Strong SDP bounds for large k-equipartition problems via a cutting plane ADMM-based algorithm

Shudian Zhao[†]

Frank de Meijer[*], Renata Sotirov[*], Angelika Wiegele[†]

[†] Alpen-Adria-Universität Klagenfurt
[*] Tilburg University

May 19
Modern Techniques of Very Large Scale Optimization

MINOA
MIXED-INTEGER NON-LINEAR OPTIMISATION:
ALGORITHMS AND APPLICATIONS

ALPEN-ADRIA
UNIVERSITÄT
KLAGENFURT

Table of Contents

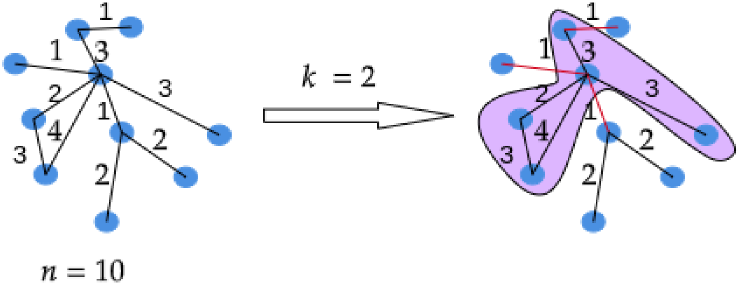
1. The k-equipartition Problem

2. Semidefinite Relaxations for the *k*-equipartition Problem

3. The Cutting-Plane ADMM-based Algorithm

The *k*-equipartition Problem

Partition an undirected graph $G(V, E)$ into k groups with equal cardinality such that the weight of edges cut by the partition is minimized.



The *k*-equipartition problem is NP-hard.

Table of Contents

1 The k-equipartition Problem

2 Semidefinite Relaxations for the k-equipartition Problem

3 The Cutting-Plane ADMM-based Algorithm

Semidefinite Programming

Positive semidefinite (PSD) matrices:
$X \in \mathcal{S}^n, X \succeq 0 \iff v^\top X v \geq 0 \; \forall v \in \mathbb{R}^n$.
The set of PSD matrices is a convex cone, denoted by $\mathcal{S}^n_+$.

Semidefinite Programming

Positive semidefinite (PSD) matrices:
$X \in \mathcal{S}^n, X \succeq 0 \iff v^\top X v \geq 0 \ \forall v \in \mathbb{R}^n$.
The set of PSD matrices is a convex cone, denoted by $\mathcal{S}^n_+$.

- Semidefinite programming (SDP) problems:

$$\begin{aligned} &\min \ \langle C, X \rangle \\ &\ \text{s.t. } \mathcal{A}(X) = b, X \succeq 0. \end{aligned} \tag{SDP-Primal}$$

  where $\mathcal{A} : \mathcal{S}^n \to \mathbb{R}^m$, $C \in \mathcal{S}^n$, $b \in \mathbb{R}^m$, and

  $\langle C, X \rangle = \mathrm{trace}(CX) = \sum_{ij} C_{ij} X_{ij}$, and $\mathcal{A}(X) = \begin{pmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{pmatrix}$.

- SDP problems can be solved by polynomial-time algorithms.

Duality in Semidefinite Programming

$$\begin{aligned} \max \ & b^\top y \\ \text{s.t. } & \mathcal{A}^* y + Z = C, \\ & Z \succeq 0. \end{aligned} \tag{SDP-Dual}$$

where $\mathcal{A}^* : \mathbb{R}^m \to \mathcal{S}^n$ is the adjoint operator of $\mathcal{A}(\cdot)$ and $\mathcal{A}^* y = \sum_{i=1}^m A_i^\top y$.

Duality in Semidefinite Programming

$$\begin{aligned} \max\ & b^\top y \\ \text{s.t. } & \mathcal{A}^* y + Z = C, \\ & Z \succeq 0. \end{aligned} \tag{SDP-Dual}$$

where $\mathcal{A}^* : \mathbb{R}^m \to \mathcal{S}^n$ is the adjoint operator of $\mathcal{A}(\cdot)$ and $\mathcal{A}^* y = \sum_{i=1}^m A_i^\top y$.

The strong duality theorem

If both, the primal and dual problems, are strictly feasible, then $p^* = d^*$ and both optima are attained.

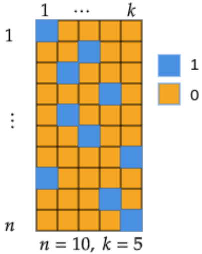
The k-equipartition Problem

An undirected graph $G(V,E)$, $|V| = n$, A is the adjacency matrix where A_{ij} is the weight of edge $(i,j) \in E$, k is a divisor of n.

The k-equipartition Problem

An undirected graph $G(V, E)$, $|V| = n$, A is the adjacency matrix where A_{ij} is the weight of edge $(i, j) \in E$, k is a divisor of n.

$Y \in \{0, 1\}^{n \times k}$ encodes a k-equipartition for $G(V, E)$ if certain constraints are satisfied.



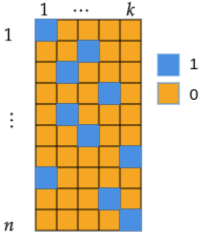
The k-equipartition Problem

An undirected graph $G(V,E)$, $|V| = n$, A is the adjacency matrix where A_{ij} is the weight of edge $(i,j) \in E$, k is a divisor of n.
$Y \in \{0,1\}^{n\times k}$ encodes a k-equipartition for $G(V,E)$ if certain constraints are satisfied.

- L: Laplacian matrix, $L = \mathrm{diag}(Ae_n) - A$.

$$\frac{1}{2}\langle L, YY^\top\rangle = \frac{1}{2}\sum_{i,j} L_{ij} y_i^\top y_j = \sum_{i,j,i<j} A_{ij}(1 - y_i^\top y_j),$$

where y_i is the i-th row of Y, e_n is the all-ones vector of length n.



$n = 10,\ k = 5$

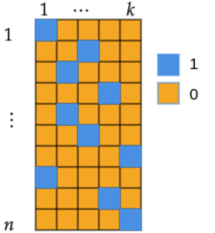
The k-equipartition Problem

An undirected graph $G(V,E)$, $|V| = n$, A is the adjacency matrix where A_{ij} is the weight of edge $(i,j) \in E$, k is a divisor of n.
$Y \in \{0,1\}^{n\times k}$ encodes a k-equipartition for $G(V,E)$ if certain constraints are satisfied.

- L: Laplacian matrix, $L = \mathrm{diag}(Ae_n) - A$.

$$\frac{1}{2}\langle L, YY^\top\rangle = \frac{1}{2}\sum_{i,j} L_{ij} y_i^\top y_j = \sum_{i,j,i<j} A_{ij}(1 - y_i^\top y_j),$$

where y_i is the i-th row of Y, e_n is the all-ones vector of length n.



$n = 10,\ k = 5$

The k-equipartition Problem

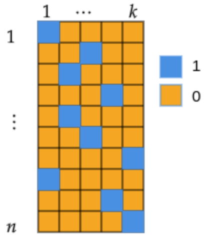
An undirected graph $G(V,E)$, $|V|=n$, A is the adjacency matrix where A_{ij} is the weight of edge $(i,j)\in E$, k is a divisor of n.
$Y\in\{0,1\}^{n\times k}$ encodes a k-equipartition for $G(V,E)$ if certain constraints are satisfied.

- L: Laplacian matrix, $L=\mathrm{diag}(Ae_n)-A$.

$$\frac{1}{2}\langle L, YY^\top\rangle = \frac{1}{2}\sum_{i,j} L_{ij}y_i^\top y_j = \sum_{i,j,i<j} A_{ij}(1-y_i^\top y_j),$$

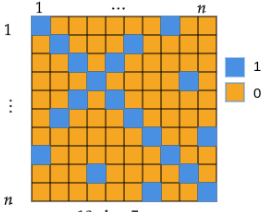
  where y_i is the i-th row of Y, e_n is the all-ones vector of length n.
- m: Cardinality in each group, $m=\frac{n}{k}$.

$$\begin{aligned} \min\ & \frac{1}{2}\langle L, YY^\top\rangle \\ \text{s.t. } & Ye_k = e_n, \\ & Y^\top e_n = me_k, \\ & Y\in\{0,1\}^{n\times k}. \end{aligned}$$



$n=10,\ k=5$

- $X := YY^\top$.
- $$X_{ij} = \begin{cases} 1, & i, j \text{ in the same group} \\ 0, & \text{otherwise} \end{cases}$$



$n = 10,\ k = 5$

$$\begin{aligned} \min\ & \frac{1}{2}\langle L, X\rangle \\ \text{s.t. } & \operatorname{diag}(X) = e, \\ & Xe = me, \\ & X \in \{0,1\}^{n\times n}, X \in \mathcal{S}^n. \end{aligned}$$

The Doubly Nonnegative (DNN) Relaxation for k-equipartitioning

Relax the binary variables:

$$X = YY^\top \implies X \succeq 0, X \geq 0.$$

$$\begin{aligned} \min \ & \frac{1}{2}\langle L, X\rangle \\ \text{s.t. } & \operatorname{diag}(X) = e, \\ & Xe = me, \\ & X \succeq 0, \\ & X \geq 0. \end{aligned} \tag{DNN}$$

The Doubly Nonnegative (DNN) Relaxation for *k*-equipartitioning

Relax the binary variables:

$$X = YY^\top \implies X \succeq 0, X \geq 0.$$

$$\begin{aligned} \min \ & \frac{1}{2}\langle L, X\rangle \\ \text{s.t. } & \operatorname{diag}(X) = e, \\ & Xe = me, \\ & X \succeq 0, \\ & X \geq 0. \end{aligned} \qquad (DNN)$$

Number of Constraints: $\frac{(n+1)n}{2}$

Valid Inequalities

- Triangle inequality constraints:
  $\mathcal{H}_T := \{X \in \mathcal{S}^n \mid X_{ij} + X_{i\ell} \leq 1 + X_{j\ell}, \forall i, j, \ell \in [n], i \neq j \neq \ell\}$.
  Number of Constraints.: $3\binom{n}{3} = \frac{n(n-1)(n-2)}{2}$.
- Independent set constraints:
  $\mathcal{H}_C := \{X \in \mathcal{S}^n \mid \sum_{i,j \in I} X_{ij} \geq 1 \ \forall I \subset V \ |I| = k+1\}$.
  Number of Constraints.: $\binom{n}{k+1} = \frac{n(n-1)\cdots(n-k)}{(k+1)!}$.

Table of Contents

1. The k-equipartition Problem

2. Semidefinite Relaxations for the k-equipartition Problem

3. The Cutting-Plane ADMM-based Algorithm

The Cutting-plane Framework

1. Solve DNN relaxations and obtain X^0
2. While stopping criteria not met
    1. Add *numCuts* most violated cuts and form a new relaxation
    2. Solve the new relaxation and obtain X^0

Problems to solve:

- The DNN relaxation;
- The DNN relaxation + Polyhedral cuts.

The Cutting-plane Framework

1. Solve DNN relaxations and obtain X^0
2. While stopping criteria not met
    1. Add *numCuts* most violated cuts and form a new relaxation
    2. Solve the new relaxation and obtain X^0

Problems to solve:

- The DNN relaxation;
- The DNN relaxation + Polyhedral cuts.

Algorithm for Solving SDPs

1. Interior point methods (IPMs): failed to solve large scale instances, e.g., Mosek.
2. The alternating direction method of multipliers (ADMM): efficient on solving large instances but cannot reach a high precision.

The Cutting-plane Framework

1. Solve DNN relaxations and obtain X^0
2. While stopping criteria not met
    1. Add *numCuts* most violated cuts and form a new relaxation
    2. Solve the new relaxation and obtain X^0

Use ADMM with post-processing to solve each SDP problem

Problems to solve:

- The DNN relaxation;
- The DNN relaxation + Polyhedral cuts.

Algorithm for Solving SDPs

1. Interior point methods (IPMs): failed to solve large scale instances, e.g., Mosek.
2. The alternating direction method of multipliers (ADMM): efficient on solving large instances but cannot reach a high precision.

Variants of ADMM for Solving SDPs

- Malick, Povh, Rendl, and Wiegele (2009) and Wen, Goldfarb, and Yin (2010): Apply the ADMM on the dual SDP:

$$\min\{\langle C, X\rangle \mid \mathcal{A}(X) = b, X \succeq 0\}, \qquad \text{(SDP-Primal)}$$
$$\max\{b^\top y \mid \mathcal{A}^* y + Z = C, Z \succeq 0\}. \qquad \text{(SDP-Dual)}$$

- Sun, Toh, Yuan, and Zhao (2020), Cerulli, Santis, Gaar, and Wiegele (2021), and Wiegele and Zhao (2022): Variants of ADMM for solving SDP problems with inequality constraints.
- Oliveira, Wolkowicz, and Xu (2018), Hu, Sotirov, and Wolkowicz (2019), and Li, Pong, Sun, and Wolkowicz (2021): The ADMM-based algorithm for symmetry and facially reduced DNN relaxations.
- de Meijer and Sotirov (2021): An augmented Lagrangian method incorporated in a cutting-plane framework for quadratic cycle cover problems.

The Reformulated DNN Relaxation with Facial Reduction

Facial Reduction

Given $V \in \mathbb{R}^{n\times(n-1)}$ such that $V^\top e = 0$ and $\mathrm{rank}(V) = n-1$, we have $X = VRV^\top + \frac{1}{k}E$ for $R \in \mathcal{S}^{n-1}$.

$$\begin{aligned} \min\ & \frac{1}{2}\langle L, X\rangle \\ \text{s.t. } & \mathrm{diag}(X) = e, \\ & Xe = me, \\ & X \geq 0, \\ & X \succeq 0. \end{aligned} \qquad (DNN)$$

The Reformulated DNN Relaxation with Facial Reduction

Facial Reduction

Given $V \in \mathbb{R}^{n\times(n-1)}$ such that $V^\top e = 0$ and $\mathrm{rank}(V) = n-1$, we have $X = VRV^\top + \frac{1}{k}E$ for $R \in \mathcal{S}^{n-1}$.

$$
\begin{aligned}
\min\ & \frac{1}{2}\langle L, X\rangle\\
\text{s.t. } & \mathrm{diag}(X) = e,\\
& Xe = me,\\
& X \geq 0,\\
& X \succeq 0.
\end{aligned}
\qquad (DNN) \qquad
\begin{aligned}
\min\ & \langle \frac{1}{2}L, VRV^\top\rangle\\
\text{s.t. } & \mathrm{diag}(VRV^\top) = \frac{k-1}{k}e_n,\\
& VRV^\top \geq -\frac{1}{k}E_n,\\
& R \succeq 0,
\end{aligned}
$$

where

$$V = \begin{pmatrix} I_{n-1} \\ -e_{n-1}^\top \end{pmatrix}.$$

The ADMM on the Facial Reduced Primal SDPs[1]

$$\begin{aligned} \min\ & \langle \frac{1}{2}L, X\rangle \\ \text{s.t. } & X = VRV^\top, X \in \mathcal{X}, R \in \mathcal{R}, \end{aligned}$$

[1] Shudian Zhao (2022). "Splitting into Pieces: Alternating Direction Methods of Multipliers and Graph Partitioning". PhD thesis. Alpen-Adria-Universität Klagenfurt.

The ADMM on the Facial Reduced Primal SDPs[1]

$$
\begin{aligned}
\min\ & \langle \frac{1}{2}L, X\rangle \\
\text{s.t. } & X = VRV^\top, X \in \mathcal{X}, R \in \mathcal{R}, \\
& \mathcal{X} := \{X \in \mathcal{S}^n \mid \operatorname{diag}(X) = \frac{k-1}{k}e_n,\ X \geq -\frac{1}{k}E_n\}, \\
& \mathcal{R} := \{R \in \mathcal{S}^{n-1} \mid R \succeq 0\}.
\end{aligned}
$$

The augmented Lagrangian function is

$$
\mathcal{L}_\sigma(X, R, Z) = \langle \frac{1}{2}L, X\rangle + \langle Z, X - VRV^\top\rangle + \frac{\sigma}{2}\|X - VRV^\top\|_F^2. \tag{2}
$$

[1] Shudian Zhao (2022). "Splitting into Pieces: Alternating Direction Methods of Multipliers and Graph Partitioning". PhD thesis. Alpen-Adria-Universität Klagenfurt.

The ADMM on the Facial Reduced Primal SDPs[1]

$$\begin{aligned} \min\ & \langle \frac{1}{2} L, X \rangle \\ \text{s.t. } & X = VRV^\top, X \in \mathcal{X}, R \in \mathcal{R}, \\ & \mathcal{X} := \{X \in \mathcal{S}^n \mid \mathrm{diag}(X) = \frac{k-1}{k} e_n,\ X \geq -\frac{1}{k} E_n\}, \\ & \mathcal{R} := \{R \in \mathcal{S}^{n-1} \mid R \succeq 0\}. \end{aligned}$$

The augmented Lagrangian function is

$$\mathcal{L}_\sigma(X, R, Z) = \langle \frac{1}{2} L, X \rangle + \langle Z, X - VRV^\top \rangle + \frac{\sigma}{2} \|X - VRV^\top\|_F^2. \tag{2}$$

The p-th iterate is

$$R^{p+1} = \arg\min_{R \in \mathcal{R}} \mathcal{L}_\sigma(R, X^p, Z^p), \tag{3a}$$

$$X^{p+1} = \arg\min_{X \in \mathcal{X}} \mathcal{L}_\sigma(R^{p+1}, X, Z^p), \tag{3b}$$

$$Z^{p+1} = Z^p + \sigma(X^{p+1} - VR^{p+1}V^\top). \tag{3c}$$

[1] Shudian Zhao (2022). "Splitting into Pieces: Alternating Direction Methods of Multipliers and Graph Partitioning". PhD thesis. Alpen-Adria-Universität Klagenfurt.

R-subproblem and X-subproblem

- The R-subproblem (3a) can be solved as follows

$$\begin{aligned} R^{p+1} =& \arg\min_{R\in\mathcal{R}} \mathcal{L}_\sigma(R, X^p, Z^p) \\ =& \arg\min_{R\in\mathcal{R}} \ \langle Z^p, -VRV^\top \rangle + \frac{\sigma}{2} \left\| X^p - VRV^\top \right\|_F^2 \\ =& \arg\min_{R\in\mathcal{R}} \ \left\| V^\top \left(X^p + \frac{1}{\sigma} Z^p \right) V - R \right\|_F^2 \\ =& \mathcal{P}_{\succeq \mathbf{0}} \left(V^\top \left(X^p + \frac{1}{\sigma} Z^p \right) V \right), \end{aligned}$$

where $\mathcal{P}_{\succeq \mathbf{0}}(\cdot)$ is the projection onto the cone of positive semidefinite matrices.

R-subproblem and X-subproblem

- The R-subproblem (3a) can be solved as follows

$$\begin{aligned} R^{p+1} =& \arg\min_{R\in\mathcal{R}} \mathcal{L}_\sigma(R, X^p, Z^p) \\ =& \mathcal{P}_{\succeq \mathbf{0}}\left(V^\top\left(X^p + \frac{1}{\sigma}Z^p\right)V\right), \end{aligned}$$

- Similarly, X-subproblem is a projection problem onto $\mathcal{X}$

$$\begin{aligned} X^{p+1} =& \arg\min_{X\in\mathcal{X}} \mathcal{L}_\sigma(R^{p+1}, X, Z^p) \\ =& \mathcal{P}_{\mathcal{X}}\left(VR^{p+1}V^\top - \frac{1}{\sigma}\left(\frac{1}{2}L + Z^p\right)\right), \end{aligned}$$

where $\mathcal{X} = \{X \in \mathcal{S}^n \mid \mathrm{diag}(X) = \frac{k-1}{k}e_n,\ X \geq -\frac{1}{k}E_n\}$.

X-subproblem with Cutting Planes

The Cutting-plane Framework

1. Solve DNN relaxations and obtain X
2. While stopping criteria not met
    1. Add *numCuts* most violated cuts and form a new relaxation $\mathcal{X} \rightsquigarrow \mathcal{X}_{\mathcal{T}}$.
    2. Solve the new relaxation and obtain X $\rightsquigarrow$ Solve $X^{p+1} = \mathcal{P}_{\mathcal{X}_{\mathcal{T}}}\left(VR^{p+1}V^{\top} - \frac{1}{\sigma}\left(\frac{1}{2}L + Z^{p}\right)\right)$ in the p-th iterate.

X-subproblem with Cutting Planes

The Cutting-plane Framework

1. Solve DNN relaxations and obtain X
2. While stopping criteria not met
    1. Add *numCuts* most violated cuts and form a new relaxation $\mathcal{X} \rightsquigarrow \mathcal{X}_{\mathcal{T}}$.
    2. Solve the new relaxation and obtain X $\rightsquigarrow$ Solve $X^{p+1} = \mathcal{P}_{\mathcal{X}_{\mathcal{T}}}\left(VR^{p+1}V^\top - \frac{1}{\sigma}\left(\frac{1}{2}L + Z^p\right)\right)$ in the p-th iterate.

Given an index set $\mathcal{T}$ for triangle cuts, then adding the cuts in $\mathcal{T}$ to the DNN relaxation, the polyhedral set $\mathcal{X}$ has to be replaced by

$$\mathcal{X}_{\mathcal{T}} := \mathcal{X} \cap \left(\bigcap_{t \in \mathcal{T}} \mathcal{H}_t\right),$$

$$\forall t = (i,j,\ell),\ \mathcal{H}_{ij\ell} := \left\{X \in \mathcal{S}^n \,:\, X_{ij} + X_{i\ell} \le \frac{k-1}{k} + X_{j\ell}\right\}.$$

Projection onto $\mathcal{H}_{ij\ell}$

Lemma

Let $M \in \mathcal{S}^n$ and let $\hat{M} := \mathcal{P}_{\mathcal{H}_{ij\ell}}(M)$. If $M \in \mathcal{H}_{ij\ell}$, then $\hat{M} = M$. If $M \notin \mathcal{H}_{ij\ell}$, then $\hat{M}$ is such that

$$\hat{M}_{pq} = \begin{cases} \frac{2}{3}M_{ij} - \frac{1}{3}M_{i\ell} + \frac{1}{3}M_{j\ell} + \frac{1}{3} - \frac{1}{3k} & \text{if } (p,q) \in \{(i,j),(j,i)\}, \\ -\frac{1}{3}M_{ij} + \frac{2}{3}M_{i\ell} + \frac{1}{3}M_{j\ell} + \frac{1}{3} - \frac{1}{3k} & \text{if } (p,q) \in \{(i,\ell),(\ell,i)\}, \\ \frac{1}{3}M_{ij} + \frac{1}{3}M_{i\ell} + \frac{2}{3}M_{j\ell} - \frac{1}{3} + \frac{1}{3k} & \text{if } (p,q) \in \{(j,\ell),(\ell,j)\}, \\ M_{pq} & \text{otherwise.} \end{cases}$$

Proof.

Solve the best approximation problem by KKT condition.

$$\min \ \|\hat{M} - M\|_F^2 \text{ s.t. } \hat{M} \in \mathcal{H}_{ij\ell}. \tag{4}$$

□

Dykstra's Algorithm

Dykstra's algorithm (Boyle and Dykstra, 1985) can project onto the intersection of a finite number of polyhedral sets.
Input: $M \in \mathcal{S}^n$.
Output: $X^p = \arg\min \|\hat{M} - M\|^2$ s.t. $\hat{M} \in \mathcal{X}_{\mathcal{T}}$.

Dykstra's Algorithm

Dykstra's algorithm (Boyle and Dykstra, 1985) can project onto the intersection of a finite number of polyhedral sets.
Input: $M \in \mathcal{S}^n$.
Output: $X^p = \arg\min \|\hat{M} - M\|^2$ s.t. $\hat{M} \in \mathcal{X}_{\mathcal{T}}$.
Initialize: The normal matrices $N_{\mathcal{X}}^0 = \mathbf{0}$ and $N_t^0 = \mathbf{0}$ for all $t \in \mathcal{T}$.
$X^0 = M$.

Dykstra's Algorithm

Dykstra's algorithm (Boyle and Dykstra, 1985) can project onto the intersection of a finite number of polyhedral sets.
Input: $M \in \mathcal{S}^n$.
Output: $X^p = \arg\min \|\hat{M} - M\|^2$ s.t. $\hat{M} \in \mathcal{X}_\mathcal{T}$.
Initialize: The normal matrices $N_\mathcal{X}^0 = \mathbf{0}$ and $N_t^0 = \mathbf{0}$ for all $t \in \mathcal{T}$.
$X^0 = M$.
The algorithm iterates for $p \geq 1$ as follows:
while $\|X^{p+1} - X^p\|_F > \varepsilon_{proj}$ do

$$X^p = \mathcal{P}_\mathcal{X}\left(X^{p-1} + N_\mathcal{X}^{p-1}\right)$$

$$N_\mathcal{X}^p = X^{p-1} + N_\mathcal{X}^{p-1} - X^p$$

Dykstra's Algorithm

Dykstra's algorithm (Boyle and Dykstra, 1985) can project onto the intersection of a finite number of polyhedral sets.
Input: $M \in \mathcal{S}^n$.
Output: $X^p = \arg\min \|\hat{M} - M\|^2$ s.t. $\hat{M} \in \mathcal{X}_{\mathcal{T}}$.
Initialize: The normal matrices $N_{\mathcal{X}}^0 = \mathbf{0}$ and $N_t^0 = \mathbf{0}$ for all $t \in \mathcal{T}$.
$X^0 = M$.
The algorithm iterates for $p \geq 1$ as follows:
while $\|X^{p+1} - X^p\|_F > \varepsilon_{proj}$ do

$$\begin{aligned} X^p &= \mathcal{P}_{\mathcal{X}}\left(X^{p-1} + N_{\mathcal{X}}^{p-1}\right) \\ N_{\mathcal{X}}^p &= X^{p-1} + N_{\mathcal{X}}^{p-1} - X^p \\ &\left.\begin{aligned} L_t &= X^p + N_t^{p-1} \\ X^p &= \mathcal{P}_{\mathcal{H}_t}(L_t) \\ N_t^p &= L_t - X^p \end{aligned}\right\} \quad \text{for all } t \in \mathcal{T} \end{aligned} \tag{CycDyk}$$

end

The Cutting Plane ADMM-based Algorithm

Algorithm 1: The CP–ADMM

```
Data: The weighted Laplacian matrix $L$, $m = \frac{n}{k}$, $V$;
Input: $UB$, $\varepsilon_{\mathrm{ADMM}}$, $\varepsilon_{\mathrm{proj}}$, maxIter, numCuts, maxOuterLoops;
Output: Valid lower bound $lb(Z^p)$ ;
Initialization: Set $(R^0, X^0, Z^0)$ and $\sigma^0$, $p = 0$, $\mathcal{T} = \emptyset$ ;
while stopping criteria not met do
    while stopping criteria not met do
        $R^{p+1} = \mathcal{P}_{\succeq \mathbf{0}}\left(V^\top\left(X^p + \frac{1}{\sigma^p}Z^p\right)V\right)$;
        $X^{p+1} = \mathcal{P}_{\mathcal{X}_\mathcal{T}}\left(VR^{p+1}V^\top - \frac{1}{\sigma^p}\left(\frac{1}{2}L + Z^p\right)\right)$ using (CycDyk);
        $Z^{p+1} = Z^p + \sigma^p(X^{p+1} - VR^{p+1}V^\top)$;
        Update $\sigma^{p+1}$;
        $p \leftarrow p + 1$;
    end
    Compute a valid lower bound $lb(Z^p)$ by post-processing ;
    Identify the violated inequalities and add the numCuts most violated cuts to $\mathcal{T}$;
end
```

Numerical Results

graph	n	k	ub	lb_{DH}[2]	lb_{DNN}	$lb_{DNN+Cuts}$
mesh.70.120	70	2	7	1.93	2.91	6.02
KKT.lowt01	82	2	13	2.47	4.88	12.43
mesh.148.265	148	4	22	5.46	8.13	21.23
$G_{124,2.5}$	124	2	13	4.59	7.29	12.01
$G_{124,10}$	124	2	178	138.24	152.86	170.88
$G_{124,20}$	124	2	449	403.08	418.67	439.96
$G_{250,2.5}$	250	2	29	10.99	15.16	28.30
$G_{250,5}$	250	2	114	70.21	81.52	105.00
$G_{250,10}$	250	2	357	280.25	303.02	330.40

Table 1: Comparison between different relaxations

[2] Wilm E. Donath and Alan J. Hoffman (1973). "Lower bounds for the partitioning of graphs". In: *Ibm Journal of Research and Development* 17, pp. 420–425.

Numerical Results

graph	ub	lb_{DNN}	CPU(s)	$lb_{DNN+Cuts}$	Imp.	CPU(s)	numCut
$G_{500,2.5}$	49	24.89	266.625	44.38	80.00%	4609.69	25000
$G_{500,5}$	218	155.58	133.03	196.61	26.68%	2144.08	25000
$G_{500,10}$	626	512.13	94.23	553.43	7.99%	567.77	13782
$G_{500,20}$	1744	1565.59	86.75	1612.89	3.00%	192.02	10781

Table 2: Computational results on large instances with $n = 500, k = 2$ [1]

graph	ub	lb_{DNN}	CPU(s)	$lb_{DNN+Cuts}$	Imp.	CPU(s)	numCut
$G_{1000,2.5}$	102	44.29	2091.5	73.33	64.44%	21443.89	45000
$G_{1000,5}$	451	306.24	1009	378.98	23.45%	6977.61	50000
$G_{1000,10}$	1367	1112.76	742.94	1178.94	5.93%	1947.53	26685
$G_{1000,20}$	3389	3006.96	683.25	3078.70	2.39%	1311.66	21008

Table 3: Computational results on large instances with $n = 1000, k = 2$ [1]

[1] $G_{|V|,|V|p}$: graphs $G(V, E)$, with $|V| \in \{500, 1000\}$ and four individual edge probabilities p.

The full results are included in

Frank de Meijer, Renata Sotirov, Angelika Wiegele, and Shudian Zhao (2022). “Partitioning through projections: strong SDP bounds for large graph partition problems”. http://arxiv.org/abs/2205.06788

What else . . .

1. Implementation details: Clustering methods and warm-starting can help speed up Dykstra’s projection;
2. Further application: The variant of this framework can solve other graph partition problems, e.g., bisection problems.

Reference I

- Boyle, James P. and Richard L. Dykstra (1985). "A method for finding projections onto the intersection of convex sets in Hilbert spaces". In: *Advances in Order Restricted Statistical Inference, Lecture Notes in Statistics*. Ed. by R. Dykstra, T. Robertson, and F. T. Wright. Vol. 37. Springer.
- Cerulli, Martina, Marianna De Santis, Elisabeth Gaar, and Angelika Wiegele (2021). "Improving ADMMs for solving doubly nonnegative programs through dual factorization". In: *4OR* 19.3, pp. 415–448. DOI: 10.1007/s10288-020-00454-x. URL: https://doi.org/10.1007/s10288-020-00454-x.
- Donath, Wilm E. and Alan J. Hoffman (1973). "Lower bounds for the partitioning of graphs". In: *Ibm Journal of Research and Development* 17, pp. 420–425.
- Hu, Hao, Renata Sotirov, and Henry Wolkowicz (2019). "Facial Reduction for Symmetry Reduced Semidefinite Doubly Nonnegative Programs". https://arxiv.org/abs/1912.10245.

Reference II

- Li, Xinxin, Ting Kei Pong, Hao Sun, and H. Wolkowicz (2021). “A strictly contractive Peaceman-Rachford splitting method for the doubly nonnegative relaxation of the minimum cut problem”. In: *Comput. Optim. Appl.* 78, pp. 853–891.
- Lorenz, Dirk A and Quoc Tran-Dinh (2018). “Non-stationary Douglas-Rachford and alternating direction method of multipliers: adaptive stepsizes and convergence”. In: *arXiv preprint arXiv:1801.03765*.
- Malick, Jérôme, Janez Povh, Franz Rendl, and Angelika Wiegele (2009). “Regularization methods for semidefinite programming”. In: *SIAM Journal on Optimization* 20.1, pp. 336–356.
- de Meijer, Frank and Renata Sotirov (2021). “SDP-Based Bounds for the Quadratic Cycle Cover Problem via Cutting-Plane Augmented Lagrangian Methods and Reinforcement Learning”. In: *INFORMS Journal on Computing* 33.4, pp. 1262–1276.
- de Meijer, Frank, Renata Sotirov, Angelika Wiegele, and Shudian Zhao (2022). “Partitioning through projections: strong SDP bounds for large graph partition problems”. http://arxiv.org/abs/2205.06788.

Reference III

Oliveira, Danilo Elias, Henry Wolkowicz, and Yangyang Xu (2018). "ADMM for the SDP relaxation of the QAP". In: *Mathematical Programming Computation* 10.4, pp. 631–658.

Sun, Defeng, Kim-Chuan Toh, Yancheng Yuan, and Xin-Yuan Zhao (2020). "SDPNAL+: A Matlab software for semidefinite programming with bound constraints (version 1.0)". In: *Optim. Methods Softw.* 35.1, pp. 87–115.

Wen, Zaiwen, Donald Goldfarb, and Wotao Yin (2010). "Alternating direction augmented Lagrangian methods for semidefinite programming". In: *Mathematical Programming Computation* 2.3-4, pp. 203–230.

Wiegele, Angelika and Shudian Zhao (2022). "SDP-based bounds for graph partition via extended ADMM". In: *Computational Optimization and Applications*. DOI: 10.1007/s10589-022-00355-1.

Zhao, Shudian (2022). "Splitting into Pieces: Alternating Direction Methods of Multipliers and Graph Partitioning". PhD thesis. Alpen-Adria-Universität Klagenfurt.

Appendix

- Stopping criteria for the inner loop of CP–ADMM:

$$\max\left\{\frac{\|X^p - VR^pV^\top\|_F}{1+\|X^p\|_F}, \sigma\frac{\|X^{p+1}-X^p\|_F}{1+\|Z^p\|_F}\right\} < \varepsilon_{\mathrm{ADMM}},$$

  where $\varepsilon_{\mathrm{ADMM}}$ is the prescribed tolerance precision.

- The adaptive stepsize as introduced in (Lorenz and Tran-Dinh, 2018).

$$\sigma^{p+1} := (1-\omega^{p+1})\sigma^p + \omega^{p+1}\mathcal{P}_{[\sigma_{\min},\sigma_{\max}]}\frac{\|Z^{p+1}\|_F}{\|X^{p+1}\|_F}, \tag{5}$$

  where $\omega^{p+1} := 2^{-p/100}$ is the weight, $\sigma_{\min}$ and $\sigma_{\max}$ are the box bounds for σ^p, and $\mathcal{P}_{[\sigma_{\min},\sigma_{\max}]}$ is the projection onto $[\sigma_{\min},\sigma_{\max}]$.

- Valid lower bound: For any $Z \in \mathcal{S}^q$ one can obtain a valid lower bound by computing:

$$lb(Z) = \min_{X\in\mathcal{X}_\mathcal{T}} \langle\frac{1}{2}L + Z, X\rangle - \mathrm{trace}(R)\lambda_{\max}(V^\top ZV). \tag{6}$$

  Since the minimization problem above is a linear programming problem, we compute valid lower bounds efficiently.