SDP AND EIGENVALUE BOUNDS FOR THE GRAPH PARTITION PROBLEM

Renata Sotirov and Edwin van Dam

Tilburg University, The Netherlands

Outline ...

THE GRAPH PARTITION PROBLEM

Outline . . .

THE GRAPH PARTITION PROBLEM

↙ ↘

matrix lifting   vector lifting

Outline . . .

THE GRAPH PARTITION PROBLEM

↙ ↘

matrix lifting   vector lifting

↙   ↘

simplify   complicate

Outline . . .

THE GRAPH PARTITION PROBLEM

$\swarrow \quad \searrow$

matrix lifting   vector lifting

$\swarrow \qquad\qquad \searrow$

simplify   complicate

$\Downarrow \qquad\qquad \Downarrow$

???

The Graph Partition Problem

- $G = (V, E)$ ...an undirected graph
  - V ...vertex set, $|V| = n$
  - E ...edge set

The Graph Partition Problem

- $G = (V, E)$ . . . an undirected graph
  - V . . . vertex set, $|V| = n$
  - E . . . edge set

The k-partition problem (GPP)

Find a partition of V into k subsets $S_1, \ldots, S_k$ of given sizes $m_1 \geq \ldots \geq m_k$, s.t.
the total weight of edges joining different S_i is minimized.

The Graph Partition Problem

- $G = (V, E)$ ...an undirected graph
  - V ...vertex set, $|V| = n$
  - E ...edge set

The k-partition problem (GPP)

Find a partition of V into k subsets $S_1, \ldots, S_k$ of given sizes $m_1 \geq \ldots \geq m_k$, s.t.
the total weight of edges joining different S_i is minimized.

- $m_i = \frac{|V|}{k}, \forall i \rightsquigarrow$ the **graph equipartition problem**
- $k = 2 \rightsquigarrow$ the **bisection problem**

The *k*-partition problem

- A . . . the adjacency matrix of G, $m := (m_1, \ldots, m_k)^{\mathrm{T}}$

The k-partition problem

- A ...the adjacency matrix of G, $m := (m_1, \ldots, m_k)^{\mathrm{T}}$
- let $X = (x_{ij}) \in \mathbb{R}^{|V| \times k}$

$$x_{ij} := \begin{cases} 1, & \text{if vertex } i \in S_j \\ 0, & \text{if vertex } i \notin S_j \end{cases}$$

The k-partition problem

- A ...the adjacency matrix of G, $m := (m_1, \dots, m_k)^{\mathrm{T}}$
- let $X = (x_{ij}) \in \mathbb{R}^{|V| \times k}$

$$x_{ij} := \begin{cases} 1, & \text{if vertex } i \in S_j \\ 0, & \text{if vertex } i \notin S_j \end{cases}$$

- $\mathcal{P}_k := \left\{ X \in \mathbb{R}^{n \times k} : \; Xu_k = u_n, \; X^{\mathrm{T}} u_n = m, \; x_{ij} \in \{0,1\} \right\}$, where u_n ...vector of all ones

The *k*-partition problem

- A ...the adjacency matrix of G, $m := (m_1, \ldots, m_k)^{\mathrm{T}}$
- let $X = (x_{ij}) \in \mathbb{R}^{|V| \times k}$

$$x_{ij} := \begin{cases} 1, & \text{if vertex } i \in S_j \\ 0, & \text{if vertex } i \notin S_j \end{cases}$$

- $\mathcal{P}_k := \left\{ X \in \mathbb{R}^{n \times k} : \; Xu_k = u_n, \; X^{\mathrm{T}} u_n = m, \; x_{ij} \in \{0,1\} \right\}$, where u_n ...vector of all ones

For $X \in \mathcal{P}_k$:

$$\mathbf{w}(\mathbf{E}_{\mathrm{cut}}) = \frac{1}{2} \operatorname{tr}(X^{\mathrm{T}} L X) = \frac{1}{2} \operatorname{tr} A(J_n - XX^{\mathrm{T}})$$

where $L := \mathrm{Diag}(Au_n) - A$ is the Laplacian matrix of G

The Graph Partition Problem

THE TRACE FORMULATION:

$$
(\text{GPP}) \quad \begin{aligned} \min \quad & \tfrac{1}{2}\mathrm{trace}(X^{\mathrm{T}} L X) \\ \text{s.t.} \quad & X u_k = u_n \\ & X^{\mathrm{T}} u_n = m \\ & x_{ij} \in \{0, 1\} \end{aligned}
$$

The Graph Partition Problem

THE TRACE FORMULATION:

$$
\text{(GPP)} \quad \begin{array}{ll} \min & \frac{1}{2}\mathrm{trace}(X^{\mathrm{T}} L X) \\ \text{s.t.} & X u_k = u_n \\ & X^{\mathrm{T}} u_n = m \\ & x_{ij} \in \{0,1\} \end{array}
$$

the GPP ...

- is NP-hard (Garey and Johnson, 1976)

The Graph Partition Problem

THE TRACE FORMULATION:

$$
\text{(GPP)} \quad \begin{array}{ll} \min & \frac{1}{2}\mathrm{trace}(X^{\mathrm{T}} L X) \\ \text{s.t.} & X u_k = u_n \\ & X^{\mathrm{T}} u_n = m \\ & x_{ij} \in \{0, 1\} \end{array}
$$

the GPP . . .

- is NP-hard (Garey and Johnson, 1976)
- **applications**: VLSI design, parallel computing, floor planning, telecommunications, etc.

matrix lifting SDP for the GPP . . .

SDP for GPP

- linearize the objective: $\mathrm{trace}(L\mathbf{X}\mathbf{X}^{\mathrm{T}}) \rightsquigarrow \mathrm{trace}(L\mathbf{Y})$

SDP for GPP

- linearize the objective: $\mathrm{trace}(\boldsymbol{L}\mathbf{X}\mathbf{X}^{\mathrm{T}}) \rightsquigarrow \mathrm{trace}(\boldsymbol{L}\mathbf{Y})$

$$Y \in \mathrm{conv}\{\tilde{Y} : \exists X \in \mathcal{P}_k \text{ s.t. } \tilde{Y} = XX^{\mathrm{T}}\} \Rightarrow kY - J_n \succeq 0.$$

SDP for GPP

- linearize the objective: $\mathrm{trace}(L\mathbf{X}\mathbf{X}^{\mathrm{T}}) \rightsquigarrow \mathrm{trace}(L\mathbf{Y})$

$$Y \in \mathrm{conv}\{\tilde{Y} : \exists X \in \mathcal{P}_k \text{ s.t. } \tilde{Y} = XX^{\mathrm{T}}\} \Rightarrow kY - J_n \succeq 0.$$

S., 2013

$$(\mathrm{GPP_m}) \quad \begin{array}{rl} \min & \frac{1}{2}\mathrm{tr}(LY) \\ \mathrm{s.t.} & \mathrm{diag}(Y) = u_n \\ & \mathrm{tr}(JY) = \sum_{i=1}^{k} m_i^2 \\ & kY - J_n \succeq 0, \ \ Y \geq 0 \end{array}$$

SDP for GPP

- linearize the objective: $\mathrm{trace}(\mathbf{L}\mathbf{X}\mathbf{X}^{\mathrm{T}}) \rightsquigarrow \mathrm{trace}(\mathbf{L}\mathbf{Y})$

$$Y \in \mathrm{conv}\{\tilde{Y} : \exists X \in \mathcal{P}_k \text{ s.t. } \tilde{Y} = XX^{\mathrm{T}}\} \Rightarrow kY - J_n \succeq 0.$$

S., 2013

$$(\mathrm{GPP_m}) \quad \begin{array}{ll} \min & \frac{1}{2}\mathrm{tr}(LY) \\ \text{s.t.} & \mathrm{diag}(Y) = u_n \\ & \mathrm{tr}(JY) = \sum_{i=1}^{k} m_i^2 \\ & kY - J_n \succeq 0, \;\; Y \geq 0 \end{array}$$

- for $k = 2$ the **nonnegativity** constraints are **redundant**

GPP_{m} and known relaxations

GPP_{m} ...

- for **equipartition** is equivalent to the relaxation from:

  S.E. Karisch, F. Rendl. Semidefnite programming and graph equipartition. In: *Topics in Semidefinite and Interior–Point Methods*, The Fields Institute for research in Math. Sc., Comm. Ser. Rhode Island, 18, 1998.

GPP_m and known relaxations

GPP_m . . .

- for **equipartition** is equivalent to the relaxation from:

  S.E. Karisch, F. Rendl. Semidefnite programming and graph equipartition. In: *Topics in Semidefinite and Interior–Point Methods*, The Fields Institute for research in Math. Sc., Comm. Ser. Rhode Island, 18, 1998.

- for **bisection** is equivalent to the relaxation from:

  S. E. Karisch, F. Rendl, J. Clausen. Solving graph bisection problems with semidefinite programming, *INFORMS J. Comput.*, 12:177-191, 2000.

Strengthening?

Strengthening?

? How to strengthen $\mathrm{GPP_m}$?

Strengthening?

? HOW TO STRENGTHEN GPP_{m}?

impose the linear inequalities:

- Δ constraints

$$y_{ab} + y_{ac} \leq 1 + y_{bc}, \quad \forall (a, b, c)$$

- independent set constraints

$$\sum_{a<b,\ a,b \in \mathcal{W}} y_{ab} \geq 1, \quad \forall\ \mathcal{W} \text{ s.t. } |\mathcal{W}| = k+1$$

Strengthening?

? HOW TO STRENGTHEN $\mathrm{GPP_m}$?

impose the linear inequalities:

- Δ constraints

$$y_{ab} + y_{ac} \leq 1 + y_{bc}, \quad \forall (a, b, c)$$

- independent set constraints

$$\sum_{a<b,\ a,b \in \mathcal{W}} y_{ab} \geq 1, \quad \forall\ \mathcal{W} \text{ s.t. } |\mathcal{W}| = k+1$$

$\Rightarrow$ there are $3\binom{n}{3}$ Δ, and $\binom{n}{k+1}$ independent set constraints

On computational issues ...

in general, for graphs with 100 vertices and $k = 3$:

- the best known vector lifting relaxation is **hopeless**

On computational issues . . .

in general, for graphs with 100 vertices and $k = 3$:

- the best known vector lifting relaxation is **hopeless**
- $\mathrm{GPP_m}$ + triangle inequalities + independent set solves $\sim$ **3 h**

On computational issues . . .

in general, for graphs with 100 vertices and $k = 3$:

- the best known vector lifting relaxation is **hopeless**
- $\mathrm{GPP}_{\mathrm{m}}$ + triangle inequalities + independent set solves ∼ **3 h**
- $\mathrm{GPP}_{\mathrm{m}}$ solves ∼ **14 min**

On computational issues . . .

in general, for graphs with 100 vertices and $k = 3$:

- the best known vector lifting relaxation is **hopeless**
- $\mathrm{GPP}_{\mathrm{m}}$ + triangle inequalities + independent set solves $\sim$ **3 h**
- $\mathrm{GPP}_{\mathrm{m}}$ solves $\sim$ **14 min**

CAN WE COMPUTE $\mathrm{GPP}_{\mathrm{m}}$ WITH/WITHOUT ADDITIONAL CONSTR. MORE EFFICIENTLY ?

On computational issues . . .

in general, for graphs with 100 vertices and $k = 3$:

- the best known vector lifting relaxation is **hopeless**
- GPP_m + triangle inequalities + independent set solves $\sim$ **3 h**
- GPP_m solves $\sim$ **14 min**

CAN WE COMPUTE GPP_m WITH/WITHOUT ADDITIONAL CONSTR. MORE EFFICIENTLY ?

YES

Simplification – ‘highly symmetric’ graphs . . .

Simplification – ‘highly symmetric’ graphs . . .

- matrix *-algebra: subspace of $\mathbb{R}^{n\times n}$ that is closed under matrix multiplication and taking transposes

Simplification – ‘highly symmetric’ graphs . . .

- matrix *-algebra: subspace of $\mathbb{R}^{n\times n}$ that is closed under matrix multiplication and taking transposes

Assumption: The data matrices of an SDP problem and I belong to a matrix *-algebra $\mathrm{span}\{A_1, \ldots, A_r\}$ where $r \ll n^2$

Simplification – 'highly symmetric' graphs . . .

- matrix *-algebra: subspace of $\mathbb{R}^{n\times n}$ that is closed under matrix multiplication and taking transposes

Assumption: The data matrices of an SDP problem and I belong to a matrix *-algebra span$\{A_1, \ldots, A_r\}$ where $r \ll n^2$

Then, if the SDP relaxation has an optimal solution
$\Rightarrow$ it has an optimal solution in the matrix *-algebra.

Simplification – 'highly symmetric' graphs . . .

- matrix *-algebra: subspace of $\mathbb{R}^{n \times n}$ that is closed under matrix multiplication and taking transposes

Assumption: The data matrices of an SDP problem and I belong to a matrix *-algebra $\text{span}\{A_1, \ldots, A_r\}$ where $r \ll n^2$

Then, if the SDP relaxation has an optimal solution
$\Rightarrow$ it has an optimal solution in the matrix *-algebra.

Schrijver, Goemans, Rendl, Parrilo, . . .

Simplification – 'highly symmetric' graphs . . .

- matrix *-algebra: subspace of $\mathbb{R}^{n\times n}$ that is closed under matrix multiplication and taking transposes

Assumption: The data matrices of an SDP problem and I belong to a matrix *-algebra $\text{span}\{A_1,\dots,A_r\}$ where $r \ll n^2$

Then, if the SDP relaxation has an optimal solution
$\Rightarrow$ it has an optimal solution in the matrix *-algebra.

Schrijver, Goemans, Rendl, Parrilo, . . .

- a basis of the matrix *-algebra
(coming from combinatorial or group symmetry):

(i) $A_i \in \{0,1\}^{n\times n}$, $A_i^{\mathrm{T}} \in \{A_1,\dots,A_r\}$, $(i = 1,\dots,r)$

(ii) $\sum_{i=1}^r A_i = J$, $\sum_{i\in\mathcal{I}} A_i = I$, $\mathcal{I} \subset \{1,\dots,r\}$

(iii) For $i,j \in \{1,\dots,r\}$, $\exists p_{ij}^h$ such that $A_iA_j = \sum_{h=1}^r p_{ij}^h A_h$.

Simplification – 'highly symmetric' graphs ...

$$\Rightarrow Y = \sum_{i=1}^{r} \mathbf{z_i} A_i, \quad \mathbf{z_i} \in \mathbb{R} \ (r \ll n^2)$$

$$(\mathrm{GPP_m}) \quad \begin{array}{ll} \min & \frac{1}{2}\operatorname{tr}(AJ_n) - \frac{1}{2}\sum_{i=1}^{r} \mathbf{z_i} \operatorname{tr}(AA_i) \\ \text{s.t.} & \sum_{i \in \mathcal{I}} \mathbf{z_i} \operatorname{diag}(A_i) = u_n \\ & \sum_{i=1}^{r} \mathbf{z_i} \operatorname{tr}(JA_i) = \sum_{i=1}^{k} m_i^2 \\ & k\sum_{j=1}^{r} \mathbf{z_i} A_i - J_n \succeq 0, \quad \mathbf{z_i} \geq 0, \quad i = 1, \ldots, r. \end{array}$$

Simplification – 'highly symmetric' graphs ...

$$\Rightarrow Y = \sum_{i=1}^{r} \mathbf{z_i} A_i, \quad \mathbf{z_i} \in \mathbb{R} \; (r \ll n^2)$$

$$
(\mathrm{GPP_m}) \quad
\begin{array}{ll}
\min & \frac{1}{2}\operatorname{tr}(AJ_n) - \frac{1}{2}\sum_{i=1}^{r} \mathbf{z_i} \operatorname{tr}(AA_i) \\
\text{s.t.} & \sum_{i \in \mathcal{I}} \mathbf{z_i} \operatorname{diag}(A_i) = u_n \\
& \sum_{i=1}^{r} \mathbf{z_i} \operatorname{tr}(JA_i) = \sum_{i=1}^{k} m_i^2 \\
& k \sum_{j=1}^{r} \mathbf{z_i} A_i - J_n \succeq 0, \quad \mathbf{z_i} \geq 0, \quad i = 1, \ldots, r.
\end{array}
$$

- LMI may be (block-)diagonalized

Simplification – 'highly symmetric' graphs ...

$$\Rightarrow Y = \sum_{i=1}^{r} \mathbf{z_i} A_i, \quad \mathbf{z_i} \in \mathbb{R} \ (r \ll n^2)$$

$$(\mathrm{GPP_m}) \quad \begin{array}{ll} \min & \frac{1}{2}\operatorname{tr}(AJ_n) - \frac{1}{2}\sum_{i=1}^{r} \mathbf{z_i} \operatorname{tr}(AA_i) \\ \text{s.t.} & \sum_{i \in \mathcal{I}} \mathbf{z_i} \operatorname{diag}(A_i) = u_n \\ & \sum_{i=1}^{r} \mathbf{z_i} \operatorname{tr}(JA_i) = \sum_{i=1}^{k} m_i^2 \\ & k\sum_{j=1}^{r} \mathbf{z_i} A_i - J_n \succeq 0, \quad \mathbf{z_i} \geq 0, \quad i = 1, \ldots, r. \end{array}$$

- LMI may be (block-)diagonalized
- exploit properties of A_i to aggregate Δ and indep. set const.

  $\Rightarrow$ extend the approach from:
  M.X. Goemans, F. Rendl. Semidefinite Programs and Association Schemes. *Computing*, 63(4):331–340, 1999.

On aggregating constraints ...

- for a given (a, b, c) consider the Δ inequality

$$y_{ab} + y_{ac} \leq 1 + y_{bc}$$

On aggregating constraints . . .

- for a given (a, b, c) consider the Δ inequality

$$y_{ab} + y_{ac} \leq 1 + y_{bc}$$

- if $(A_i)_{ab} = 1$, $(A_h)_{ac} = 1$, $(A_j)_{bc} = 1 \Rightarrow$ type (i, j, h) ineq.,

On aggregating constraints . . .

- for a given (a, b, c) consider the Δ inequality

$$y_{ab} + y_{ac} \leq 1 + y_{bc}$$

- if $(A_i)_{ab} = 1$, $(A_h)_{ac} = 1$, $(A_j)_{bc} = 1 \Rightarrow$ type (i, j, h) ineq.,

- by summing all ineq. of type (i, j, h), the aggregated Δ ineq.:

$$p^i_{hj'} \operatorname{tr} A_i Y + p^h_{ij} \operatorname{tr} A_h Y \leq p^i_{hj'} \operatorname{tr} A_i J + p^j_{i'h} \operatorname{tr} A_j Y,$$

where p^h_{ij}: $A_i A_j = \sum_{h=1}^r p^h_{ij} A_h$

j': is the index s.t. $A_{j'} = A_j^{\mathrm{T}}$

On aggregating constraints . . .

- for a given (a, b, c) consider the Δ inequality

$$y_{ab} + y_{ac} \leq 1 + y_{bc}$$

- if $(A_i)_{ab} = 1$, $(A_h)_{ac} = 1$, $(A_j)_{bc} = 1 \Rightarrow$ type (i, j, h) ineq.,

- by summing all ineq. of type (i, j, h), the aggregated Δ ineq.:

$$p^i_{hj'} \operatorname{tr} A_i Y + p^h_{ij} \operatorname{tr} A_h Y \leq p^i_{hj'} \operatorname{tr} A_i J + p^j_{i'h} \operatorname{tr} A_j Y,$$

  where p^h_{ij}: $A_i A_j = \sum_{h=1}^{r} p^h_{ij} A_h$

  j': is the index s.t. $A_{j'} = A_j^{\mathrm{T}}$

- use: $Y = \sum_{j=1}^{r} z_j A_j$

On aggregating constraints ...

- for a given (a, b, c) consider the Δ inequality

$$y_{ab} + y_{ac} \leq 1 + y_{bc}$$

- if $(A_i)_{ab} = 1$, $(A_h)_{ac} = 1$, $(A_j)_{bc} = 1 \Rightarrow$ type (i, j, h) ineq.,

- by summing all ineq. of type (i, j, h), the aggregated Δ ineq.:

$$p^i_{hj'} \operatorname{tr} A_i Y + p^h_{ij} \operatorname{tr} A_h Y \leq p^i_{hj'} \operatorname{tr} A_i J + p^j_{i'h} \operatorname{tr} A_j Y,$$

where p^h_{ij}: $A_i A_j = \sum_{h=1}^r p^h_{ij} A_h$

j': is the index s.t. $A_{j'} = A_j^{\mathrm{T}}$

- use: $Y = \sum_{j=1}^r z_j A_j$

- $\sharp$ of aggregated Δ constraints is bounded by r^3

On aggregating constraints . . .

- for a given (a, b, c) consider the Δ inequality

$$y_{ab} + y_{ac} \leq 1 + y_{bc}$$

- if $(A_i)_{ab} = 1$, $(A_h)_{ac} = 1$, $(A_j)_{bc} = 1 \Rightarrow$ type (i, j, h) ineq.,

- by summing all ineq. of type (i, j, h), the aggregated Δ ineq.:

$$p^i_{hj'} \operatorname{tr} A_i Y + p^h_{ij} \operatorname{tr} A_h Y \leq p^i_{hj'} \operatorname{tr} A_i J + p^j_{i'h} \operatorname{tr} A_j Y,$$

where p^h_{ij}: $A_i A_j = \sum_{h=1}^r p^h_{ij} A_h$

j': is the index s.t. $A_{j'} = A_j^{\mathrm{T}}$

- use: $Y = \sum_{j=1}^r z_j A_j$

- $\sharp$ of aggregated Δ constraints is bounded by r^3
- similar approach applies to independent set constr. when $k = 2$

Simplification – 'highly symmetric' graphs . . .

Example. Strongly regular graph (SRG)

Simplification – 'highly symmetric' graphs . . .

Example. Strongly regular graph (SRG)

- n vertices, κ the *valency* of the graph

Simplification – 'highly symmetric' graphs . . .

Example. Strongly regular graph (SRG)

- n vertices, κ the *valency* of the graph
- A has exactly two eigenvalues $r \geq 0$ and $s < 0$ associated with eigenvectors $\perp u_n$

Simplification – 'highly symmetric' graphs . . .

Example. Strongly regular graph (SRG)

- n vertices, κ the *valency* of the graph
- A has exactly two eigenvalues $r \geq 0$ and $s < 0$ associated with eigenvectors $\perp u_n$
- A belongs to the *-algebra spanned by $\{I, A, J - A - I\}$

Simplification – 'highly symmetric' graphs . . .

Example. Strongly regular graph (SRG)

- n vertices, κ the *valency* of the graph
- A has exactly two eigenvalues $r \geq 0$ and $s < 0$ associated with eigenvectors $\perp u_n$
- A belongs to the *-algebra spanned by $\{I, A, J - A - I\}$

$$\Rightarrow Y = I + z_1 A + z_2 (J - A - I)$$

Simplification – 'highly symmetric' graphs ...

Example. Strongly regular graph (SRG)

- n vertices, κ the *valency* of the graph
- A has exactly two eigenvalues $r \geq 0$ and $s < 0$ associated with eigenvectors $\perp u_n$
- A belongs to the *-algebra spanned by $\{I, A, J - A - I\}$

$$\Rightarrow Y = I + z_1 A + z_2 (J - A - I)$$

$$(\mathrm{GPP_m}) \quad \begin{array}{ll} \min & \frac{1}{2}\kappa n(1 - z_1) \\ \text{s.t.} & \kappa z_1 + (n - \kappa - 1) z_2 = \frac{1}{n}\sum_{i=1}^{k} m_i^2 - 1 \\ & 1 + r z_1 - (r + 1) z_2 \geq 0 \\ & 1 + s z_1 - (s + 1) z_2 \geq 0 \\ & z_1, z_2 \geq 0 \end{array}$$

SRG

THEOREM.
Let $G = (V, E)$ be a SRG with eigenvalues κ, r, s.
Let $m_i \in \mathbf{N}$, $i = 1, \ldots, k$ s.t. $\sum_{j=1}^{k} m_j = n$.

Then the SDP bound for the minimum k-partition is

$$\max\left\{\frac{\kappa - r}{n}\sum_{i<j} m_i m_j,\ \frac{1}{2}\left(n(\kappa+1) - \sum_i m_i^2\right)\right\}$$

Similarly, the SDP bound for the maximum k-partition is

$$\min\left\{\frac{\kappa - s}{n}\sum_{i<j} m_i m_j,\ \frac{1}{2}\kappa n\right\}.$$

SRG

THEOREM.
Let $G = (V, E)$ be a SRG with eigenvalues κ, r, s.
Let $m_i \in \mathbf{N}$, $i = 1, \ldots, k$ s.t. $\sum_{j=1}^{k} m_j = n$.

Then the SDP bound for the minimum k-partition is

$$\max\left\{\tfrac{\kappa-r}{n}\sum_{i<j} m_i m_j,\ \tfrac{1}{2}\left(n(\kappa+1)-\sum_i m_i^2\right)\right\}$$

Similarly, the SDP bound for the maximum k-partition is

$$\min\left\{\tfrac{\kappa-s}{n}\sum_{i<j} m_i m_j,\ \tfrac{1}{2}\kappa n\right\}.$$

- this is an extension of the result for the equipartition:
  De Klerk, Pasechnik, S., Dobre: On SDP relaxations of maximum k-section, *Math. Program. Ser. B*, 136(2):253-278, 2012.

SRG

- after aggregating, $3\binom{n}{3}$ Δ constraints remain:

$$\begin{array}{rcl} z_1 & \leq & 1 \\ z_2 & \leq & 1 \\ 2z_1 - z_2 & \leq & 1 \\ -z_1 + 2z_2 & \leq & 1 \end{array}$$

SRG

- after aggregating, $3\binom{n}{3}$ Δ constraints remain:

$$\begin{array}{rcl} z_1 & \leq & 1 \\ z_2 & \leq & 1 \\ 2z_1 - z_2 & \leq & 1 \\ -z_1 + 2z_2 & \leq & 1 \end{array}$$

PROP. For SRG with $n > 5$ the Δ ineq. are redundant in $\mathrm{GPP}_{\mathrm{m}}$.

SRG

- after aggregating, $3\binom{n}{3}$ Δ constraints remain:

$$\begin{aligned} z_1 &\leq 1 \\ z_2 &\leq 1 \\ 2z_1 - z_2 &\leq 1 \\ -z_1 + 2z_2 &\leq 1 \end{aligned}$$

PROP. For SRG with $n > 5$ the Δ ineq. are redundant in $\mathrm{GPP_m}$.

- However, the independent set constraints improve $\mathrm{GPP_m}$.

Simplification – 'not a special' graph . . .

Simplification – 'not a special' graph . . .

- closed form expression for the GPP for 'any' graph

Simplification – 'not a special' graph . . .

- closed form expression for the GPP for 'any' graph
- $L = \mathrm{Diag}(Au_n) - A$ . . . the Laplacian matrix of G

Simplification – 'not a special' graph . . .

- closed form expression for the GPP for 'any' graph
- $L = \mathrm{Diag}(Au_n) - A$ . . . the Laplacian matrix of G
- $\mathcal{L} := \mathrm{span}\{F_0, \ldots, F_d\}$ the Laplacian algebra corr. to L

Simplification – 'not a special' graph ...

- closed form expression for the GPP for 'any' graph
- $L = \mathrm{Diag}(Au_n) - A$ ... the Laplacian matrix of G
- $\mathcal{L} := \mathrm{span}\{F_0, \dots, F_d\}$ the Laplacian algebra corr. to L
- $F_i = U_i U_i^{\mathrm{T}}, \forall i$ ... where U_i corr. to the distinct eig. λ_i
  - $\sum_{i=0}^{d} F_i = I$
  - $F_i F_j = \delta_{ij} F_i$ for $i \neq j$
  - $\mathrm{tr}(F_i) = f_i$ ... the multiplicity of i-th eigenvalue of L

Simplification – 'not a special' graph ...

in $\mathrm{GPP_m}$:

- **relax** $\mathrm{diag}(Y) = u_n \rightsquigarrow \mathrm{tr}(Y) = n$
- **remove** nonnegativity constraints

Simplification – 'not a special' graph ...

in $\mathrm{GPP}_{\mathrm{m}}$:

- **relax** $\mathrm{diag}(Y) = u_n \rightsquigarrow \mathrm{tr}(Y) = n$
- **remove** nonnegativity constraints

$$
(\mathrm{GPP}_{\mathrm{eig}}) \qquad \begin{array}{ll} \min & \frac{1}{2}\,\mathrm{tr}\, LY \\ \text{s.t.} & \mathrm{tr}(Y) = n \\ & \mathrm{tr}(JY) = \sum_{i=1}^{k} m_i^2 \\ & kY - J_n \succeq 0 \end{array}
$$

Simplification – 'not a special' graph . . .

in $\mathrm{GPP}_{\mathrm{m}}$:

- **relax** $\mathrm{diag}(Y) = u_n \rightsquigarrow \mathrm{tr}(Y) = n$
- **remove** nonnegativity constraints

$$(\mathrm{GPP}_{\mathrm{eig}}) \quad \begin{array}{ll} \min & \frac{1}{2}\,\mathrm{tr}\, LY \\ \text{s.t.} & \mathrm{tr}(Y) = n \\ & \mathrm{tr}(JY) = \sum_{i=1}^{k} m_i^2 \\ & kY - J_n \succeq 0 \end{array}$$

- $Y = \sum_{i=0}^{d} z_i F_i, \quad z_i \in \mathbb{R} \; (i = 0, \ldots, d)$

Simplification – 'not a special' graph ...

in $\mathrm{GPP}_{\mathrm{m}}$:

- **relax** $\operatorname{diag}(Y) = u_n \rightsquigarrow \operatorname{tr}(Y) = n$
- **remove** nonnegativity constraints

$$(\mathrm{GPP}_{\mathrm{eig}}) \qquad \begin{array}{rl} \min & \frac{1}{2}\operatorname{tr} LY \\ \text{s.t.} & \operatorname{tr}(Y) = n \\ & \operatorname{tr}(JY) = \sum_{i=1}^{k} m_i^2 \\ & kY - J_n \succeq 0 \end{array}$$

- $Y = \sum_{i=0}^{d} z_i F_i, \qquad z_i \in \mathbb{R} \ (i = 0, \ldots, d)$

$$\operatorname{tr}(LY) = \operatorname{tr}\Big(\sum_{j=0}^{d} \lambda_j F_j \big(\sum_{i=0}^{d} z_i F_i\big)\Big) = \sum_{i=0}^{d} \lambda_i f_i z_i$$

where $0 = \lambda_0 \leq \ldots \leq \lambda_d$ distinct eigenvalues of L

etc. ...

Simplification – 'not a special' graph . . .

THEOREM

Let $G = (V, E)$ be a graph, $m^{\mathrm{T}} = (m_1, \ldots, m_k)$ s.t. $\sum_{j=1}^{k} m_j = n$.

Then the $\mathrm{GPP}_{\mathrm{eig}}$ bound for the minimum k-partition of G equals

$$\frac{\lambda_1}{n} \sum_{i<j} m_i m_j,$$

and the bound $\mathrm{GPP}_{\mathrm{eig}}$ for the maximum k-partition of G equals

$$\frac{\lambda_d}{n} \sum_{i<j} m_i m_j.$$

Simplification – 'not a special' graph . . .

Theorem

Let $G = (V, E)$ be a graph, $m^{\mathrm{T}} = (m_1, \dots, m_k)$ s.t. $\sum_{j=1}^{k} m_j = n$.

Then the $\mathrm{GPP}_{\mathrm{eig}}$ bound for the minimum k-partition of G equals

$$\frac{\lambda_1}{n} \sum_{i<j} m_i m_j ,$$

and the bound $\mathrm{GPP}_{\mathrm{eig}}$ for the maximum k-partition of G equals

$$\frac{\lambda_d}{n} \sum_{i<j} m_i m_j .$$

- for the **bisection** the above results coincide with:
  M. Juvan, B. Mohar: Optimal linear labelings and eigenvalues of graphs. *Discrete Appl. Math.*, 36:153–168, 1992.

- for the **min** 3-**partition**:
  J. Falkner, F. Rendl, H. Wolkowicz. A computational study of graph partitioning. *Math. Program.*, 66:211–239, 1994.

computational results . . .

Quality of the presented bounds

G	n	partition	$\mathrm{GPP}_{\mathrm{eig}}$	$\mathrm{GPP}_{\mathrm{m}}$
Doob	64	8	112	160
design	90	9	360	360
grid graph	100	(50,25,25)	4	6
Higman-Sims	100	20	950	950

Table : Lower bounds for the min graph partition.

Quality of the presented bounds

G	n	partition	$GPP_{\rm eig}$	$GPP_{\rm m}$
Doob	64	8	112	160
design	90	9	360	360
grid graph	100	(50,25,25)	4	6
Higman-Sims	100	20	950	950

Table : Lower bounds for the min graph partition.

G	n	m	$GPP_{\rm m}$	$GPP_{\rm m-\Delta}$	$GPP_{\rm m-ind}$
$J(7,2)$	21	(11,10)	37	37	40
Foster	90	(45,45)	13	18	14
Biggs-Smith	102	(70,32)	10	15	10

Table : Lower bounds for the min bisection.

- each bound computed in a few seconds

vector lifting for the GPP . . .

Vector lifting for GPP

- let $m = (m_1, \ldots, m_k)^{\mathrm{T}}, \quad \sum_i m_i = n$

Vector lifting for GPP

- let $m = (m_1, \dots, m_k)^{\mathrm{T}}, \quad \sum_i m_i = n$
- $X \in \mathcal{P}_k := \left\{X \in \mathbb{R}^{n\times k} : \; Xu_k = u_n, \; X^{\mathrm{T}} u_n = m, \; x_{ij} \in \{0,1\}\right\}$

Vector lifting for GPP

- let $m = (m_1, \ldots, m_k)^{\mathrm{T}}, \quad \sum_i m_i = n$
- $X \in \mathcal{P}_k := \left\{ X \in \mathbb{R}^{n \times k} : \; X u_k = u_n, \; X^{\mathrm{T}} u_n = m, \; x_{ij} \in \{0,1\} \right\}$
  define $y := \mathrm{vec}(X), \; Y := yy^{\mathrm{T}} \rightsquigarrow$ relax $Y - yy^{\mathrm{T}} \succeq 0$

Vector lifting for GPP

- let $m = (m_1, \ldots, m_k)^{\mathrm{T}}, \quad \sum_i m_i = n$
- $X \in \mathcal{P}_k := \{X \in \mathbb{R}^{n \times k} : \; Xu_k = u_n, \; X^{\mathrm{T}} u_n = m, \; x_{ij} \in \{0,1\}\}$
  define $y := \mathrm{vec}(X)$, $Y := yy^{\mathrm{T}} \rightsquigarrow$ relax $Y - yy^{\mathrm{T}} \succeq 0$

$$
(\mathrm{GPP_v}) \quad
\begin{array}{ll}
\min & \frac{1}{2}\mathrm{tr}((J_k - I_k) \otimes A)Y \\
\text{s.t.} & \mathrm{tr}((J_k - I_k) \otimes I_n)Y = 0 \\
& \mathrm{tr}(I_k \otimes J_n)Y + \mathrm{tr}(Y) = -(\sum_{i=1}^{k} m_i^2 + n) \\
& \qquad + 2y^{\mathrm{T}}((m + u_k) \otimes u_n) \\
& \begin{pmatrix} 1 & y^{\mathrm{T}} \\ y & Y \end{pmatrix} \in \mathcal{S}_{nk+1}^{+}, \quad Y \geq 0
\end{array}
$$

H. Wolkowicz and Q. Zhao. Semidefinite programming relaxations for the graph partitioning problem. *Discrete Appl. Math.*, 96–97:461–479, 1999.

- original Zhao-Wolkowicz relaxation does not include $Y \geq 0$

Vector lifting for GPP

Theorem (S., 2012)
When restricted to the equipartition, $\mathrm{GPP_v}$ and $\mathrm{GPP_m}$ are equivalent.

Vector lifting for GPP

Theorem (S., 2012)
When restricted to the equipartition, $\mathrm{GPP_v}$ and $\mathrm{GPP_m}$ are equivalent.

Theorem (S., 2013)
When restricted to the bisection, $\mathrm{GPP_v}$ dominates $\mathrm{GPP_m}$.

Vector lifting for GPP

Theorem (S., 2012)
When restricted to the equipartition, $\mathrm{GPP_v}$ and $\mathrm{GPP_m}$ are equivalent.

Theorem (S., 2013)
When restricted to the bisection, $\mathrm{GPP_v}$ dominates $\mathrm{GPP_m}$.

NUMERICAL EXPERIMENTS SHOW:

- gap between $\mathrm{GPP_v}$ and $\mathrm{GPP_m}$ reduces for $k > 5$

How to strengthen $\mathrm{GPP}_{\mathrm{v}}$?

How to strengthen $\mathrm{GPP_v}$?

We demonstrate for the bisection problem.

New bound for the bisection

$\Rightarrow$ assign a pair of vertices of G to different parts of the partition

New bound for the bisection

$\Rrightarrow$ assign a pair of vertices of G to different parts of the partition

Which pair of vertices?

New bound for the bisection

$\Rightarrow$ assign a pair of vertices of G to different parts of the partition

Which pair of vertices?

- consider the action of $\mathrm{aut}(A)$ on the pair of vertices (i, j)
  i.e., orbital: $\{(Pe_i, Pe_j) : P \in \mathrm{aut}(A)\}$

New bound for the bisection

$\Rrightarrow$ assign a pair of vertices of G to different parts of the partition

Which pair of vertices?

- consider the action of $\mathrm{aut}(A)$ on the pair of vertices (i, j)
  i.e., orbital: $\{(Pe_i, Pe_j) : P \in \mathrm{aut}(A)\}$
- orbitals represent the '**different**' kinds of pairs of vertices

New bound for the bisection

$\Rrightarrow$ assign a pair of vertices of G to different parts of the partition

Which pair of vertices?

- consider the action of $\mathrm{aut}(A)$ on the pair of vertices (i, j) i.e., orbital: $\{(Pe_i, Pe_j) : P \in \mathrm{aut}(A)\}$
- orbitals represent the '**different**' kinds of pairs of vertices
- assume that there are t such orbitals: $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$

$\Rrightarrow$ we prove the following

New bound for the bisection

THEOREM. Let G be an undirected graph with adjacency matrix A, and t orbitals $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$ of edges and nonedges.

New bound for the bisection

THEOREM. Let G be an undirected graph with adjacency matrix A, and t orbitals $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$ of edges and nonedges.

Let (r_{h1}, r_{h2}) be an arbitrary pair of vertices in $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$.

New bound for the bisection

THEOREM. Let G be an undirected graph with adjacency matrix A, and t orbitals $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$ of edges and nonedges.

Let (r_{h1}, r_{h2}) be an arbitrary pair of vertices in $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$.

Then

$$\min_{Z \in \mathcal{P}_2} \operatorname{tr} Z^{\mathrm{T}} A Z (J_2 - I_2) = \min_{h=1,2,\ldots,t} \min_{X \in \mathcal{P}_2(h)} \operatorname{tr} X^{\mathrm{T}} A X (J_2 - I_2),$$

where $\mathcal{P}_2(h) = \{X \in \mathcal{P}_2 : X_{r_{h1},1} = 1, X_{r_{h2},2} = 1\}$ $\;(h = 1, 2, \ldots, t)$.

New bound for the bisection

THEOREM. Let G be an undirected graph with adjacency matrix A, and t orbitals $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$ of edges and nonedges.

Let (r_{h1}, r_{h2}) be an arbitrary pair of vertices in $\mathcal{O}_h$ $(h = 1, 2, \ldots, t)$.

Then

$$\min_{Z \in \mathcal{P}_2} \operatorname{tr} Z^{\mathrm{T}} A Z (J_2 - I_2) = \min_{h=1,2,\ldots,t} \min_{X \in \mathcal{P}_2(h)} \operatorname{tr} X^{\mathrm{T}} A X (J_2 - I_2),$$

where $\mathcal{P}_2(h) = \{X \in \mathcal{P}_2 : X_{r_{h1},1} = 1, X_{r_{h2},2} = 1\}$ $\ (h = 1, 2, \ldots, t)$.

$\Rightarrow$ for each h, compute:

$$\mu_h^* := \{\mathrm{GPP_v} \ \text{ with two additional constraints}\}$$

New bound for the bisection

THEOREM. Let G be an undirected graph with adjacency matrix A, and t orbitals $\mathcal{O}_h$ $(h = 1, 2, \dots, t)$ of edges and nonedges.

Let (r_{h1}, r_{h2}) be an arbitrary pair of vertices in $\mathcal{O}_h$ $(h = 1, 2, \dots, t)$.

Then

$$\min_{Z \in \mathcal{P}_2} \operatorname{tr} Z^{\mathrm{T}} A Z (J_2 - I_2) = \min_{h=1,2,\dots,t} \min_{X \in \mathcal{P}_2(h)} \operatorname{tr} X^{\mathrm{T}} A X (J_2 - I_2),$$

where $\mathcal{P}_2(h) = \{X \in \mathcal{P}_2 : X_{r_{h1},1} = 1, X_{r_{h2},2} = 1\}$ $\;(h = 1, 2, \dots, t)$.

$\Rightarrow$ for each h, compute:

$$\mu_h^* := \{\mathrm{GPP}_{\mathrm{v}} \text{ with two additional constraints}\}$$

$\Rightarrow$ the new lower bound for the bisection problem is:

$$\mathrm{GPP}_{\mathrm{fix}} := \min_{h=1,\dots,t} \mu_h^*$$

computational results . . .

Comparison of bounds ...

- in general, it is *difficult* to solve $\mathrm{GPP}_{\mathrm{fix}}$

Comparison of bounds ...

- in general, it is *difficult* to solve $\mathrm{GPP}_{\mathrm{fix}}$

but for graphs with symmetry ...

G	n	m^{T}	$\mathrm{GPP}_{\mathrm{m}}$	$\mathrm{GPP}_{\mathrm{v}}$	$\mathrm{GPP}_{\mathrm{m-ind}}$	$\mathrm{GPP}_{\mathrm{fix}}$
$J(6,2)$	15	(8,7)	23	23	26	24
Gewirtz	56	(53,3)	23	24	23	26
M_{22}	77	(74,3)	41	42	41	44
Higman-Sims	100	25-part.	960	960	960	964

Table : Lower bounds for the min GPP

- each bound computed with IPM in $< 30s$

Example: the bandwidth problem

Example: the bandwidth problem

The Bandwidth Problem in graphs:

label the vertices v_i of G with distinct integers $\phi(v_i)$

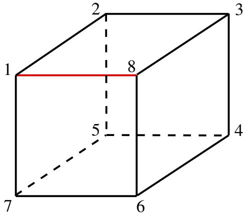
$$\text{s.t.} \qquad \max_{(v_i,v_j)\in E} |\phi(v_i) - \phi(v_j)| \quad \text{minimal}$$

Example: the bandwidth problem

The Bandwidth Problem in graphs:

label the vertices v_i of G with distinct integers $\phi(v_i)$

$$\text{s.t.} \quad \max_{(v_i,v_j)\in E} |\phi(v_i) - \phi(v_j)| \quad \text{minimal}$$

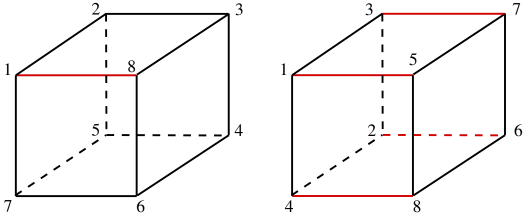


Example: the bandwidth problem

The Bandwidth Problem in graphs:

label the vertices v_i of G with distinct integers $\phi(v_i)$

$$\text{s.t.} \qquad \max_{(v_i,v_j)\in E} |\phi(v_i) - \phi(v_j)| \quad \text{minimal}$$

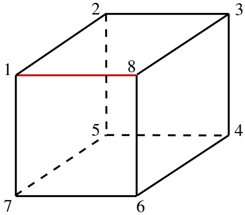


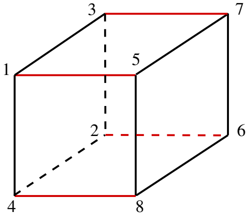
Example: the bandwidth problem

The Bandwidth Problem in graphs:

label the vertices v_i of G with distinct integers $\phi(v_i)$

$$\text{s.t.} \qquad \max_{(v_i,v_j)\in E} |\phi(v_i) - \phi(v_j)| \quad \text{minimal}$$





$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

The bandwidth problem

$\Rrightarrow$ the bandwidth problem is related to the following GPP problem

The bandwidth problem

⇒ the bandwidth problem is related to the following GPP problem

The min-cut problem is:

$$\begin{aligned} \mathrm{OPT}_{\mathrm{MC}} := \quad & \min \sum_{i \in S_1, j \in S_2} a_{ij} \\ & \text{s.t.} \quad (S_1, S_2, S_3) \text{ partitions } V \\ & \qquad |S_i| = m_i, \quad i = 1, 2, 3, \end{aligned}$$

where $A = (a_{ij})$ is the adjacency matrix of G.

The bandwidth problem

$\Rrightarrow$ the bandwidth problem is related to the following GPP problem

The min-cut problem is:

$$\begin{aligned} \mathrm{OPT}_{\mathrm{MC}} := \quad \min \quad & \sum_{i \in S_1, j \in S_2} a_{ij} \\ \text{s.t.} \quad & (S_1, S_2, S_3) \ \text{ partitions } \ V \\ & |S_i| = m_i, \quad i = 1, 2, 3, \end{aligned}$$

where $A = (a_{ij})$ is the adjacency matrix of G.

bandwidth lower bound (Povh-Rendl (2007), van Dam-S.):

If for some $m = (m_1, m_2, m_3)$ it holds that $\mathrm{OPT}_{\mathrm{MC}} \geq \nu > 0$, then
$$\sigma_\infty(G) \geq m_3 + \left\lceil -\frac{1}{2} + \sqrt{2\nu + \frac{1}{4}} \right\rceil$$

The bandwidth problem - SDP relaxation

SDP relaxations for the min-cut:

- solve $\mathrm{GPP}_{\mathrm{v}}$ and $\mathrm{GPP}_{\mathrm{fix}}$ with objective

$$\frac{1}{2}\mathrm{trace}(D \otimes A)Y$$

where

$$D = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Bandwidth of Hamming graphs . . .

- Hamming graph $H(d,q)$ is the graph Cartesian product of d copies of the complete graph K_q.

Bandwidth of Hamming graphs ...

- Hamming graph $H(d,q)$ is the graph Cartesian product of d copies of the complete graph K_q.

q	♯ nodes	old	bw_v	time(s)	$\mathrm{bw}_{\mathrm{fix}}$	time(s)	u.b.
3	27	9	10	0	12	44	13
4	64	22	22	3	25	176	31
5	125	42	43	15	47	536	60
6	216	72	74	76	78	1756	101

Table : Bounds on the bandwidth of $H(3,q)$

- bw_{v} and $\mathrm{bw}_{\mathrm{fix}}$ obtained by use of: $m_3 + \left\lceil -\frac{1}{2} + \sqrt{2\alpha + \frac{1}{4}} \right\rceil$
- upper bounds obtained by improved rev. Cuthill-McKee algor.

More on bounds . . .

we also compute the best known lower/upper bounds for:

- $H(4, q)$
- the 3-dimensional generalized Hamming graphs $H_{q1,q2,q3}$
- the Johnson and Kneser graphs

ThAnK YoU!