\ 0 & \text{otherwise.} \end{cases}$$

Hence,

$$\text{cut}(S, \bar{S}) = \frac{1}{2} \sum_{ij \in E} w_{ij}(1 - x_i x_j) = \frac{1}{4} \sum_{1 \leq i,j \leq n} w_{ij}(1 - x_i x_j).$$

A binary QP formulation

Let $\boldsymbol{x} \in \{-1, 1\}^n$ be an indicator vector for the cut, i.e. $x_i = 1$ if $i \in S$, and $x_i = -1$ if $i \in \bar{S}$.

Then,

$$1 - x_i x_j = \begin{cases} 2 & \text{if } \{i, j\} \text{ is a cut-edge} \\ 0 & \text{otherwise.} \end{cases}$$

Hence,

$$\text{cut}(S, \bar{S}) = \frac{1}{2} \sum_{ij \in E} w_{ij}(1 - x_i x_j) = \frac{1}{4} \sum_{1 \leq i,j \leq n} w_{ij}(1 - x_i x_j).$$

So the maximum cut problem can be formulated as

$$\begin{aligned} \underset{\boldsymbol{x}}{\textbf{maximize}} \quad & \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) \\ & \boldsymbol{x} \in \{-1, 1\}^n. \end{aligned}$$

A rank-constrained SDP

Lemma

The matrix $X \in \mathbb{S}^n$ satisfies $X_{ij} = x_i x_j$ for some vector $\boldsymbol{x} \in \{-1, 1\}^n$ if and only if

$$X \succeq 0, \quad \mathbf{diag}\,(X) = \mathbf{1}, \quad \text{and} \quad \mathbf{rank}\,(X) = 1.$$

A rank-constrained SDP

Lemma

The matrix $X \in \mathbb{S}^n$ satisfies $X_{ij} = x_i x_j$ for some vector $\boldsymbol{x} \in \{-1, 1\}^n$ if and only if

$$X \succeq 0, \quad \mathbf{diag}\,(X) = \mathbf{1}, \quad \text{and} \quad \mathbf{rank}\,(X) = 1.$$

If X satisfies the property of the lemma, then

$$\text{cut}(S, \bar{S}) = \frac{1}{4} \sum_{i,j} w_{ij}(1 - x_i x_j) = \frac{1}{4} \langle W, J - X \rangle.$$

If MAXCUT is equivalent to:

$$\begin{aligned} \underset{X \in \mathbb{S}^n}{\textbf{maximize}} \quad & \frac{1}{4} \langle W, J - X \rangle \\ & \mathbf{diag}\,(X) = \mathbf{1} \\ & X \succeq 0, \quad \mathbf{rank}\,(X) = 1. \end{aligned}$$

MAXCUT SDP

$$\begin{aligned}
\underset{X\in\mathbb{S}^n}{\mathbf{maximize}} \quad & \frac{1}{4}\langle W, J-X\rangle \\
& \mathbf{diag}\,(X) = \mathbf{1} \\
& X \succeq 0, \quad \mathbf{rank}\,(X) = 1.
\end{aligned}$$

MAXCUT SDP

$$\begin{aligned} \underset{X \in \mathbb{S}^n}{\textbf{maximize}} \quad & \frac{1}{4}\langle W, J - X\rangle \\ & \textbf{diag}\,(X) = \mathbf{1} \\ & X \succeq 0, \quad \cancel{\textbf{rank}\,(X) = 1}. \end{aligned}$$

- We obtain an SDP relaxation by removing the rank constraint:

$$\text{maxcut}(G) \leq \text{SDP}\,.$$

MAXCUT SDP

$$\begin{aligned} \underset{X \in \mathbb{S}^n}{\textbf{maximize}} \quad & \frac{1}{4}\langle W, J - X \rangle \\ & \textbf{diag}\,(X) = \mathbf{1} \\ & X \succeq 0, \quad \cancel{\textbf{rank}\,(X) = 1}. \end{aligned}$$

- We obtain an SDP relaxation by removing the rank constraint:

$$\text{maxcut}(G) \leq \text{SDP}\,.$$

- Alternative interpretation of the relaxation: $x_i x_j = \langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle$, where $\boldsymbol{h}_k = x_k \boldsymbol{e}_1 = \pm \boldsymbol{e}_1, \ \forall k$. We obtain the SDP by allowing the vectors $\boldsymbol{h}$ to vary over the whole sphere $\{\boldsymbol{h} : \|\boldsymbol{h}\| = 1\}$: we define $X_{ij} = \langle \boldsymbol{h}_i, \boldsymbol{h}_j \rangle$, so $X \succeq 0$ and $X_{ii} = \|\boldsymbol{h}_i\|^2 = 1$.

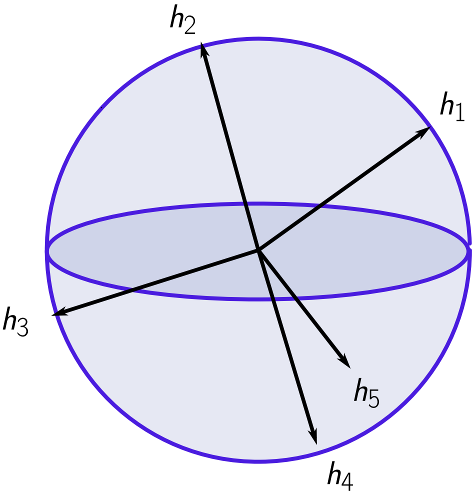
Goemans & Williamson's rounding

1. Compute a solution X^* of the SDP.
2. Compute a decomposition $X^* = H^T H$ (for example, a Cholesky decomposition). Let $H = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n]$, so $X^*_{ii} = 1 \implies \boldsymbol{h}_i^T \boldsymbol{h}_i = \|\boldsymbol{h}_i\|^2 = 1$.
3. Draw a vector $\boldsymbol{r}$ uniformly at random over the unit sphere of $\mathbb{R}^n$. To do this, one can draw independently $z_i \sim \mathcal{N}(0,1)$ for $i = 1, \ldots, n$, and then take $\boldsymbol{r} = \frac{1}{\|\boldsymbol{z}\|}\boldsymbol{z}$.
4. Finally, return the cut defined by

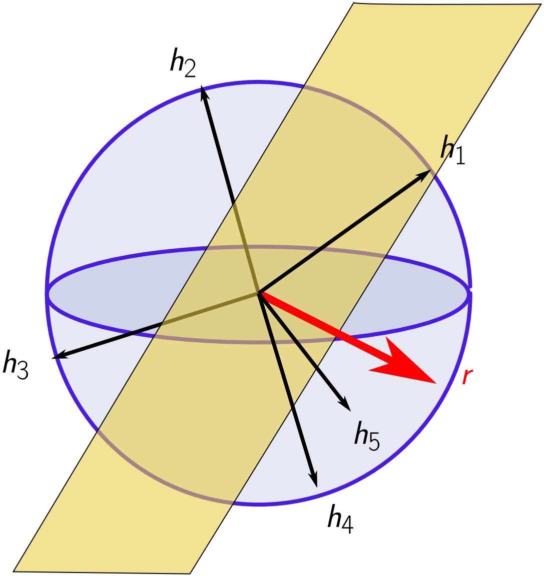
$$S = \{i \in V : \ \boldsymbol{r}^T \boldsymbol{h}_i > 0\}.$$

(Or, equivalently, define the cut through the vector $x_i = \text{sign}(\boldsymbol{r}^T \boldsymbol{h}_i), \forall i \in [n]$.)

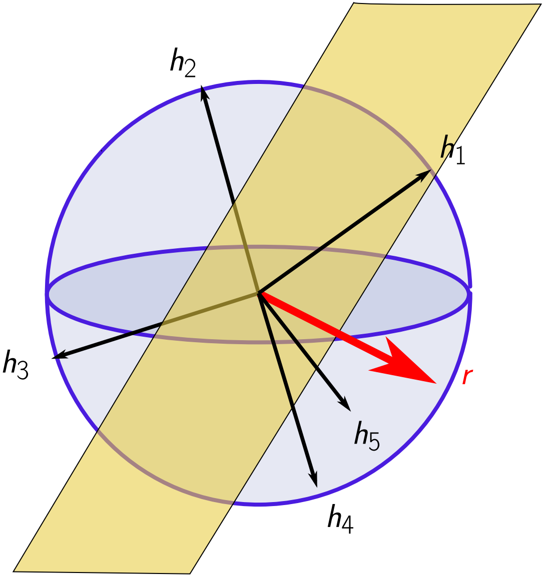
Goemans & Williamson's rounding



Goemans & Williamson's rounding



Goemans & Williamson's rounding



The random hyperplane with normal vector $\boldsymbol{r}$ produces the cut

$$S = \{2, 3\},$$

$$\bar{S} = \{1, 4, 5\}.$$

Analysis of GW randomized algorithm

Theorem (Goemans & Williamson).

Let $(S, \bar{S})$ be the (random) cut returned by the random projection algorithm. Then,

$$\mathbb{E}[\mathrm{cut}(S, \bar{S})] \geq \alpha \ \mathrm{SDP} \geq \alpha \ \mathrm{maxcut}(G),$$

where $\alpha \simeq 0.87856$.

A geometric lemma

Lemma

Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ with $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1$, and let $\boldsymbol{r}$ be a random vector drawn uniformly at random on the sphere.

Denote by H be the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^T \boldsymbol{r} = 0\}$.

Then, the probability that H separates $\boldsymbol{u}$ and $\boldsymbol{v}$ is $\dfrac{\theta}{\pi}$, where $\theta = \arccos(\boldsymbol{u}^T \boldsymbol{v})$ is the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$.

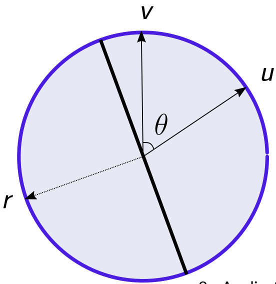
A geometric lemma

Lemma

Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ with $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1$, and let $\boldsymbol{r}$ be a random vector drawn uniformly at random on the sphere.

Denote by H be the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^T \boldsymbol{r} = 0\}$.

Then, the probability that H separates $\boldsymbol{u}$ and $\boldsymbol{v}$ is $\dfrac{\theta}{\pi}$, where $\theta = \arccos(\boldsymbol{u}^T \boldsymbol{v})$ is the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$.



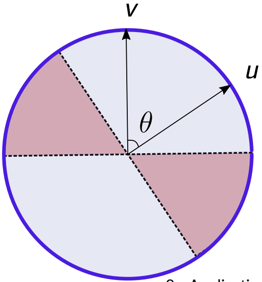
A geometric lemma

Lemma

Let $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ with $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1$, and let $\boldsymbol{r}$ be a random vector drawn uniformly at random on the sphere.

Denote by H be the hyperplane $\{\boldsymbol{x} \in \mathbb{R}^n : \boldsymbol{x}^T \boldsymbol{r} = 0\}$.

Then, the probability that H separates $\boldsymbol{u}$ and $\boldsymbol{v}$ is $\dfrac{\theta}{\pi}$, where $\theta = \arccos(\boldsymbol{u}^T \boldsymbol{v})$ is the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$.



Proof of GW theorem

The expected weight of the cut $(S, \bar{S})$ is

$$\begin{aligned}\mathbb{E}[\text{cut}(S, \bar{S})] &= \sum_{ij \in E} w_{ij} \mathbb{P}[\{i, j\} \text{ belongs to the cut set}] \\ &= \sum_{ij \in E} w_{ij} \frac{\arccos(\boldsymbol{h_i}^T \boldsymbol{h}_j)}{\pi}\end{aligned}$$

Proof of GW theorem

The expected weight of the cut $(S, \bar{S})$ is

$$\mathbb{E}[\text{cut}(S, \bar{S})] = \sum_{ij \in E} w_{ij} \mathbb{P}[\{i, j\} \text{ belongs to the cut set}]$$
$$= \sum_{ij \in E} w_{ij} \frac{\arccos(\boldsymbol{h_i}^T \boldsymbol{h}_j)}{\pi}$$

Multiply/divide each term by $\frac{1}{2}(1 - \boldsymbol{h_i}^T \boldsymbol{h}_j) = \frac{1}{2}(1 - X_{ij}^*)$:

$$\mathbb{E}[\text{cut}(S, \bar{S})] = \sum_{ij \in E} \frac{1}{2} w_{ij} (1 - X_{ij}^*) \frac{2 \arccos(\boldsymbol{h_i}^T \boldsymbol{h}_j)}{\pi(1 - \boldsymbol{h_i}^T \boldsymbol{h}_j)}$$

Proof of GW theorem

The expected weight of the cut $(S, \bar{S})$ is

$$\mathbb{E}[\text{cut}(S, \bar{S})] = \sum_{ij \in E} w_{ij} \mathbb{P}[\{i, j\} \text{ belongs to the cut set}]$$

$$= \sum_{ij \in E} w_{ij} \frac{\arccos(\boldsymbol{h_i}^T \boldsymbol{h}_j)}{\pi}$$

Multiply/divide each term by $\frac{1}{2}(1 - \boldsymbol{h_i}^T \boldsymbol{h}_j) = \frac{1}{2}(1 - X_{ij}^*)$:

$$\mathbb{E}[\text{cut}(S, \bar{S})] = \sum_{ij \in E} \frac{1}{2} w_{ij} (1 - X_{ij}^*) \qquad \underbrace{\frac{2 \arccos(\boldsymbol{h_i}^T \boldsymbol{h}_j)}{\pi(1 - \boldsymbol{h_i}^T \boldsymbol{h}_j)}}_{\geq \alpha := \inf_{\theta \in [0, \pi]} \frac{2}{\pi} \frac{\theta}{1 - \cos(\theta)} \simeq 0.87856}$$

Proof of GW theorem

The expected weight of the cut $(S, \bar{S})$ is

$$\mathbb{E}[\mathrm{cut}(S,\bar{S})] = \sum_{ij \in E} w_{ij} \mathbb{P}[\{i,j\} \text{ belongs to the cut set}]$$

$$= \sum_{ij \in E} w_{ij} \frac{\arccos(\boldsymbol{h_i}^T \boldsymbol{h}_j)}{\pi}$$

Multiply/divide each term by $\frac{1}{2}(1 - \boldsymbol{h_i}^T \boldsymbol{h}_j) = \frac{1}{2}(1 - X_{ij}^*)$:

$$\mathbb{E}[\mathrm{cut}(S,\bar{S})] = \sum_{ij \in E} \frac{1}{2} w_{ij} (1 - X_{ij}^*) \qquad \underbrace{\frac{2 \arccos(\boldsymbol{h_i}^T \boldsymbol{h}_j)}{\pi(1 - \boldsymbol{h_i}^T \boldsymbol{h}_j)}}_{\geq \alpha := \inf_{\theta \in [0,\pi]} \frac{2}{\pi} \frac{\theta}{1-\cos(\theta)} \simeq 0.87856}$$

Hence,

$$\mathbb{E}[\mathrm{cut}(S,\bar{S})] \geq \alpha \sum_{ij \in E} \frac{1}{2} w_{ij} (1 - X_{ij}^*) = \alpha \, \frac{1}{4} \langle W, J - X^* \rangle = \alpha \ \mathrm{SDP} \,.$$

Outline

1. Stable Set Problem
2. Maxcut
   - Goemans & Williamson randomized rounding
   - Nesterov $\frac{2}{\pi}-$ rounding
   - MAX-BISECTION
   - MAX-3-CUT via complex SDP
3. Nonconvex QCQPs
   - SDP relaxations for nonconvex QCQPs
   - Completely positive programming
   - Burer's result for binary QPs

Laplacian formulation of MAXCUT

Using $x_i^2 = 1,\ \forall i$, we can rewrite the objective function

$$\begin{aligned}\frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j) &= \frac{1}{4}\sum_{i<j} w_{ij}(x_i^2 + x_j^2 - 2x_i x_j)\\ &= \frac{1}{4}\sum_{i<j} w_{ij}\left((\boldsymbol{e}_i - \boldsymbol{e}_j)^T \boldsymbol{x}\right)^2\\ &= \frac{1}{4}\sum_{i<j} w_{ij}\boldsymbol{x}^T(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T\boldsymbol{x} = \frac{1}{4}\boldsymbol{x}^T L \boldsymbol{x},\end{aligned}$$

$L := \sum_{i<j} w_{ij}(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T \succeq 0$ is the Laplacian matrix of G.

Laplacian formulation of MAXCUT

Using $x_i^2 = 1,\ \forall i$, we can rewrite the objective function

$$\begin{aligned}
\frac{1}{2}\sum_{i<j} w_{ij}(1 - x_i x_j) &= \frac{1}{4}\sum_{i<j} w_{ij}(x_i^2 + x_j^2 - 2x_i x_j) \\
&= \frac{1}{4}\sum_{i<j} w_{ij}\left((\boldsymbol{e}_i - \boldsymbol{e}_j)^T \boldsymbol{x}\right)^2 \\
&= \frac{1}{4}\sum_{i<j} w_{ij}\boldsymbol{x}^T(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T \boldsymbol{x} = \frac{1}{4}\boldsymbol{x}^T L \boldsymbol{x},
\end{aligned}$$

$L := \sum_{i<j} w_{ij}(\boldsymbol{e}_i - \boldsymbol{e}_j)(\boldsymbol{e}_i - \boldsymbol{e}_j)^T \succeq 0$ is the Laplacian matrix of G.

Can GW-rounding algorithm be used for any binary QP

$$\underset{\boldsymbol{x}\in\{-1,1\}^n}{\textbf{maximize}} \quad \boldsymbol{x}^T Q \boldsymbol{x}, \qquad \text{where } Q \succeq 0?$$

GW rounding for binary QPs

Can GW-rounding algorithm be used for any binary QP

$$\underset{x \in \{-1,1\}^n}{\textbf{maximize}} \quad x^T Q x, \qquad \text{where } Q \succeq 0?$$

GW rounding for binary QPs

Can GW-rounding algorithm be used for any binary QP

$$\underset{x \in \{-1,1\}^n}{\textbf{maximize}} \quad x^T Q x, \qquad \text{where } Q \succeq 0?$$

Consider the SDP $\quad \underset{X \in \mathbb{S}^n}{\textbf{maximize}} \{\langle X, Q \rangle : \textbf{diag}\,(X) = \mathbf{1},\ X \succeq 0\}.$

GW rounding for binary QPs

Can GW-rounding algorithm be used for any binary QP

$$\underset{\boldsymbol{x}\in\{-1,1\}^n}{\textbf{maximize}} \quad \boldsymbol{x}^T Q \boldsymbol{x}, \qquad \text{where } Q \succeq 0?$$

Consider the SDP $\underset{X\in\mathbb{S}^n}{\textbf{maximize}}\{\langle X, Q\rangle : \textbf{diag}\,(X) = \mathbf{1},\ X \succeq 0\}.$

As before, define $x_i = \operatorname{sign}(\boldsymbol{r}^T \boldsymbol{h}_i)$, where $\boldsymbol{h}_i^T \boldsymbol{h}_j = X^*_{ij}$, $r \sim \mathcal{N}(0, I)$.

GW rounding for binary QPs

Can GW-rounding algorithm be used for any binary QP

$$\underset{\boldsymbol{x}\in\{-1,1\}^n}{\textbf{maximize}} \quad \boldsymbol{x}^T Q \boldsymbol{x}, \qquad \text{where } Q \succeq 0?$$

Consider the SDP $\underset{X\in\mathbb{S}^n}{\textbf{maximize}}\{\langle X, Q\rangle : \textbf{diag}\,(X) = \mathbf{1},\ X \succeq 0\}.$

As before, define $x_i = \text{sign}(\boldsymbol{r}^T \boldsymbol{h}_i)$, where $\boldsymbol{h}_i^T \boldsymbol{h}_j = X^*_{ij}$, $r \sim \mathcal{N}(0, I)$.

$$\mathbb{E}[x_i x_j] = 1 \cdot \underbrace{\mathbb{P}[x_i = x_j]}_{1-\arccos(X_{ij})/\pi} + (-1) \cdot \underbrace{\mathbb{P}[x_i \neq x_j]}_{\arccos(X_{ij})/\pi} = 1 - \frac{2}{\pi}\arccos(X_{ij})$$

$$= \frac{2}{\pi}\arcsin(X_{ij}).$$

GW rounding for binary QPs

Can GW-rounding algorithm be used for any binary QP

$$\underset{\boldsymbol{x}\in\{-1,1\}^n}{\textbf{maximize}} \quad \boldsymbol{x}^T Q \boldsymbol{x}, \qquad \text{where } Q \succeq 0?$$

Consider the SDP $\underset{X\in\mathbb{S}^n}{\textbf{maximize}}\{\langle X, Q\rangle : \textbf{diag}\,(X) = \mathbf{1},\ X \succeq 0\}.$

As before, define $x_i = \operatorname{sign}(\boldsymbol{r}^T \boldsymbol{h}_i)$, where $\boldsymbol{h}_i^T \boldsymbol{h}_j = X_{ij}^*$, $r \sim \mathcal{N}(0, I)$.

$$\mathbb{E}[x_i x_j] = 1 \cdot \underbrace{\mathbb{P}[x_i = x_j]}_{1-\arccos(X_{ij})/\pi} + (-1) \cdot \underbrace{\mathbb{P}[x_i \neq x_j]}_{\arccos(X_{ij})/\pi} = 1 - \frac{2}{\pi} \arccos(X_{ij})$$

$$= \frac{2}{\pi} \arcsin(X_{ij}).$$

Then,

$$\mathbb{E}[\boldsymbol{x}^T Q \boldsymbol{x}] = \langle Q, \mathbb{E}[\boldsymbol{x}\boldsymbol{x}^T]\rangle = \frac{2}{\pi} \langle Q, \arcsin(X)\rangle,$$

Nesterov analysis of GW algo for binary QPs

Proposition (Schur Product Theorem).

Let $X \succeq 0$ and $Y \succeq 0$. Then, $X \circ Y \succeq 0$, where $X \circ Y \in \mathbb{S}^n$ denotes the Hadamard (elementwise) product of X and Y, i.e., $(X \circ Y)_{ij} = X_{ij} Y_{ij}$.

Nesterov analysis of GW algo for binary QPs

Proposition (Schur Product Theorem).

Let $X \succeq 0$ and $Y \succeq 0$. Then, $X \circ Y \succeq 0$, where $X \circ Y \in \mathbb{S}^n$ denotes the Hadamard (elementwise) product of X and Y, i.e., $(X \circ Y)_{ij} = X_{ij} Y_{ij}$.

Theorem (Nesterov.)

$$\mathbb{E}[\boldsymbol{x}^T Q \boldsymbol{x}] \geq \frac{2}{\pi} SDP \geq \frac{2}{\pi} OPT.$$

$$\arcsin(x) = x + \frac{1}{6}x^3 + \frac{3}{40}x^5 \ldots \implies \arcsin(X) = X + \frac{1}{6}X^{\circ 3} + \frac{3}{40}X^{\circ 5} + \ldots \succeq X$$

Then,

$$\mathbb{E}[\boldsymbol{x}^T Q \boldsymbol{x}] \frac{2}{\pi} \langle Q, \arcsin(X) \rangle \geq = \frac{2}{\pi} \langle Q, X \rangle = \frac{2}{\pi} SDP.$$

Outline

MAX-BISECTION

In MAX-BISECTION, the input is a weighted graph G with an even number of nodes, and we search a cut $(S, \bar{S})$ of maximal weight such that $|S| = |\bar{S}| = \frac{n}{2}$.

- We present an extension of the GW rounding algorithm for MAX-BISECTION with a 0.651 performance guarantee by Frieze & Jerrum.
- The result has been improved, now the best polytime approximation is very close to $\alpha = 0.878$, though highly impracticable.

Thightened SDP

For MAX BISECTION, we have the constraint $\mathbf{1}^T \boldsymbol{x} = 0$, which implies $\mathbf{1}^T \boldsymbol{x}\boldsymbol{x}^T \mathbf{1} = 0$, so the equality $\mathbf{1}^T X \mathbf{1} = \langle J, X \rangle = 0$ is a valid equality constraint.

Thightened SDP

For MAX BISECTION, we have the constraint $\mathbf{1}^T \boldsymbol{x} = 0$, which implies $\mathbf{1}^T \boldsymbol{x}\boldsymbol{x}^T \mathbf{1} = 0$, so the equality $\mathbf{1}^T X \mathbf{1} = \langle J, X \rangle = 0$ is a valid equality constraint.

This SDP defines an upper bound for MAX-BISECTION:

$$
\begin{aligned}
\underset{X \in \mathbb{S}^n}{\textbf{maximize}} \quad & \frac{1}{4}\langle W, J - X \rangle \\
& \textbf{diag}\,(X) = \mathbf{1} \\
& \langle J, X \rangle = 0 \\
& X \succeq 0
\end{aligned}
$$

FJ rounding algorithm

For $\boldsymbol{p} \in \mathbb{R}^n$, define the bisection $(S(\boldsymbol{p}), \overline{S(\boldsymbol{p})})$ as follows:

- $S_1 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} \geq 0\}, S_2 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} < 0\}$
  w.l.o.g. $|S_1| \geq |S_2|$, so $\ell := |S_1| \geq \dfrac{n}{2}$.
- Sort $S_1 = \{i_1, \ldots, i_\ell\}$ so that $\zeta_{i_1} \geq \zeta_{i_2} \geq \ldots \geq \zeta_{i_\ell}$, where $\zeta_i = \sum_{j \in S_2} w_{ij}$
- $S(\boldsymbol{p}) := \{i_1, \ldots, i_{\frac{n}{2}}\}$

FJ rounding algorithm

For $\boldsymbol{p} \in \mathbb{R}^n$, define the bisection $(S(\boldsymbol{p}), \overline{S(\boldsymbol{p})})$ as follows:

- $S_1 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} \geq 0\}, S_2 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} < 0\}$
  w.l.o.g. $|S_1| \geq |S_2|$, so $\ell := |S_1| \geq \frac{n}{2}$.
- Sort $S_1 = \{i_1, \ldots, i_\ell\}$ so that $\zeta_{i_1} \geq \zeta_{i_2} \geq \ldots \geq \zeta_{i_\ell}$, where $\zeta_i = \sum_{j \in S_2} w_{ij}$
- $S(\boldsymbol{p}) := \{i_1, \ldots, i_{\frac{n}{2}}\}$

Theorem (Frieze & Jerrum).

Generate independent $\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K \sim \mathcal{N}(0, I)$, and output the best bisection cut S from $S(\boldsymbol{p}_1), \ldots, S(\boldsymbol{p}_K)$. If $K \geq 2\epsilon^{-1} \log(2\epsilon^{-1})$ for some $\epsilon > 0$, then

$$\mathbb{E}[\text{cut}(S, \bar{S})] \geq \Big(\underbrace{2(\sqrt{2\alpha} - 1)}_{>0.6511} - \epsilon \Big) \text{ MAX-BISECTION}.$$

Analysis of FJ approximation algorithm (1/4)

$\boldsymbol{p} \sim \mathcal{N}(0, I)$, $S_1 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} \geq 0\}, S_2 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} < 0\}$

Let $C = \text{cut}(S_1, S_2)$. Following GW analysis,

$$\mathbb{E}[C] \geq \alpha\, SDP \geq \alpha \text{ MAX-BISECTION}.$$

Analysis of FJ approximation algorithm (1/4)

$\boldsymbol{p} \sim \mathcal{N}(0, I)$, $S_1 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} \geq 0\}, S_2 := \{i : \boldsymbol{h}_i^T \boldsymbol{p} < 0\}$

Let $C = \text{cut}(S_1, S_2)$. Following GW analysis,

$$\mathbb{E}[C] \geq \alpha\, SDP \geq \alpha \text{ MAX-BISECTION}.$$

Let $Y := |S_1| \cdot |S_2| \leq n^2/4$. We have:

$$\begin{aligned}
\mathbb{E}[Y] &= \sum_{i<j} \mathbb{P}[(i,j) \text{ are separated by cut } (S_1, S_2)] \\
&= \sum_{i<j} \frac{\arccos(X_{ij})}{\pi} \\
&\geq \sum_{i<j} \frac{\alpha}{2}(1 - X_{ij}) \qquad \text{(by definition of } \alpha) \\
&= \frac{\alpha}{2} \left(\sum_{i<j} 1 - \frac{1}{2}(\underbrace{\langle J, X \rangle}_{=0} - \underbrace{\textbf{trace}\, X}_{=n}) \right) \\
&= \frac{\alpha}{2} \left(\frac{n(n-1)}{2} + \frac{n}{2} \right) = \alpha \frac{n^2}{4}.
\end{aligned}$$

Analysis of FJ approximation algorithm (2/4)

We have shown: $\mathbb{E}[C] \geq \alpha \text{ OPT}$ and $\mathbb{E}[Y] \geq \alpha \frac{n^2}{4}$.

Next we define $Z = \frac{C}{\text{OPT}} + \frac{Y}{n^2/4}$

- It can be seen that Z is bounded $(0 \leq Z \leq 3)$

Analysis of FJ approximation algorithm (2/4)

We have shown: $\mathbb{E}[C] \geq \alpha$ OPT and $\mathbb{E}[Y] \geq \alpha \frac{n^2}{4}$.

Next we define $Z = \frac{C}{\text{OPT}} + \frac{Y}{n^2/4}$

- It can be seen that Z is bounded ($0 \leq Z \leq 3$)
- $\mathbb{E}[Z] \geq 2\alpha$

Analysis of FJ approximation algorithm (2/4)

We have shown: $\mathbb{E}[C] \geq \alpha$ OPT and $\mathbb{E}[Y] \geq \alpha \frac{n^2}{4}$.

Next we define $Z = \frac{C}{\text{OPT}} + \frac{Y}{n^2/4}$

- It can be seen that Z is bounded ($0 \leq Z \leq 3$)
- $\mathbb{E}[Z] \geq 2\alpha$
- So if we take enough random samples $z_1, \ldots, z_K$ of Z, we will (with large probability) get a $z_i \geq 2\alpha(1-\epsilon)$

Analysis of FJ approximation algorithm (2/4)

We have shown: $\mathbb{E}[C] \geq \alpha$ OPT and $\mathbb{E}[Y] \geq \alpha \frac{n^2}{4}$.

Next we define $Z = \frac{C}{\text{OPT}} + \frac{Y}{n^2/4}$

- It can be seen that Z is bounded ($0 \leq Z \leq 3$)
- $\mathbb{E}[Z] \geq 2\alpha$
- So if we take enough random samples $z_1, \ldots, z_K$ of Z, we will (with large probability) get a $z_i \geq 2\alpha(1-\epsilon)$
- For the sake of simplicity, we assume that we obtained a cut S_1, S_2 such that

$$\frac{\text{cut}(S_1, S_2)}{OPT} + \frac{|S_1| \cdot |S_2|}{n^2/4} \geq \mathbb{E}[Z] = 2\alpha.$$

Analysis of FJ approximation algorithm (3/4)

$$\frac{\text{cut}(S_1, S_2)}{OPT} + \frac{|S_1| \cdot |S_2|}{n^2/4} \geq \mathbb{E}[Z] = 2\alpha.$$

Analysis of FJ approximation algorithm (3/4)

$$\frac{\text{cut}(S_1, S_2)}{OPT} + \frac{|S_1| \cdot |S_2|}{n^2/4} \geq \mathbb{E}[Z] = 2\alpha.$$

Assume $\text{cut}(S_1, S_2) = \lambda \cdot OPT$ and $|S_1| = \delta n$, so this rewrites

$$\lambda + \frac{\delta n(n - \delta n)}{n^2/4} \geq 2\alpha \iff \lambda \geq 2\alpha - 4\delta(1 - \delta).$$

Analysis of FJ approximation algorithm (3/4)

$$\frac{\text{cut}(S_1, S_2)}{OPT} + \frac{|S_1| \cdot |S_2|}{n^2/4} \geq \mathbb{E}[Z] = 2\alpha.$$

Assume $\text{cut}(S_1, S_2) = \lambda \cdot OPT$ and $|S_1| = \delta n$, so this rewrites

$$\lambda + \frac{\delta n(n - \delta n)}{n^2/4} \geq 2\alpha \iff \lambda \geq 2\alpha - 4\delta(1 - \delta).$$

Now, consider the bisection cut $S = \{i_1, \ldots, i_{n/2}\}$ returned by the greedy swapping procedure applied to S_1 and S_2:

$$\text{cut}(S, \bar{S}) = \sum_{\substack{i \in S \\ j \in \bar{S}}} w_{ij} \geq \sum_{\substack{i \in S \\ j \in S_2}} w_{ij} = \sum_{i \in S} \zeta_i \geq \frac{n/2}{|S_1|} \sum_{i \in S_1} \zeta_i = \frac{n/2}{|S_1|} \text{cut}(S_1, S_2)$$

Analysis of FJ approximation algorithm (3/4)

$$\frac{\text{cut}(S_1, S_2)}{OPT} + \frac{|S_1| \cdot |S_2|}{n^2/4} \geq \mathbb{E}[Z] = 2\alpha.$$

Assume $\text{cut}(S_1, S_2) = \lambda \cdot OPT$ and $|S_1| = \delta n$, so this rewrites

$$\lambda + \frac{\delta n(n - \delta n)}{n^2/4} \geq 2\alpha \iff \lambda \geq 2\alpha - 4\delta(1 - \delta).$$

Now, consider the bisection cut $S = \{i_1, \ldots, i_{n/2}\}$ returned by the greedy swapping procedure applied to S_1 and S_2:

$$\text{cut}(S, \bar{S}) = \sum_{\substack{i \in S \\ j \in \bar{S}}} w_{ij} \geq \sum_{\substack{i \in S \\ j \in S_2}} w_{ij} = \sum_{i \in S} \zeta_i \geq \frac{n/2}{|S_1|} \sum_{i \in S_1} \zeta_i = \frac{n/2}{|S_1|} \text{cut}(S_1, S_2)$$

Putting all together,

$$\text{cut}(S, \bar{S}) \geq \frac{n/2}{\delta n} \lambda \cdot OPT \geq \frac{1}{2\delta} \big(2\alpha - 4\delta(1 - \delta)\big) \cdot OPT.$$

Analysis of FJ approximation algorithm (4/4)

If $\text{cut}(S_1, S_2) = \lambda \cdot OPT$ and $|S_1| = \delta n$, we have shown

$$\text{cut}(S, \bar{S}) \geq \frac{1}{2\delta}\big(2\alpha - 4\delta(1-\delta)\big) \cdot OPT.$$

Analysis of FJ approximation algorithm (4/4)

If $\text{cut}(S_1, S_2) = \lambda \cdot OPT$ and $|S_1| = \delta n$, we have shown

$$\text{cut}(S, \bar{S}) \geq \frac{1}{2\delta}\big(2\alpha - 4\delta(1-\delta)\big) \cdot OPT.$$

The above bound depends on δ, but simple calculus shows that it is minimized for $\delta = \sqrt{\frac{\alpha}{2}} \in [0, 1]$. After substitution,

$$\text{cut}(S, \bar{S}) \geq 2(\sqrt{2\alpha} - 1)OPT.$$

Analysis of FJ approximation algorithm (4/4)

If $\text{cut}(S_1, S_2) = \lambda \cdot OPT$ and $|S_1| = \delta n$, we have shown

$$\text{cut}(S, \bar{S}) \geq \frac{1}{2\delta}\big(2\alpha - 4\delta(1-\delta)\big) \cdot OPT.$$

The above bound depends on δ, but simple calculus shows that it is minimized for $\delta = \sqrt{\dfrac{\alpha}{2}} \in [0, 1]$. After substitution,

$$\text{cut}(S, \bar{S}) \geq 2(\sqrt{2\alpha} - 1) OPT.$$

The analysis is valid (up to some ϵ) for at least one of the K cuts we sampled (a cut associated with a value $Z \geq 2\alpha$). So in particular, the bound also holds for the best of the K cuts $S(\boldsymbol{p}_1), \ldots, S(\boldsymbol{p}_K)$.

Outline

MAX-3-CUT

MAX-3-CUT: partition the vertices of G in 3 subsets S_0, S_1, S_2, so that the weight of all edges with endpoints in two different subsets is maximized.

- An elegant extension of the MAX-CUT result (also due to Goemans and Williamson) to the case of MAX-3-CUT, relying on complex semidefinite programming
- That is, conic programming over the cone of $n \times n$ Hermitian matrices:

$$\begin{aligned}\mathcal{H}_+^n &= \{X \in \mathbb{C}^{n\times n} : X = X^*, \quad z^* X z \geq 0, \forall z \in \mathbb{C}^n\} \\ &= \{HH^* : H \in \mathbb{C}^{n\times n}\}\end{aligned}$$

Recall that for $X \in \mathbb{C}^{m\times n}$, the complex conjugate of X is

$$X^* = \overline{X^T} \in \mathbb{C}^{n\times m}.$$

Representing a cut with third roots of unity

- $\omega = e^{2\boldsymbol{i}\frac{\pi}{3}} = -\frac{1}{2} + \boldsymbol{i}\frac{\sqrt{3}}{2} \in \mathbb{C}$

Representing a cut with third roots of unity

- $\omega = e^{2\boldsymbol{i}\frac{\pi}{3}} = -\frac{1}{2} + \boldsymbol{i}\frac{\sqrt{3}}{2} \in \mathbb{C}$
- Consider a 3-Cut (S_1, S_2, S_3) defined by $\boldsymbol{x} \in \{0, 1, 2\}^n$. (i.e., we have $v \in S_{x_v}$).

Representing a cut with third roots of unity

- $\omega = e^{2\boldsymbol{i}\frac{\pi}{3}} = -\frac{1}{2} + \boldsymbol{i}\frac{\sqrt{3}}{2} \in \mathbb{C}$
- Consider a 3-Cut (S_1, S_2, S_3) defined by $\boldsymbol{x} \in \{0, 1, 2\}^n$. (i.e., we have $v \in S_{x_v}$).
- Let $z_v = \omega^{x_v}$

Representing a cut with third roots of unity

- $\omega = e^{2\boldsymbol{i}\frac{\pi}{3}} = -\frac{1}{2} + \boldsymbol{i}\frac{\sqrt{3}}{2} \in \mathbb{C}$
- Consider a 3-Cut (S_1, S_2, S_3) defined by $\boldsymbol{x} \in \{0, 1, 2\}^n$. (i.e., we have $v \in S_{x_v}$).
- Let $z_v = \omega^{x_v}$
- For all $i, j \in V$,

$$2\,\mathrm{Re}(z_i \bar{z}_j) = z_i \bar{z}_j + z_j \bar{z}_i = \begin{cases} 2 & \text{if } x_i \neq x_j; \\ -1 & \text{otherwise.} \end{cases}$$

Representing a cut with third roots of unity

- $\omega = e^{2\boldsymbol{i}\frac{\pi}{3}} = -\frac{1}{2} + \boldsymbol{i}\frac{\sqrt{3}}{2} \in \mathbb{C}$
- Consider a 3-Cut (S_1, S_2, S_3) defined by $\boldsymbol{x} \in \{0, 1, 2\}^n$. (i.e., we have $v \in S_{x_v}$).
- Let $z_v = \omega^{x_v}$
- For all $i, j \in V$,

$$2\,\mathrm{Re}(z_i \bar{z}_j) = z_i \bar{z}_j + z_j \bar{z}_i = \begin{cases} 2 & \text{if } x_i \neq x_j; \\ -1 & \text{otherwise.} \end{cases}$$

Hence, the weight of the $3-$cut defined by $\boldsymbol{x}$ can be expressed as

$$\mathrm{cut}(S_1, S_2, S_3) = \sum_{i<j} w_{ij} \mathbf{1}_{\{x_i \neq x_j\}} = \sum_{i<j} \frac{w_{ij}}{3}(2 - z_i \bar{z}_j - z_j \bar{z}_i).$$

Complex SDP relaxation of MAX-3-CUT

$$\mathrm{cut}(S_1, S_2, S_3) = \sum_{i<j} w_{ij} \mathbf{1}_{\{x_i \neq x_j\}} = \sum_{i<j} \frac{w_{ij}}{3}(2 - z_i \bar{z}_j - z_j \bar{z}_i).$$

- Define the matrix variable Z with elements $Z_{ij} = z_i \bar{z}_j$

Complex SDP relaxation of MAX-3-CUT

$$\mathrm{cut}(S_1, S_2, S_3) = \sum_{i<j} w_{ij} \mathbf{1}_{\{x_i \neq x_j\}} = \sum_{i<j} \frac{w_{ij}}{3}(2 - z_i \bar{z}_j - z_j \bar{z}_i).$$

- Define the matrix variable Z with elements $Z_{ij} = z_i \bar{z}_j$
- $Z \in \mathcal{H}_+^n$ because $Z = \boldsymbol{z}\boldsymbol{z}^*$, it has rank one, and its diagonal elements are equal to $z_i \bar{z}_i = |z_i|^2 = 1$

Complex SDP relaxation of MAX-3-CUT

$$\mathrm{cut}(S_1, S_2, S_3) = \sum_{i<j} w_{ij} \mathbf{1}_{\{x_i \neq x_j\}} = \sum_{i<j} \frac{w_{ij}}{3}(2 - z_i \bar{z}_j - z_j \bar{z}_i).$$

- Define the matrix variable Z with elements $Z_{ij} = z_i \bar{z}_j$
- $Z \in \mathcal{H}_+^n$ because $Z = \boldsymbol{z}\boldsymbol{z}^*$, it has rank one, and its diagonal elements are equal to $z_i \bar{z}_i = |z_i|^2 = 1$
- The off-diagonal elements of Z are in $R_3 := \{1, \omega, \omega^2\}$

Complex SDP relaxation of MAX-3-CUT

$$\text{cut}(S_1, S_2, S_3) = \sum_{i<j} w_{ij} \mathbf{1}_{\{x_i \neq x_j\}} = \sum_{i<j} \frac{w_{ij}}{3}(2 - z_i\bar{z}_j - z_j\bar{z}_i).$$

- Define the matrix variable Z with elements $Z_{ij} = z_i\bar{z}_j$
- $Z \in \mathcal{H}_+^n$ because $Z = \boldsymbol{z}\boldsymbol{z}^*$, it has rank one, and its diagonal elements are equal to $z_i\bar{z}_i = |z_i|^2 = 1$
- The off-diagonal elements of Z are in $R_3 := \{1, \omega, \omega^2\}$
- We obtain a complex-SDP relaxation by removing the rank constraint, and by imposing $Z_{ij} \in \textbf{conv}\,(R_3), \forall i \neq j$:

$$\begin{aligned}
\underset{Z \in \mathcal{H}^n}{\textbf{maximize}} \quad & \sum_{i<j} \frac{w_{ij}}{3}(2 - Z_{ij} - Z_{ji}) \\
s.t. \quad & Z_{ii} = 1, \qquad \forall i \in V \\
& \alpha Z_{ij} + \bar{\alpha} Z_{ji} \geq -1, \quad \forall \alpha \in R^3,\ \forall i < j \\
& Z \succeq_{\mathcal{H}_+^n} 0.
\end{aligned}$$

Complex semidefinite programming

- How can we solve complex-SDPs?
- What can be said about $\mathcal{H}_+^n$-representability?

Lemma

Let $Z = X + \boldsymbol{i} Y \in \mathcal{H}^n$ be a decomposition of Z into real and imaginary parts, with $X \in \mathbb{S}^n$ and Y skew-symmetric. Then,

$$Z \succeq_{\mathcal{H}_+^n} 0 \iff M = \begin{pmatrix} X & -Y \\ Y & X \end{pmatrix} \succeq 0.$$

Consequence: Any complex-LMI of size n can be recast as a real-valued LMI of size $2n$.

SDP relaxation for MAX-3-CUT

Applying the previous result, we can recast the complex-SDP relaxation of MAX-3-CUT as follows:

$$\begin{aligned}
\underset{X\in\mathbb{S}^n, Y\in\mathbb{R}^{n\times n}}{\textbf{maximize}} \quad & \sum_{i<j} \frac{w_{ij}}{3}(2-2X_{ij}) \\
s.t. \quad & X_{ii} = 1, \qquad \forall i \in V \\
& 2\,\mathrm{Re}(\alpha)X_{ij} - 2\,\mathrm{Im}(\alpha)Y_{ij} \geq -1, \quad \forall \alpha \in R^3,\ \forall i<j \\
& \begin{pmatrix} X & -Y \\ Y & X \end{pmatrix} \succeq 0.
\end{aligned}$$

GW rounding algorithm for MAX-3-CUT

- Solve the complex-SDP, and get a decomposition $Z = UU^*$ of the optimal matrix.
- By construction, the rows $\boldsymbol{u}_i^T$ of U satisfy $Z_{ij} = \langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle$, so we have $\|\boldsymbol{u}_i\| := \sqrt{\boldsymbol{u}_i^* \boldsymbol{u}_i} = (Z_{ii})^{1/2} = 1$.
- Then, generate a random vector $\boldsymbol{p} \in \mathbb{C}^n$ with independent complex Gaussian coordinates
- The randomized rounding is obtained from the argument of the scalar products $\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle = \boldsymbol{u}_i^* \boldsymbol{p} \in \mathbb{C}$:

$$x_i = \begin{cases} 0 \text{ if } \arg\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle \in [0, 2\pi/3) \\ 1 \text{ if } \arg\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle \in [2\pi/3, 4\pi/3) \\ 2 \text{ if } \arg\langle \boldsymbol{p}, \boldsymbol{u}_i \rangle \in [4\pi/3, 2\pi) \end{cases}.$$

Analysis of GW algorithm for MAX-3-CUT

Lemma

If $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = re^{\boldsymbol{i}\theta}$, then the probability that $x_i \neq x_j$ is

$$P(r,\theta) = \frac{2}{3} - \frac{3}{8\pi^2}\Big[2\arccos^2\big(-r\cos(\theta)\big) - \arccos^2\big(-r\cos(\theta+\frac{2\pi}{3})\big) - \arccos^2\big(-r\cos(\theta-\frac{2\pi}{3})\big)\Big].$$

Analysis of GW algorithm for MAX-3-CUT

Lemma

If $\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = re^{\boldsymbol{i}\theta}$, then the probability that $x_i \neq x_j$ is

$$P(r,\theta) = \frac{2}{3} - \frac{3}{8\pi^2}\Big[2\arccos^2\big(-r\cos(\theta)\big) - \arccos^2\big(-r\cos(\theta+\frac{2\pi}{3})\big) - \arccos^2\big(-r\cos(\theta-\frac{2\pi}{3})\big)\Big].$$

Lemma

If $\boldsymbol{z} = re^{\boldsymbol{i}\theta}$, belongs to the triangle $\mathbf{conv}\,(R_3)$, then $P(r,\theta) \geq \psi \cdot \frac{2}{3}(1-\mathrm{Re}(\boldsymbol{z}))$, where $\psi := \frac{7}{12} + \frac{3}{\pi^2}\arccos^2(-\frac{1}{4}) > 0.836008$.

Analysis of GW algorithm for MAX-3-CUT

Theorem (Goemans and Williamson)

Let (S_0, S_1, S_2) be the (random) 3-cut returned by the complex-SDP rounding algorithm. Then,

$$\mathbb{E}[\mathrm{cut}(S_0, S_1, S_2)] \geq \psi \ \mathrm{SDP} \geq \psi \ \text{max-3-cut}(G).$$

Analysis of GW algorithm for MAX-3-CUT

Theorem (Goemans and Williamson)

Let (S_0, S_1, S_2) be the (random) 3-cut returned by the complex-SDP rounding algorithm. Then,

$$\mathbb{E}[\text{cut}(S_0, S_1, S_2)] \geq \psi \text{ SDP} \geq \psi \text{ max-3-cut}(G).$$

Proof. Let $Z_{ij} = \langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle = r_{ij} e^{\boldsymbol{i}\theta_{ij}}$

$$\begin{aligned}
\mathbb{E}[\text{cut}(S_0, S_1, S_2)] &= \mathbb{E}\Big[\sum_{i<j} w_{ij} \mathbf{1}_{\{x_i \neq x_j\}}\Big] \\
&= \sum_{i<j} w_{ij} \mathbb{P}[\boldsymbol{x}_i \neq \boldsymbol{x}_j] \\
&= \sum_{i<j} w_{ij} P(r_{ij}, \theta_{ij}) && \text{(geometric lemma)} \\
&\geq \psi \sum_{i<j} w_{ij} \frac{2}{3}\Big(1 - \text{Re}(\langle \boldsymbol{u}_i, \boldsymbol{u}_j \rangle)\Big) && \text{(definition of } \psi \simeq 0.836) \\
&= \psi \sum_{i<j} \frac{w_{ij}}{3}\Big(2 - Z_{ij} - Z_{ji}\Big) = \psi \text{ SDP}.
\end{aligned}$$

Outline

Outline

Nonconvex QCQP

- Can the SDPs from previous sections be obtained in some systematic manner?
- Yes ! In fact, first round of Lasserre hierarchy (cf. next chapter)

Consider an optimization problem of the form

$$\begin{aligned} \operatorname*{\mathbf{minimize}}_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \boldsymbol{x}^T Q_0 \boldsymbol{x} + \boldsymbol{c}_0^T \boldsymbol{x} + q_0 \\ s.t. \quad & \boldsymbol{x}^T Q_i \boldsymbol{x} + \boldsymbol{c}_i^T \boldsymbol{x} + q_i \lesseqgtr 0, \quad \forall i \in [m], \end{aligned}$$

- Very general problem, very hard
- In particular, can handle binary variables, e.g.

$$x_i \in \{0,1\} \iff x_i^2 = x_i.$$

SDP relaxation

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \boldsymbol{x}^T Q_0 \boldsymbol{x} + \boldsymbol{c}_0^T \boldsymbol{x} + q_0 \\ s.t. \quad & \boldsymbol{x}^T Q_i \boldsymbol{x} + \boldsymbol{c}_i^T \boldsymbol{x} + q_i \leqslant 0, \quad \forall i \in [m], \end{aligned}$$

By introducing the variable $X = \boldsymbol{x}\boldsymbol{x}^T$, everything becomes linear (except the "nasty" constraint $X = \boldsymbol{x}\boldsymbol{x}^T$).

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n, X\in\mathbb{S}^n}{\textbf{minimize}} \quad & \langle Q_0, X\rangle + \boldsymbol{c}_0^T \boldsymbol{x} + q_0 \\ s.t. \quad & \langle Q_i, X\rangle + \boldsymbol{c}_i^T \boldsymbol{x} + q_i \leqslant 0, \quad \forall i \in [m], \\ & X = \boldsymbol{x}\boldsymbol{x}^T. \end{aligned}$$

SDP relaxation

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \boldsymbol{x}^T Q_0 \boldsymbol{x} + \boldsymbol{c}_0^T \boldsymbol{x} + q_0 \\ s.t. \quad & \boldsymbol{x}^T Q_i \boldsymbol{x} + \boldsymbol{c}_i^T \boldsymbol{x} + q_i \lesseqgtr 0, \quad \forall i \in [m], \end{aligned}$$

By introducing the variable $X = \boldsymbol{x}\boldsymbol{x}^T$, everything becomes linear (except the “nasty” constraint $X = \boldsymbol{x}\boldsymbol{x}^T$).

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n, X\in\mathbb{S}^n}{\textbf{minimize}} \quad & \langle Q_0, X\rangle + \boldsymbol{c}_0^T \boldsymbol{x} + q_0 \\ s.t. \quad & \langle Q_i, X\rangle + \boldsymbol{c}_i^T \boldsymbol{x} + q_i \lesseqgtr 0, \quad \forall i \in [m], \\ & X = \boldsymbol{x}\boldsymbol{x}^T. \end{aligned}$$

We can relax this constraint to $X \succeq \boldsymbol{x}\boldsymbol{x}^T \iff \begin{pmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{pmatrix} \succeq 0$

SDP relaxation of nonconvex QCQPs

Proposition

The SDP

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n, X\in\mathbb{S}^n}{\textbf{minimize}} \quad & \langle Q_0, X\rangle + \boldsymbol{c}_0^T\boldsymbol{x} + q_0 \\ s.t. \quad & \langle Q_i, X\rangle + \boldsymbol{c}_i^T\boldsymbol{x} + q_i \lesseqgtr 0, \quad \forall i \in [m], \\ & \begin{pmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{pmatrix} \succeq 0 \end{aligned}$$

is a relaxation of the nonconvex QCQP: its optimal value gives a lower bound for the original problem.

Alternative interpretation of relaxation

As we did for MAXCUT, an alternative interpretation of the SDP relaxation is that we removed a rank-one constraint:

Lemma

If (X, x) solves the SDP-relaxation and

$$\textbf{rank} \begin{pmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{pmatrix} = 1,$$

then x is also optimal for the original nonconvex QCQP.

Proof.

$$\begin{aligned} \begin{pmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{pmatrix} \succeq 0 \text{ is of rank } 1 \iff & \begin{pmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{pmatrix} = \begin{bmatrix} \boldsymbol{u} \\ \alpha \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ \alpha \end{bmatrix}^T \text{ for some } \boldsymbol{u} \in \mathbb{R}^n, \alpha \in \mathbb{R} \\ \iff & X = \boldsymbol{u}\boldsymbol{u}^T \text{ for } \boldsymbol{u} = \alpha \boldsymbol{x}, \quad \alpha = \pm 1 \\ \iff & X = \boldsymbol{x}\boldsymbol{x}^T. \end{aligned}$$

Outline

Completely positive & Copositive cones

Definition (Copositive cone).

The cone of $n \times n$ copositive matrices is

$$\mathcal{C}_n := \{X \in \mathbb{S}^n | \; \boldsymbol{u}^T X \boldsymbol{u} \geq 0, \; \forall \boldsymbol{u} \in \mathbb{R}^n_+\}.$$

Note the difference with $\mathbb{S}^n_+$: restriction "$\forall \boldsymbol{u} \geq \mathbf{0}$".

Completely positive & Copositive cones

Definition (Copositive cone).

The cone of $n \times n$ copositive matrices is

$$\mathcal{C}_n := \{X \in \mathbb{S}^n | \, \boldsymbol{u}^T X \boldsymbol{u} \geq 0, \ \forall \boldsymbol{u} \in \mathbb{R}^n_+\}.$$

Note the difference with $\mathbb{S}^n_+$: restriction "$\forall \boldsymbol{u} \geq \mathbf{0}$".

Similarly, define

Definition (Completely positive cone).

The cone of $n \times n$ completely positive matrices is

$$\mathcal{C}_n^* := \{\sum_{k=1}^{q} \boldsymbol{u}_k \boldsymbol{u}_k^T | \, q \in \mathbb{N}, \ \boldsymbol{u}_k \in \mathbb{R}^n_+, \ \forall k \in [q]\}.$$

Properties of $\mathcal{C}_n$ and $\mathcal{C}_n^*$

Proposition

(i) We require at most $q \leq n(n+1)/2$ vectors to decompose a completely positive matrix:

$$X \in \mathcal{C}_n^* \iff \exists \ell \leq \frac{1}{2}n(n+1), \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\ell \in \mathbb{R}_+^n : X = \sum_{j=1}^{\ell} \boldsymbol{x}_i \boldsymbol{x}_i^T.$$

(ii) The following inclusions hold:

$$\mathcal{C}_n^* \subseteq \left(\mathbb{S}_+^n \cap \mathbb{R}_+^{n\times n}\right) \subseteq \mathbb{S}_+^n \subseteq \left(\mathbb{S}_+^n + \mathbb{R}_+^{n\times n}\right) \subseteq \mathcal{C}_n.$$

(iii) The cones $\mathcal{C}_n$ and $\mathcal{C}_n^*$ are proper, and dual from each other.

Outline

SDP relaxation of binary QPs

Now, consider a mixed-integer QP of the form

$$\begin{aligned} \underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} \\ s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{x} = b_i,\ \forall i \in [m] \\ & \boldsymbol{x} \geq \boldsymbol{0} \\ & x_i \in \{0,1\},\ \forall i \in B, \end{aligned}$$

where B is a subset of $[n]$.

SDP relaxation of binary QPs

Now, consider a mixed-integer QP of the form

$$\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{c}^T \boldsymbol{x} \\
s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{x} = b_i,\ \forall i \in [m] \\
& \boldsymbol{x} \geq \boldsymbol{0} \\
& x_i \in \{0,1\},\ \forall i \in B,
\end{aligned}$$

where B is a subset of $[n]$. By applying the general recipe seen earlier, we obtain the following SDP relaxation:

$$\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \langle Q, X \rangle + \boldsymbol{c}^T \boldsymbol{x} \\
s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{x} = b_i,\ \forall i \in [m] \\
& \boldsymbol{x} \geq \boldsymbol{0} \\
& X_{ii} = x_i,\ \forall i \in B, \\
& \begin{pmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{pmatrix} \succeq 0
\end{aligned}$$

Burer's $\mathcal{C}_n^*$- programming formulation

- The relaxation can be strengthened by adding the squared equalities $(\boldsymbol{a}_i^T \boldsymbol{x})^2 = b_i^2$ in the original formulation, which become $\boldsymbol{a}_i^T X \boldsymbol{a}_i = b_i^2$
- In addition, $\boldsymbol{x} \geq \boldsymbol{0}$ implies that $\begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ 1 \end{bmatrix}^T \in \mathcal{C}_n^*$.

Theorem (Burer).

Under mild assumptions, the following completely positive program is equivalent to the mixed-integer QP.

$$
\begin{aligned}
\underset{\boldsymbol{x}\in\mathbb{R}^n}{\textbf{minimize}} \quad & \langle Q, X\rangle + \boldsymbol{c}^T \boldsymbol{x} \\
s.t. \quad & \boldsymbol{a}_i^T \boldsymbol{x} = b_i,\ \forall i \in [m] \\
& \boldsymbol{a}_i^T X \boldsymbol{a}_i = b_i^2,\ \forall i \in [m] \\
& X_{ii} = x_i,\ \forall i \in B, \\
& \begin{pmatrix} X & \boldsymbol{x} \\ \boldsymbol{x}^T & 1 \end{pmatrix} \succeq_{\mathcal{C}_n^*} 0
\end{aligned}
$$

Burer's result: Is something wrong?

- First result of this kind was obtained for the special case of the stable set problem
- So NP-hard problems can be formulated as convex conic optimization problems over $\mathcal{C}_n^*$
- In fact, it is NP-hard to solve the separation problem for $\mathcal{C}_n^*$ (or even to test membership)
- Nevertheless, we can use the completely positive formulation to construct hierarchies of SDP converging to the optimal value (cf. next chapter)