Semidefinite Relaxations for Quantum Max Cut

Igor Klep  University of Ljubljana, Slovenia

POP23 - Future Trends in Polynomial OPtimization
LAAS-CNRS, Toulouse, November 2023

Based on joint works with

Adam Bene Watts  U Waterloo
Anirban Chowdhury  U Waterloo
Aidan Epperly  UC Davis
Bill Helton  UC San Diego
Tea Štrekelj  U Ljubljana
Jurij Volčič  Drexel U
Outline

Classical Max Cut
   Goemans & Williamson = 1st Lasserre relaxation

Quantum Max Cut
   Physics motivation
   Pauli matrices
   2-local Hamiltonian problem

Swap operators
   Schur-Weyl duality
   Noncommutative Lasserre relaxations
   Numerical examples
   Rounding

Exact solutions
   Clique
   Star graph
   An algorithm

Takeaway messages
Classical Max Cut

- Graph $G$ with $n$ vertices
  - $V(G) =$ vertices of the graph $G$
  - $E(G) =$ edges of $G$.
- $w_{ij} =$ weight of the edge from $i$ to $j$ in $G$.
  - If no edge exists, the weight is assumed to be zero.
  - Most of the time we will use $w_{ij} = 1$ for $(i, j) \in E(G)$.

A max cut of $G$ is a subset $S$ of $V(G)$ such that

$$\sum_{i \in S} \sum_{j \notin S} w_{i,j}$$

is maximized. This can be thought of as coloring all the vertices in $S$ red and all of the vertices in $V(G) \setminus S$ green and then summing the weights of all the bicolored edges (edges connecting red vertices to green vertices).
Classical Max Cut

Alternatively, assign to each vertex $i$ a value $x_i \in \{-1, 1\}$ in such a way as to maximize

$$\sum_{(i,j) \in E(G)} w_{ij} \frac{1 - x_i x_j}{2}.$$
Classical Max Cut

Alternatively, assign to each vertex $i$ a value $x_i \in \{-1, 1\}$ in such a way as to maximize

$$\sum_{(i,j) \in E(G)} w_{ij} \frac{1 - x_i x_j}{2}.$$
Classical Max Cut

Goemans & Williamson SDP relaxation $X = xx^T$

The **SDP relaxation** of the Max Cut

$$c_{\text{max}} = \max_{x \in \mathbb{R}^n} \left\{ \sum_{(i,j) \in E(G)} w_{ij} \frac{1 - x_i x_j}{2} \mid x \in \{\pm 1\}^n \right\}$$  \hspace{1cm} (MC)

of Goemans and Williamson is

$$c_{\text{max}}^{GW} = \max_{X \in \mathbb{S}_n} \left\{ \sum_{(i,j) \in E(G)} w_{ij} \frac{1 - X_{ij}}{2} \mid X \succeq 0, X_{ii} = 1 \right\}. \hspace{1cm} (GW)$$
Classical Max Cut

Goemans & Williamson SDP relaxation \( X = xx^T \)

The SDP relaxation of the Max Cut

\[
c_{\text{max}} = \max_{x \in \mathbb{R}^n} \left\{ \sum_{(i,j) \in E(G)} w_{ij} \frac{1 - x_i x_j}{2} \mid x \in \{\pm 1\}^n \right\}
\]

of Goemans and Williamson is

\[
c_{\text{GW}}^{\text{max}} = \max_{X \in \mathbb{S}_n} \left\{ \sum_{(i,j) \in E(G)} w_{ij} \frac{1 - X_{ij}}{2} \mid X \succeq 0, X_{ii} = 1 \right\}
\]

(GW) is the same as the first level of Lasserre’s Moment-SOS hierarchy for solving (MC).
Classical Max Cut

Goemans & Williamson SDP relaxation $X = xx^T$

The SDP relaxation of the Max Cut

$$c_{\text{max}} = \max_{x \in \mathbb{R}^n} \left\{ \sum_{(i,j) \in E(G)} w_{ij} \frac{1 - x_i x_j}{2} \mid x \in \{\pm 1\}^n \right\}$$

of Goemans and Williamson is

$$c^{GW}_{\text{max}} = \max_{X \in \mathbb{S}_n} \left\{ \sum_{(i,j) \in E(G)} w_{ij} \frac{1 - X_{ij}}{2} \mid X \succeq 0, X_{ii} = 1 \right\}.$$  \hspace{1cm} (GW)

\(\text{info (Goemans & Williamson)}\) $c_{\text{max}} \leq c^{GW}_{\text{max}} \leq \frac{1}{0.878} \cdot c_{\text{max}}$
Quantum Max Cut (QMC)

Physics motivation

See/recall Hamza’s talk

- QMC (a special local Hamiltonian problem) was named by Gharibian & Parekh (2019);

- QMC is a natural maximization variant of the anti-ferromagnetic Heisenberg XYZ model;

- MC is NP-hard,
  QMC is a prototype of a QMA-hard problem.
  Piddock & Montanaro (2017), Cubitt & Montanaro (2016)
Quantum Max Cut

Pauli matrices

The Pauli matrices are the following three self-adjoint $2 \times 2$ matrices

$$
\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.
$$

(Pauli)
Quantum Max Cut

Pauli matrices

The Pauli matrices are the following three self-adjoint $2 \times 2$ matrices

$$\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$ (Pauli)

Their multiplication table is as follows:

<table>
<thead>
<tr>
<th></th>
<th>$\sigma_X$</th>
<th>$\sigma_Y$</th>
<th>$\sigma_Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma_X$</td>
<td>$I_2$</td>
<td>$i \sigma_Z$</td>
<td>$-i \sigma_Y$</td>
</tr>
<tr>
<td>$\sigma_Y$</td>
<td>$-i \sigma_Z$</td>
<td>$I_2$</td>
<td>$i \sigma_X$</td>
</tr>
<tr>
<td>$\sigma_Z$</td>
<td>$i \sigma_Y$</td>
<td>$-i \sigma_X$</td>
<td>$I_2$</td>
</tr>
</tbody>
</table>
Quantum Max Cut

Pauli matrices

The Pauli matrices are the following three self-adjoint $2 \times 2$ matrices

$$
\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.
$$

(Pauli)

For $W \in \{X, Y, Z\}$ and $k, n \in \mathbb{N}$ we shall also use

$$
\sigma^k_W = I_{2} \otimes \cdots \otimes I_{2} \otimes \sigma_W \otimes I_{2} \otimes \cdots \otimes I_{2} \in M_2(\mathbb{C})^{\otimes n} = M_{2^n}(\mathbb{C}).
$$
The Pauli matrices are the following three self-adjoint $2 \times 2$ matrices:

$$
\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.
$$

For $W \in \{X, Y, Z\}$ and $k, n \in \mathbb{N}$ we shall also use

$$
\sigma^k_W = I_2 \otimes \cdots \otimes I_2 \otimes \sigma_W \otimes I_2 \otimes \cdots \otimes I_2 \in M_2(\mathbb{C})^\otimes n = M_{2^n}(\mathbb{C}).
$$

Letting $\sigma_I := I_2$, observe that

$$
\{\sigma^1_{W_1} \sigma^2_{W_2} \cdots \sigma^n_{W_n} \mid W_j \in \{I, X, Y, Z\}\}
$$

is a basis of $M_2(\mathbb{C})^\otimes n$. 
**Quantum Max Cut**

**Pauli matrices**

The **Pauli matrices** are the following three self-adjoint $2 \times 2$ matrices

\[
\sigma_X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad \text{(Pauli)}
\]

For $W \in \{X, Y, Z\}$ and $k, n \in \mathbb{N}$ we shall also use

\[
\sigma^k_W = I_2 \otimes \cdots \otimes I_2 \otimes \sigma_W \otimes I_2 \otimes \cdots \otimes I_2 \in M_2(\mathbb{C})^{\otimes n} = M_{2^n}(\mathbb{C}).
\]

Letting $\sigma_I := I_2$, observe that

\[
\{\sigma^1_{W_1} \sigma^2_{W_2} \cdots \sigma^n_{W_n} \mid W_j \in \{I, X, Y, Z\}\}
\]

is a basis of $M_2(\mathbb{C})^{\otimes n}$.

Given $i \neq j$, then $\sigma^i_W, \sigma^j_W$, commute:

\[
\sigma^i_W \sigma^j_W = \sigma^j_W \sigma^i_W.
\]
Quantum Max Cut

QMC Hamiltonian (Pauli Form)

The QMC Hamiltonian of a graph $G$ is given by

$$H_G = \sum_{(i,j) \in E(G)} \omega_{ij} \left( I - \sigma^i_X \sigma^j_X - \sigma^i_Y \sigma^j_Y - \sigma^i_Z \sigma^j_Z \right) \in \mathbb{S}_{2^n}$$

where the $\sigma_W$ are Pauli matrices and

$$\sigma^k_W = I_2 \otimes \cdots \otimes I_2 \otimes \sigma_W \otimes I_2 \otimes \cdots \otimes I_2.$$

$\sum_{(i,j) \in E(G)} \omega_{ij} \left( I - \sigma^i_X \sigma^j_X - \sigma^i_Y \sigma^j_Y - \sigma^i_Z \sigma^j_Z \right) \in \mathbb{S}_{2^n}$

where the $\sigma_W$ are Pauli matrices and

$$\sigma^k_W = I_2 \otimes \cdots \otimes I_2 \otimes \sigma_W \otimes I_2 \otimes \cdots \otimes I_2.$$
Quantum Max Cut

QMC Hamiltonian (Pauli Form)
The QMC Hamiltonian of a graph $G$ is given by

$$H_G = \sum_{(i,j) \in E(G)} w_{ij} \left( I - \sigma^i_X \sigma^j_X - \sigma^i_Y \sigma^j_Y - \sigma^i_Z \sigma^j_Z \right) \in S_{2^n}$$

where the $\sigma_W$ are Pauli matrices and

$$\sigma^k_W = I_2 \otimes \cdots \otimes I_2 \otimes \sigma_W \otimes I_2 \otimes \cdots \otimes I_2.$$

Quantum Max Cut
QMC asks for the biggest eigenvalue of $H_G$
(and, if possible, the associated eigenvector/state).
Quantum Max Cut

Pauli-based SDP relaxations
Quantum Max Cut

SWAP operators

The matrix

\[
\text{Swap}_{ij} = \frac{1}{2}(I + \sigma^i_X \sigma^j_X + \sigma^i_Y \sigma^j_Y + \sigma^i_Z \sigma^j_Z)
\]

is called a **SWAP operator**.
Quantum Max Cut

SWAP operators

The matrix

\[ \text{Swap}_{ij} = \frac{1}{2} (I + \sigma^i_X \sigma^j_X + \sigma^i_Y \sigma^j_Y + \sigma^i_Z \sigma^j_Z) \]

is called a SWAP operator.

For instance, if \( n = 2 \), then

\[
\text{Swap}_{12} = \begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\]
Quantum Max Cut

SWAP operators

The matrix

$$\text{Swap}_{ij} = \frac{1}{2}(I + \sigma_i^X \sigma_j^X + \sigma_i^Y \sigma_j^Y + \sigma_i^Z \sigma_j^Z)$$

is called a SWAP operator.

Thus, we can rewrite the QMC Hamiltonian as

QMC Hamiltonian (SWAP Form)

$$H_G = \sum_{(i,j) \in E(G)} 2w_{ij}(I - \text{Swap}_{ij})$$
Quantum Max Cut

SWAP operators

The SWAP operator

$$\text{Swap}_{ij} = \frac{1}{2}(I + \sigma^i_X \sigma^j_X + \sigma^i_Y \sigma^j_Y + \sigma^i_Z \sigma^j_Z)$$

sends the rank one tensor

$$v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n \in (\mathbb{C}^2)^\otimes n$$

to the rank one tensor

$$v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n \in (\mathbb{C}^2)^\otimes n,$$

where $v_k \in \mathbb{C}^2$. 

Permutations
Quantum Max Cut

SWAP operators

The SWAP operator

\[
\text{Swap}_{ij} = \frac{1}{2}(I + \sigma^i_X \sigma^j_X + \sigma^i_Y \sigma^j_Y + \sigma^i_Z \sigma^j_Z)
\]

sends the rank one tensor

\[
v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n \in (\mathbb{C}^2)^{\otimes n}
\]

to the rank one tensor

\[
v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n \in (\mathbb{C}^2)^{\otimes n},
\]

where \(v_k \in \mathbb{C}^2\).

Let \(M_n^{\text{Swap}}\) be the SWAP algebra generated by the \(\text{Swap}_{ij}\) inside \(M_{2n}(\mathbb{C})\).
SWAP operators

\[ \text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n. \]

Some relations satisfied by the SWAP operators (indices \(i, j, k, l\) distinct):
SWAP operators

\[
\text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n.
\]

Some relations satisfied by the SWAP operators (indices \(i, j, k, l\) distinct):

\[
\begin{align*}
\text{Swap}_{ij}^2 &= I_2, \\
\text{Swap}_{ij} \text{ Swap}_{jk} &= \text{ Swap}_{ik} \text{ Swap}_{ij}, \\
\text{Swap}_{ij} \text{ Swap}_{kl} &= \text{ Swap}_{kl} \text{ Swap}_{ij}.
\end{align*}
\]

\begin{center}
\rightbrace
\end{center}

\text{symmetric group}
SWAP operators

\[ \text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n. \]

Some relations satisfied by the SWAP operators (indices \( i, j, k, l \) distinct):

\[ \text{Swap}_{ij}^2 = I_2, \]
\[ \text{Swap}_{ij} \text{Swap}_{jk} = \text{Swap}_{ik} \text{Swap}_{ij}, \]
\[ \text{Swap}_{ij} \text{Swap}_{kl} = \text{Swap}_{kl} \text{Swap}_{ij}. \]

\[ \text{Swap}_{ij} \text{Swap}_{jk} + \text{Swap}_{jk} \text{Swap}_{ij} = \text{Swap}_{ij} + \text{Swap}_{jk} + \text{Swap}_{ik} - I_2. \]
SWAP algebra

Symmetric group

\[ \text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n. \]

Since the transpositions \((i, j)\) generate the symmetric group \(S_n\), the map

\[ (i, j) \mapsto \text{Swap}_{ij} \]

gives a representation of the symmetric group \(S_n\) on \((\mathbb{C}^2)^\otimes n\).
**SWAP algebra**

**Symmetric group**

\[
\text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n.
\]

Since the transpositions \((i, j)\) generate the symmetric group \(S_n\), the map

\[
(i, j) \mapsto \text{Swap}_{ij}
\]

gives a representation of the symmetric group \(S_n\) on \((\mathbb{C}^2)^{\otimes n}\).

By Maschke’s Theorem (the group algebra \(\mathbb{C}S_n\) is semisimple), this representation decomposes into a direct sum of irreps (\(=\)irreducible representations).
**SWAP algebra**

**Symmetric group**

\[
\text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n.
\]

Since the transpositions \((i, j)\) generate the symmetric group \(S_n\), the map

\[
(i, j) \mapsto \text{Swap}_{ij}
\]

gives a **representation of the symmetric group** \(S_n\) on \((\mathbb{C}^2)^\otimes n\).

By Maschke’s Theorem (the group algebra \(\mathbb{C}S_n\) is semisimple), this representation decomposes into a **direct sum of irreps** (=irreducible representations).

It is **well known** that the irreps of the symmetric group \(S_n\) are indexed by partitions \(\lambda\) of \(n\), or equivalently, **Young diagrams**:  

\[
\ldots \mathcal{S}_\lambda
\]
SWAP algebra

Schur-Weyl duality

$$\text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n.$$ 

$\text{GL}_2(\mathbb{C})$ also acts on $(\mathbb{C}^2)^{\otimes n}$:

$$g \cdot (v_1 \otimes \cdots \otimes v_n) = gv_1 \otimes \cdots \otimes gv_n.$$
**SWAP algebra**

**Schur-Weyl duality**

\[ \text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n. \]

\( \text{GL}_2(\mathbb{C}) \) also acts on \((\mathbb{C}^2)^\otimes n\):

\[ g \cdot (v_1 \otimes \cdots \otimes v_n) = gv_1 \otimes \cdots \otimes gv_n. \]

- This action commutes with the action of the SWAP operators:
  \[ \text{Swap}_{ij} \circ g = g \circ \text{Swap}_{ij}. \]
\textbf{SWAP algebra} \\
Schur-Weyl duality

\[
\text{Swap}_{ij} (v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n.
\]

\(\text{GL}_2(\mathbb{C})\) also acts on \((\mathbb{C}^2)^\otimes n\):

\[
g \cdot (v_1 \otimes \cdots \otimes v_n) = gv_1 \otimes \cdots \otimes gv_n.
\]

- This action commutes with the action of the SWAP operators:

\[
\text{Swap}_{ij} \circ g = g \circ \text{Swap}_{ij}.
\]

- Irreps of \(\text{GL}_2(\mathbb{C})\) are indexed by \textbf{two row Young diagrams} with an arbitrary number of boxes.

\[\cdots \mathcal{L}[n-k,k] \]
SWAP algebra

Schur-Weyl duality

\[
\text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n.
\]

\(\text{GL}_2(\mathbb{C})\) also acts on \((\mathbb{C}^2)^\otimes n:\)

\[
g \cdot (v_1 \otimes \cdots \otimes v_n) = gv_1 \otimes \cdots \otimes gv_n.
\]

**Theorem (Schur-Weyl duality)**

The space \((\mathbb{C}^2)^\otimes n\) decomposes under the action of \(\text{GL}_2(\mathbb{C}) \times S_n\) as

\[
(\mathbb{C}^2)^\otimes n \cong \bigoplus_{k=0}^{[n/2]} \mathcal{L}_{n-k,k} \otimes \mathcal{L}_{n-k,k}.
\]

In particular, as \(S_n\)-module (or SWAP algebra-module),

\[
(\mathbb{C}^2)^\otimes n \cong \bigoplus_{k=0}^{[n/2]} \left(\mathcal{L}_{n-k,k}\right)^{\text{dim} \mathcal{L}_{n-k,k}}.
\]
SWAP algebra
Schur-Weyl duality (cont’d)

\[ \text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n. \]

**Corollary**

The Swap Matrix Algebra \( M^\text{Swap}_n \) is the direct sum of simple algebras generated by the **two row irreps** of the symmetric group \( S_n \):

\[
M^\text{Swap}_n \cong \bigoplus_{k=0}^{\lfloor \frac{n}{2} \rfloor} M_{\dim \mathcal{A}_{[n-k,k]}(\mathbb{C})}
\]
**SWAP algebra**

Schur-Weyl duality (cont’d)

\[ \text{Swap}_{ij}(v_1 \otimes \cdots \otimes v_i \otimes \cdots \otimes v_j \otimes \cdots \otimes v_n) = v_1 \otimes \cdots \otimes v_j \otimes \cdots \otimes v_i \otimes \cdots \otimes v_n. \]

**Corollary**

The Swap Matrix Algebra \( M_n^{\text{Swap}} \) is the direct sum of simple algebras generated by the two row irreps of the symmetric group \( S_n \):

\[ M_n^{\text{Swap}} \cong \bigoplus_{k=0}^\left\lfloor \frac{n}{2} \right\rfloor M_{\dim \mathscr{L}_{[n-k,k]}(\mathbb{C})} \]

**Theorem**

The Swap Matrix Algebra \( M_n^{\text{Swap}} \) is given by the following presentation:

\[ M_n^{\text{Swap}} \cong \mathbb{C}\langle \text{Swap}_{ij} \rangle / \mathcal{J}_{\text{Swap}}, \text{ where } \mathcal{J}_{\text{Swap}} \text{ is the ideal generated by} \]

\[ \text{Swap}_{ij}^2 = I_2, \]
\[ \text{Swap}_{ij} \text{ Swap}_{jk} = \text{Swap}_{ik} \text{ Swap}_{ij}, \]
\[ \text{Swap}_{ij} \text{ Swap}_{kl} = \text{Swap}_{kl} \text{ Swap}_{ij}, \]

\[ \text{Swap}_{ij} \text{ Swap}_{jk} + \text{Swap}_{jk} \text{ Swap}_{ij} = \text{Swap}_{ij} + \text{Swap}_{jk} + \text{Swap}_{ik} - I_2. \]

\[ \dim M_n^{\text{Swap}} = \sum_{k=0}^{\left\lfloor \frac{n}{2} \right\rfloor} \left( \frac{n-2k+1}{n-k+1} \binom{n}{k} \right)^2 = \frac{1}{n+1} \binom{2n}{n} \text{ is the } n\text{-th Catalan number } C_n. \]
To $h \in \mathbb{C}\langle\text{Swap}\rangle$ let

$$\nu_d(h) := \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + J_{\text{Swap}} \},$$

where $\text{SOS}_{2d}$ denotes the set of all sums of squares of polynomials in the variables $\text{Swap}_{ij}$ each having degree $\leq d$. 
To $h \in \mathbb{C}\langle\text{Swap}\rangle$ let

$$\nu_d(h) := \min \{ \nu \mid \nu - h \in \text{SOS}_2 + \mathcal{J}_{\text{Swap}} \},$$

where $\text{SOS}_2$ denotes the set of all sums of squares of polynomials in the variables $\text{Swap}_{ij}$ each having degree $\leq d$. 

- $\nu_d(h) \geq \text{eig}_{\text{max}} h(\text{Swap})$
To $h \in \mathbb{C}\langle\text{Swap}\rangle$ let

$$\nu_d(h) := \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + \mathcal{J}_{\text{Swap}} \} ,$$

where $\text{SOS}_{2d}$ denotes the set of all sums of squares of polynomials in the variables $\text{Swap}_{ij}$ each having degree $\leq d$.

- $\nu_d(h) \geq \text{eig}_{\text{max}} h(\text{Swap})$

- $\nu_{\lceil n/2 \rceil}(h) = \text{eig}_{\text{max}} h(\text{Swap})$
To $h \in \mathbb{C}\langle \text{Swap} \rangle$ let

$$\nu_d(h) := \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + \mathcal{J}_{\text{Swap}} \},$$

where $\text{SOS}_{2d}$ denotes the set of all sums of squares of polynomials in the variables $\text{Swap}_{ij}$ each having degree $\leq d$.

Veroneses are column vectors $V_d(n)$, which consist of degree $d$ monomials in the $n(n - 1)/2$ variables $\text{Swap}_{ij}$, $i < j$, ordered w.r.t. grlex.
To $h \in \mathbb{C}\langle\text{Swap}\rangle$ let

$$
\nu_d(h) := \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + \mathcal{J}_{\text{Swap}} \},
$$

where \(\text{SOS}_{2d}\) denotes the set of all sums of squares of polynomials in the variables \(\text{Swap}_{ij}\) each having degree \(\leq d\).

Veroneses are column vectors \(V_d(n)\), which consist of degree \(d\) monomials in the \(n(n-1)/2\) variables \(\text{Swap}_{ij}, \ i < j\), ordered w.r.t. grlex.

**Lemma**

Let \(h \in \mathbb{C}\langle\text{Swap}\rangle\). Then \(h \in \text{SOS}_{2d} + \mathcal{J}_{\text{Swap}}\) iff there is a PsD matrix \(\Gamma\) such that

$$
h - V_d(n)^* \Gamma V_d(n) \in \mathcal{J}_{\text{Swap}}.
$$

Finding such a \(\Gamma\) can be done with an SDP.

Given a “good” generating set (e.g., a Gröbner basis) for \(\mathcal{J}_{\text{Swap}}\).
\[ \nu_d(h) = \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + \mathcal{J}_{\text{Swap}} \}, \]

\[ \eta_d(h) = \max L(h) \]

s.t. \( L \in (\text{SOS}_{2d} + \mathcal{J}_{\text{Swap}})^\vee \)

\[ L(1) = 1. \]

Here \((\text{SOS}_{2d} + \mathcal{J}_{\text{Swap}})^\vee\) denotes the dual cone to the cone \(\text{SOS}_{2d} + \mathcal{J}_{\text{Swap}},\)

\[ (\text{SOS}_{2d} + \mathcal{J}_{\text{Swap}})^\vee = \left\{ L : \mathbb{C}\langle\text{Swap}\rangle_{2d} \to \mathbb{C} \mid L \text{ linear with } L(\text{SOS}_{2d}) \subseteq \mathbb{R}_{\geq 0}, \right. \]

\[ \left. L(\mathcal{J}_{\text{Swap}} \cap \mathbb{C}\langle\text{Swap}\rangle_{2d}) = \{0\} \right\}. \]
nc Lasserre relaxations (cont’d)

\[ \nu_d(h) = \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + J_{\text{Swap}} \} , \]

\[ \alpha_d(h) = \max L(h) \]
\[ \text{s.t. } L \in (\text{SOS}_{2d} + J_{\text{Swap}})^\vee \]
\[ L(1) = 1. \]

Here \((\text{SOS}_{2d} + J_{\text{Swap}})^\vee\) denotes the dual cone to the cone \(\text{SOS}_{2d} + J_{\text{Swap}}\).

This is another SDP.

- (strong duality) \( \alpha_d(h) = \nu_d(h) \).
- (pseudomoments) Implement \( \alpha_d(h) \) with the help of moment matrices.
\( \nu_d(h) = \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + J_{\text{Swap}} \} , \)

\( \alpha_d(h) = \max \{ L(h) \mid L \in (\text{SOS}_{2d} + J_{\text{Swap}}) \lor , L(1) = 1 \} . \)

Take \( n = 3, \ d = 1. \) Then \( V_1(3) = (1, s_{12}, s_{13}, s_{23})^* \)

The symbolic Hankel matrix is

\[
\mathcal{M}_1(3) = V_1(3)V_1(3)^* = \begin{bmatrix}
1 & s_{12} & s_{13} & s_{23} \\
 s_{12} & s_{12}^2 & s_{12}s_{13} & s_{12}s_{23} \\
 s_{13} & s_{13}s_{12} & s_{13}^2 & s_{13}s_{23} \\
 s_{23} & s_{23}s_{12} & s_{23}s_{13} & s_{23}^2
\end{bmatrix}
\]

and the pseudomoments of \( L \in (\text{SOS}_{2d} + J_{\text{Swap}}) \lor \) are

\[
\mathcal{M}_1(L) = \begin{bmatrix}
 L(1) & L(s_{12}) & L(s_{13}) & L(s_{23}) \\
 L(s_{12}) & L(s_{12}^2) & L(s_{12}s_{13}) & L(s_{12}s_{23}) \\
 L(s_{13}) & L(s_{13}s_{12}) & L(s_{13}^2) & L(s_{13}s_{23}) \\
 L(s_{23}) & L(s_{23}s_{12}) & L(s_{23}s_{13}) & L(s_{23}^2)
\end{bmatrix}
\]
\( n = 3, \ d = 1, \ V_1(3) = (1, s_{12}, s_{13}, s_{23})^* \)

The space of **quadratics** in the SWAPs is spanned by the entries of \( V_1(3) \) together with one element, e.g., \( s_{12}s_{13} \).
The space of **quadratics** in the SWAPs is spanned by the entries of $V_1(3)$ together with one element, e.g., $s_{12}s_{13}$. Namely,

\[
\begin{align*}
    s_{i,j}^2 &= 1 \\
    s_{13}s_{23} &= s_{12}s_{13} \\
    s_{23}s_{12} &= s_{12}s_{13} \\
    s_{12}s_{23} &= -1 + s_{12} + s_{13} + s_{23} - s_{12}s_{13} \\
    s_{13}s_{12} &= -1 + s_{12} + s_{13} + s_{23} - s_{12}s_{13} \\
    s_{23}s_{13} &= -1 + s_{12} + s_{13} + s_{23} - s_{12}s_{13}
\end{align*}
\]
\[ n = 3, \ d = 1, \ V_1(3) = (1, s_{12}, s_{13}, s_{23})^* \]

The space of **quadratics** in the SWAPs is spanned by the entries of \( V_1(3) \) together with one element, e.g., \( s_{12}s_{13} \). Namely,

\[
\begin{align*}
  s_{ij}^2 &= 1, & s_{12}s_{23} &= -1 + s_{12} + s_{13} + s_{23} - s_{12}s_{13} \\
  s_{13}s_{23} &= s_{12}s_{13}, & s_{13}s_{12} &= -1 + s_{12} + s_{13} + s_{23} - s_{12}s_{13} \\
  s_{23}s_{12} &= s_{12}s_{13}, & s_{23}s_{13} &= -1 + s_{12} + s_{13} + s_{23} - s_{12}s_{13}
\end{align*}
\]

With this the **pseudomoments** of \( L \in (\text{SOS}_{2d} + J_{\text{Swap}})^\lor \) simplify

\[
\mathcal{M}_1(L) = \begin{bmatrix}
  L(1) & L(s_{12}) & L(s_{13}) & L(s_{23}) \\
  L(s_{12}) & L(s_{12}^2) & L(s_{12}s_{13}) & L(s_{12}s_{23}) \\
  L(s_{13}) & L(s_{13}s_{12}) & L(s_{13}^2) & L(s_{13}s_{23}) \\
  L(s_{23}) & L(s_{23}s_{12}) & L(s_{23}s_{13}) & L(s_{23}^2)
\end{bmatrix}
\]

\[
= \begin{bmatrix}
  1 & \ell_{12} & \ell_{13} & \ell_{23} \\
  \ell_{12} & 1 & q & 1 + \ell_{12} + \ell_{13} + \ell_{23} - q \\
  \ell_{13} & q^* & 1 & q \\
  \ell_{23} & -1 + \ell_{12} + \ell_{13} + \ell_{23} - q^* & q^* & 1
\end{bmatrix},
\]

where \( \ell_{ij} = L(s_{ij}) \) and \( q = L(s_{12}s_{13}) \).
\[ \nu_d(h) = \min \{ \nu \mid \nu - h \in \text{SOS}_{2d} + \mathcal{J}_{\text{Swap}} \} , \]

\[ \alpha_d(h) = \max \{ L(h) \mid L \in (\text{SOS}_{2d} + \mathcal{J}_{\text{Swap}})^\vee , L(1) = 1 \} . \]

We can now rewrite \( \alpha_d(h) \) as an SDP as follows:

\[ \alpha_d(h) = \max \langle \mathcal{M}_d(L), \Gamma_h \rangle \]

s.t. \( \mathcal{M}_d(L) \succeq 0 \)

\[ \mathcal{M}_d(L)_{1,1} = 1 \]

\[ L(\mathcal{J}_{\text{Swap}} \cap \mathbb{C}\langle \text{Swap} \rangle_{2d}) = \{0\} , \]

where \( \Gamma_h \) is a (not necessarily positive semidefinite) Gram matrix for \( h \),

\[ h = V_d(n)^* \Gamma_h V_d(n) . \]
Takahashi, Rayudu, Zhou, King, Thompson, Parekh (2023) give many examples of the 1st nc Lasserre hierarchy.

- It is **exact** for
  - star graphs
  - even cliques
  - certain crown graphs

- It is **non-exact** for odd cliques, and many small ($n \leq 6$) graphs.
The second nc Moment-SOS relaxation for QMC has size

\[ 1 + \binom{n}{2} + \binom{n}{3} + 3\binom{n}{4} = \frac{1}{24} \left( 3n^4 - 14n^3 + 33n^2 - 22n + 24 \right) \]

<table>
<thead>
<tr>
<th>n</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>12</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>14</td>
<td>36</td>
<td>81</td>
<td>162</td>
<td>295</td>
<td>499</td>
<td>796</td>
<td>1772</td>
<td>4656</td>
<td>15866</td>
</tr>
</tbody>
</table>
The second nc Moment-SOS relaxation for QMC has size

$$1 + \binom{n}{2} + \binom{n}{3} + 3 \binom{n}{4} = \frac{1}{24} (3n^4 - 14n^3 + 33n^2 - 22n + 24)$$

**Proposition**

For $n \leq 8$ the second nc Moment-SOS relaxation for QMC of an $n$ vertex QMC with uniform edge weights is up to the tolerance of $10^{-7}$ exact, i.e., equal to the true max.

Uses nc Gröbner bases.
The second nc Moment-SOS relaxation for QMC has size

\[ 1 + \binom{n}{2} + \binom{n}{3} + 3 \binom{n}{4} = \frac{1}{24} (3n^4 - 14n^3 + 33n^2 - 22n + 24) \]

**Proposition**

For \( n \leq 8 \) the second nc Moment-SOS relaxation for QMC of an \( n \) vertex QMC with uniform edge weights is up to the tolerance of \( 10^{-7} \) exact, i.e., equal to the true max.

Uses nc Gröbner bases.

? It would be interesting to find the smallest graph on which the second relaxation is not exact.

! It *appears* that the first classical relaxation is worse than the quantum one for swaps.
QMC

Rounding

\[ \text{eig}_{\text{max}}(H) = \langle Hv, v \rangle = \text{tr}(Hvv^T), \quad \rho \text{ is a state} \]

- Round SDP solutions to **product states** \( \rho = \rho_1 \otimes \cdots \otimes \rho_n \)
  
  Brandao & Harrow (2016), Bravyi & Gosset & König & Temme (2019), Gharibian & Parekh (2019), Parekh & Thompson (2021);

- Parekh & Thompson (2022): “optimal” rounding to product state = 1/2−approximation;

- Anshu & Gosset & Morenz (2020): 0.531−approximation;

- Parekh & Thompson (2021): 0.533−approximation;

- King (2023): 0.582−approximation;

- Hwang & Neeman & Parekh & Thompson & Wright (2023): Unique Games hardness of \((0.956 + \varepsilon)\)−approximation for QMC, assuming a plausible conjecture in Gaussian geometry;

- depends heavily on the \(k\)-rank rounding of Briët & de Oliveira Filho & Vallentin (2010)
Algorithm 1 PT2021 Approximation Algorithm for QMC

1. Input graph $G = (V, E)$ with weights $w = \{w_e \geq 0\}_{e \in E}$, solve 1st nc Lasserre. Let the matrix $\mathcal{M}$ be an optimal solution.

2. For each $(i, j) \in E$ calculate $x_{ij} := \frac{1 - 2\mathcal{M}(\text{Swap}_{ij}, 1)}{3}$.

3. Pick $d \in \mathbb{N}$, and define $L := \{e \in E \mid x_e > \alpha(d) := \frac{d+3}{3(d+1)}\}$. Find a maximum-weight matching $F$ in the graph $G_L := (V, L)$ w.r.t weights $\{w_e\}_{e \in L}$. Let $U$ be the vertices unmatched by $F$.

4. Define a quantum state:
   $$\rho_F := \prod_{ij \in F} \left(\frac{I - \text{Swap}_{ij}}{2}\right) \prod_{v \in U} I_2 \cdot (1)$$

5. Find the optimal product state $\rho_{PS}$.

6. Output the better of $\rho_F$ and $\rho_{PS}$. 
QMC

Exact solutions – clique

\[ H_G = \sum_{(i,j) \in E(G)} 2(I - \text{Swap}_{ij}), \quad M_n^\text{Swap} \cong \bigoplus_{k=0}^{\lfloor \frac{n}{2} \rfloor} M_{\text{dim} \mathcal{S}_{[n-k,k]}(\mathbb{C})} \]

Example

Let \( G = K_n \) be the clique on \( n \) vertices. Then

\[ H_{K_n} = 2 \sum_{i < j} (I - \text{Swap}_{ij}). \]

- Under each irrep \( \lambda \), \( H_{K_n}^\lambda \) is a scalar matrix.
Exact solutions – clique

\[ H_G = \sum_{(i,j) \in E(G)} 2(I - \text{Swap}_{ij}), \quad M_n^{\text{Swap}} \cong \bigoplus_{k=0}^{\left\lfloor \frac{n}{2} \right\rfloor} M_{\dim \mathcal{S}_{n-k,k}}(\mathbb{C}) \]

**Example**

Let \( G = K_n \) be the clique on \( n \) vertices. Then

\[ H_{K_n} = 2 \sum_{i<j} (I - \text{Swap}_{ij}). \]

- Under each irrep \( \lambda \), \( H_{K_n}^{\lambda} \) is a scalar matrix.
- \( H_{K_n}^{[n-k,k]} = \binom{n}{2} + k^2 - k(n + 1) \) (hook length & Murnaghan-Nakayama rule).
QMC

Exact solutions – clique

\[ H_G = \sum_{(i,j) \in E(G)} 2(I - \text{Swap}_{ij}), \quad M_n^{\text{swap}} \cong \bigoplus_{k=0}^{\lfloor \frac{n}{2} \rfloor} M_{\dim S_{[n-k,k]}}(\mathbb{C}) \]

Example

Let \( G = K_n \) be the clique on \( n \) vertices. Then

\[ H_{K_n} = 2 \sum_{i<j} (I - \text{Swap}_{ij}). \]

- Under each irrep \( \lambda \), \( H_{K_n}^\lambda \) is a scalar matrix.
- \( H_{K_n}^{[n-k,k]} = \binom{n}{2} + k^2 - k(n+1) \) (hook length & Murnaghan-Nakayama rule).
- QMC value of \( K_n \) is the max of \( H_{K_n}^{[n-k,k]} \) for \( k = 0, \ldots, \lfloor \frac{n}{2} \rfloor \),
  and is attained at \( k = \lfloor \frac{n}{2} \rfloor \).

\[ \begin{array}{c}
\hline
1 \\
\hline
7 \hline
\hline
2 \hline
\hline
6 \hline
\hline
5 \hline
\hline
4
\end{array} \quad \begin{array}{c}
\hline
\hline
\hline
\hline
\hline
\hline
\hline
\end{array} \quad \begin{array}{c}
\hline
\hline
\hline
\hline
\hline
\end{array} \quad \hbox{or} \quad \begin{array}{c}
\hline
\hline
\hline
\hline
\hline
\hline
\hline
\end{array} \]
QMC

Exact solutions – clique

\[ H_G = \sum_{(i,j) \in E(G)} 2(I - \text{Swap}_{ij}), \quad M_{\text{Swap}}^n = \bigoplus_{k=0}^{\lfloor \frac{n}{2} \rfloor} M_{\text{dim}} \mathcal{S}_{[n-k,k]}(\mathbb{C}) \]

Example

Let \( G = K_n \) be the clique on \( n \) vertices. Then

\[ H_{K_n} = 2 \sum_{i<j} (I - \text{Swap}_{ij}). \]

- Under each irrep \( \lambda \), \( H_{K_n}^\lambda \) is a scalar matrix.
- \( H_{K_n}^{[n-k,k]} = \binom{n}{2} + k^2 - k(n + 1) \) (hook length & Murnaghan-Nakayama rule).

This allows us to write an nc Moment-SOS relaxation scheme for optimizing \( H_G^\lambda \) inside a two row irrep \( \lambda \).
Exact solutions – star graph

\[ H_G = \sum_{(i,j) \in E(G)} 2(I - \text{Swap}_{ij}), \quad M_{n}^{\text{Swap}} \cong \bigoplus_{k=0}^{\lfloor \frac{n}{2} \rfloor} M_{\text{dim} \mathcal{H}_{[n-k,k]}(\mathbb{C})} \]

Example

Let \( G = \star_n \) be the star graph on \( n \) vertices. Then

\[
H_{\star_n} = 2 \sum_{j<n} (I - \text{Swap}_{jn}).
\]

- \( H_{\star_n}^{[n-k,k]} \) has two eigenvalues, namely \( 2(n - k + 1) > 2k \).
**QMC**

**Exact solutions – star graph**

\[ H_G = \sum_{(i,j) \in E(G)} 2(I - \text{Swap}_{ij}), \quad M_n^{\text{Swap}} \cong \bigoplus_{k=0}^{\lfloor \frac{n}{2} \rfloor} M_{\dim \mathcal{S}_{[n-k,k]}(\mathbb{C})} \]

**Example**

Let \( G = \star_n \) be the star graph on \( n \) vertices. Then

\[ H_{\star_n} = 2 \sum_{j<n} (I - \text{Swap}_{jn}). \]

- \( H_{\star_n}^{[n-k,k]} \) has two eigenvalues, namely \( 2(n - k + 1) > 2k \).
Exact solutions – star graph (cont’d)

\[ \star_n = K_n - K_{n-1} \]

\[
H_{\star_n}^{[n-k,k]} = H_{K_n}^{[n-k,k]} - H_{K_{n-1}}^{[n-k,k]}.
\]
Exact solutions – star graph (cont’d)

\[ \star_n = K_n - K_{n-1} \]

\[ H^{[n-k,k]}_{\star_n} = H^{[n-k,k]}_{K_n} - H^{[n-k,k]}_{K_{n-1}}. \]

Branching rule:
Exact solutions – star graph (cont’d)

\( \star_n = K_n - K_{n-1} \)

\[
H_{\star_n}^{[n-k,k]} = H_{K_n}^{[n-k,k]} - H_{K_{n-1}}^{[n-k,k]}.
\]

Branching rule:

\[
H_{\star_n}^{[n-k,k]} = H_{K_n}^{[n-k,k]} - H_{K_{n-1}}^{[n-k,k]} = H_{K_n}^{[n-k,k-1]} + H_{K_{n-1}}^{[n-k-1,k]}
\]
For any connected graph $G$, the tree clique decomposition of $G$, denoted $T(G)$, consists of a rooted tree $T = \{v_1, \ldots, v_m\}$, and connected graphs $\{G(v_1) = G, \ldots, G(v_m)\}$ such that:

- For any vertex $v_i$ of $T$ which is not a leaf vertex, let $c_1, \ldots, c_k$ be its children. Then
  \[
  G(v_i)^c = \bigcup_{j \in \{1,2,\ldots,k\}} G(c_j),
  \]

- For any leaf vertex $v_j$ of $T$ we have that $G(v_j)^c$ is connected or $G(v_j)^c$ is totally disconnected.
QMC

Tree-clique decomposition – example

Graph $G$ and its tree-clique decomposition: $G(v_1), G(v_2), G(v_3), G(v_4), G(v_5), G(v_6), G(v_7), G(v_8), G(v_9)$.
Tree-clique decomposition – example
Theorem
Let $G$ be a graph and $\mathcal{T}(G) = \{T, \{G(v_1), \ldots, G(v_m)\}\}$ be its tree-clique decomposition. Then

- For any vertex $v \in T$ with children $c_1, \ldots, c_k$.
  \[
  H_{G(v)} = H_{K(G(v))} - \sum_{j \in \{1, \ldots, k\}} H_{G(c_j)}
  \]
- Let $L$ denote the set of leaf vertices in $T$, and $R$ be all non-leaf vertices. Let $d(v)$ denote the depth of vertex $v$ in the tree, with root $d(v_1) = 0$. Then
  \[
  H_G = \sum_{r \in R} (-1)^{d(r)} H_{K(G(r))} + \sum_{l \in L} (-1)^{d(l)} H_{G(l)}
  \]
Theorem
Let $G$ be a graph and $\mathcal{T}(G) = \{T, \{G(v_1), \ldots, G(v_m)\}\}$ be its tree-clique decomposition. Then

1. For any vertex $v \in T$ with children $c_1, \ldots, c_k$.

   $$H_{G(v)} = H_{K(G(v))} - \sum_{j \in \{1, \ldots, k\}} H_{G(c_j)}$$

2. Let $L$ denote the set of leaf vertices in $T$, and $R$ be all non-leaf vertices. Let $d(v)$ denote the depth of vertex $v$ in the tree, with root $d(v_1) = 0$. Then

   $$H_G = \sum_{r \in R} (-1)^{d(r)} H_{K(G(r))} + \sum_{l \in L} (-1)^{d(l)} H_{G(l)}$$

Given min and max eigenvalues under all two row irreps of $G(l)$ for every leaf vertex $l$ of $T$, one can inductively compute min and max eigenvalues under all two row irreps of $G$. 

QMC
Tree-clique decomposition and QMC
Quantum Max Cut (QMC) is fun 😊

✓ QMC Hamiltonian expressed in terms of SWAP operators
  Identify the SWAP algebra via Schur-Weyl duality

✓ Noncommutative Lasserre’s relaxation produces an SDP hierarchy for QMC

✓ Exact solutions for various simple graphs

✓ Tree-clique decomposition algorithm for writing a graph as a sum of ±cliques
  Yields a recursive algorithm for solving QMC exactly

✓ $\mathbb{C}^2 \mapsto \mathbb{C}^d$: qudits instead of qubits

¿ Any ideas/thoughts for a better rounding algorithm?