January 26, 2023

An Exact Solver for QUBO Problems using the Mixing Method

Joint work with Valentin Durante

FWF Der Wissenschaftsfonds.

UNIVERSITÄT KLAGENFURT

Modeling - Analysis - Optimization DOC.FUNDS DOCTORAL SCHOOL

Quadratic Unconstrained Binary Optimization (QUBO)

- goal: branch-and-bound solver for

QUBO in $\{-1, 1\}$-variables

Given $C \in \mathbb{R}^{n \times n}$, solve

$$\begin{aligned} \max \quad & x^\top C x \\ \text{s.t.} \quad & x \in \{-1, 1\}^n. \end{aligned} \tag{QUBO}$$

Quadratic Unconstrained Binary Optimization (QUBO)

- goal: branch-and-bound solver for

QUBO in $\{-1,1\}$-variables

Given $C \in \mathbb{R}^{n\times n}$, solve

$$\begin{aligned} \max \quad & x^\top C x \\ \text{s.t.} \quad & x \in \{-1,1\}^n. \end{aligned} \tag{QUBO}$$

- $\mathcal{NP}$-hard
- LP based approaches exist only for sparse C
- we want to tackle QUBO problems with dense C

Quadratic Unconstrained Binary Optimization (QUBO)

- goal: branch-and-bound solver for

QUBO in $\{-1, 1\}$-variables

Given $C \in \mathbb{R}^{n \times n}$, solve

$$\begin{aligned} \max \quad & x^\top C x \\ \text{s.t.} \quad & x \in \{-1, 1\}^n. \end{aligned} \qquad \text{(QUBO)}$$

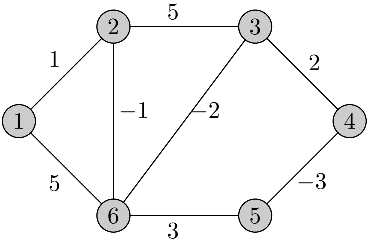
- $\mathcal{NP}$-hard
- LP based approaches exist only for sparse C
- we want to tackle QUBO problems with dense C

Example

Max-Cut Problem: $C = \frac{1}{4} L(G)$, where $L(G)$ Laplacian matrix

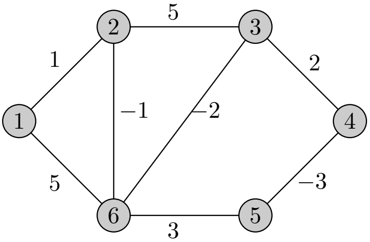
The (Weighted) Max-Cut Problem

Given: undirected graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E$



The (Weighted) Max-Cut Problem

Given: undirected graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E$



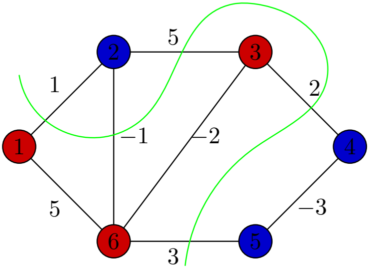
Max-Cut Problem

Find a maximum cut in G, i.e., an optimal solution of

$$\max_{S \subseteq V} \sum_{i \in S,\ j \in V \setminus S} w_{ij}. \tag{MC}$$

The (Weighted) Max-Cut Problem

Given: undirected graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E$



Max-Cut Problem

Find a maximum cut in G, i.e., an optimal solution of

$$\max_{S \subseteq V} \sum_{i \in S,\ j \in V \setminus S} w_{ij}. \tag{MC}$$

(QUBO) is quite general...

- minimization $\leftrightarrow$ maximization
- linear quadratic objective $x^\top Qx + q^\top x$
- variables in $\{0,1\}^n \leftrightarrow \{-1,1\}^n$
- linear constraints $Ax = b$

(QUBO) is quite general...

- minimization $\leftrightarrow$ maximization
- linear quadratic objective $x^\top Qx + q^\top x$
- variables in $\{0,1\}^n \leftrightarrow \{-1,1\}^n$
- linear constraints $Ax = b$

Linearly constrained binary quadratic problems

$$\begin{aligned} \min \quad & x^\top Qx + q^\top x \\ \text{s. t.} \quad & Ax = b \\ & x \in \{0,1\}^n \end{aligned} \tag{BQP}$$

where $Q \in \mathbb{R}^{n\times n}$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$.

(QUBO) is quite general...

- minimization $\leftrightarrow$ maximization
- linear quadratic objective $x^\top Qx + q^\top x$
- variables in $\{0,1\}^n \leftrightarrow \{-1,1\}^n$
- linear constraints $Ax = b$

Linearly constrained binary quadratic problems

$$\begin{aligned} \min \quad & x^\top Qx + q^\top x \\ \text{s.t.} \quad & Ax = b \\ & x \in \{0,1\}^n \end{aligned} \tag{BQP}$$

where $Q \in \mathbb{R}^{n\times n}$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$.

- Any BQP instance in n variables can be reformulated as a QUBO instance in $n+1$ variables!

Example: Exact Penalty Function

- undirected, simple graph $G = (V, E)$ with $|V| = n$

Maximum Stable Set Problem

$$\begin{aligned} \max \quad & e^\top x \\ \text{s.t.} \quad & x_i x_j = 0, \quad \forall ij \in E \\ & x \in \{0,1\}^n \end{aligned} \tag{MSSP}$$

Example: Exact Penalty Function

- undirected, simple graph $G = (V, E)$ with $|V| = n$

Maximum Stable Set Problem

$$\begin{aligned} \max \quad & e^\top x \\ \text{s. t.} \quad & x_i x_j = 0, \quad \forall ij \in E \\ & x \in \{0,1\}^n \end{aligned} \tag{MSSP}$$

Reformulation of (MSSP)

$$\begin{aligned} \max \quad & \left\{ \frac{n}{2} + \frac{1}{2} e^\top x - n \sum_{ij \in E} (x_i + 1)(x_j + 1) \right\} \\ \text{s. t.} \quad & x \in \{-1, 1\}^n \end{aligned}$$

Semidefinite Relaxation of (QUBO)

We introduce $X := xx^\top$:

- $x^\top C x = \langle C, xx^\top \rangle = \langle C, X \rangle$
- $\mathrm{diag}(X) = e$
- $X \succeq 0$
- $\mathrm{rank}(X) = 1$

Semidefinite Relaxation of (QUBO)

We introduce $X := xx^\top$:

- $x^\top Cx = \langle C, xx^\top \rangle = \langle C, X \rangle$
- $\operatorname{diag}(X) = e$
- $X \succeq 0$
- $\operatorname{rank}(X) = 1$

Equivalent formulations

$$\begin{aligned} \max \quad & x^\top Cx \\ \text{s. t.} \quad & x \in \{-1, 1\}^n \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \max \quad & \langle C, X \rangle \\ \text{s. t.} \quad & \operatorname{diag}(X) = e \\ & X \succeq 0 \\ & \operatorname{rank}(X) = 1 \end{aligned}$$

Semidefinite Relaxation of (QUBO)

We introduce $X := xx^\top$:

- $x^\top Cx = \langle C, xx^\top \rangle = \langle C, X \rangle$
- $\operatorname{diag}(X) = e$
- $X \succeq 0$
- $\operatorname{rank}(X) = 1$

Semidefinite relaxation (SDP)

$$\begin{array}{ll} \max & x^\top Cx \\ \text{s.t.} & x \in \{-1, 1\}^n \end{array} \quad \leq \quad \begin{array}{ll} \max & \langle C, X \rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \\ & \sout{\operatorname{rank}(X) = 1} \end{array}$$

Semidefinite Relaxation of (QUBO)

We introduce $X := xx^\top$:

- $x^\top Cx = \langle C, xx^\top \rangle = \langle C, X \rangle$
- $\operatorname{diag}(X) = e$
- $X \succeq 0$
- $\operatorname{rank}(X) = 1$

Semidefinite relaxation (SDP)

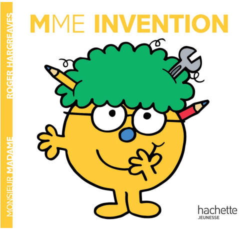
$$\begin{array}{rl} \max & x^\top Cx \\ \text{s.t.} & x \in \{-1,1\}^n \end{array} \quad \leq \quad \begin{array}{rl} \max & \langle C, X \rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \\ & \cancel{\operatorname{rank}(X) = 1} \end{array}$$

All solvers in the literature use additional 'clique' inequalities:

- `BiqMac` (2010)
- `MADAM` (2021)
- `BiqCrunch` (2016)
- `BiqBin` (2022)

(QUBO) Solvers using Semidefinite Programming

(QUBO) Solvers using Semidefinite Programming

BiqMac (2010)	BiqCrunch (2016)
	
MADAM (2021)	BiqBin (2022)
	

Low-rank Factorization

Factorization of $X \succeq 0$

$$X = V^\top V \succeq 0$$

for some $V = (v_1 | \dots | v_n) \in \mathbb{R}^{k \times n}$ with $k \leq n$.

Low-rank Factorization

Factorization of $X \succeq 0$

$$X = V^\top V \succeq 0$$

for some $V = (v_1 | \dots | v_n) \in \mathbb{R}^{k \times n}$ with $k \leq n$.

- $X_{ij} = v_i^\top v_j \quad \Rightarrow \quad \langle C, X \rangle = \sum_{i,j=1}^{n} C_{ij} X_{ij} = \sum_{i,j=1}^{n} C_{ij} v_i^\top v_j$
- $\mathrm{diag}(X) = e \quad \Leftrightarrow \quad \|v_i\| = 1, \ i = 1, \dots, n$

Low-rank Factorization

Factorization of $X \succeq 0$

$$X = V^\top V \succeq 0$$

for some $V = (v_1 | \dots | v_n) \in \mathbb{R}^{k \times n}$ with $k \leq n$.

- $X_{ij} = v_i^\top v_j \quad \Rightarrow \quad \langle C, X \rangle = \sum_{i,j=1}^n C_{ij} X_{ij} = \sum_{i,j=1}^n C_{ij} v_i^\top v_j$
- $\operatorname{diag}(X) = e \quad \Leftrightarrow \quad \|v_i\| = 1, \; i = 1, \dots, n$

Optimization problem (SDP-vec)

$$\begin{aligned} \max \quad & \sum_{i,j=1}^n C_{ij} v_i^\top v_j \\ \text{s. t.} \quad & \|v_i\| = 1, \; i = 1, \dots, n \end{aligned} \qquad \text{(SDP-vec)}$$

Low-rank Factorization

Factorization of $X \succeq 0$

$$X = V^\top V \succeq 0$$

for some $V = (v_1|\dots|v_n) \in \mathbb{R}^{k \times n}$ with $k \leq n$.

- $X_{ij} = v_i^\top v_j \quad \Rightarrow \quad \langle C, X\rangle = \sum_{i,j=1}^n C_{ij}X_{ij} = \sum_{i,j=1}^n C_{ij} v_i^\top v_j$
- $\operatorname{diag}(X) = e \quad \Leftrightarrow \quad \|v_i\| = 1, \; i = 1, \dots, n$

Optimization problem (SDP-vec)

$$\begin{aligned} \max \quad & \sum_{i,j=1}^n C_{ij} v_i^\top v_j \\ \text{s. t.} \quad & \|v_i\| = 1, \; i = 1, \dots, n \end{aligned} \qquad \text{(SDP-vec)}$$

- (SDP) $\Leftrightarrow$ (SDP-vec) for $k > \sqrt{2n}$ [cf. Pataki, 1998]

Coordinate Ascent Method

Optimization Problem (SDP-vec)

$$
\begin{aligned}
\max \quad & \sum_{i,j=1}^{n} C_{ij} v_i^\top v_j \\
\text{s.t.} \quad & \|v_i\| = 1, \; i = 1, \ldots, n
\end{aligned}
\tag{SDP-vec}
$$

Coordinate Ascent Method

Optimization Problem (SDP-vec)

$$\begin{aligned} \max \quad & \sum_{i,j=1}^{n} C_{ij} v_i^\top v_j \\ \text{s.t.} \quad & \|v_i\| = 1, \ i = 1, \dots, n \end{aligned} \qquad \text{(SDP-vec)}$$

Coordinate Ascent

We fix all but one column v_i. (SDP-vec) reduces to

$$\begin{aligned} \max \quad & g^\top v_i = \|g\| \cdot \|v_i\| \cdot \cos \measuredangle(g, v_i) \\ \text{s.t.} \quad & \|v_i\| = 1, \ v_i \in \mathbb{R}^k \end{aligned}$$

where $g = \sum_j^n c_{ij} v_j = V \cdot c_i$.

Coordinate Ascent Method

Optimization Problem (SDP-vec)

$$\begin{aligned} \max \quad & \sum_{i,j=1}^{n} C_{ij} v_i^\top v_j \\ \text{s. t.} \quad & \|v_i\| = 1, \; i = 1, \ldots, n \end{aligned} \qquad \text{(SDP-vec)}$$

Coordinate Ascent

We fix all but one column v_i. (SDP-vec) reduces to

$$\begin{aligned} \max \quad & g^\top v_i = \|g\| \cdot \|v_i\| \cdot \cos \measuredangle(g, v_i) \\ \text{s. t.} \quad & \|v_i\| = 1, \; v_i \in \mathbb{R}^k \end{aligned}$$

where $g = \sum_j^n c_{ij} v_j = V \cdot c_i$.

- closed-form solution: $v_i = \frac{g}{\|g\|}$ for $g \neq 0$

Algorithm: Mixing Method

Algorithm 1: Mixing Method (Wang et al., 2018)

Input: $C = (c_1|\dots|c_n) \in \mathbb{R}^{n\times n}$ with $\operatorname{diag}(C) = 0,\ k \in \mathbb{N}_{\geq 1}$
Output: approximate solution $V = (v_1|\dots|v_n) \in \mathbb{R}^{k\times n}$ of (SDP-vec)

for $i \leftarrow 1$ **to** n **do**
  $v_i \leftarrow$ random vector on the unit sphere $\mathcal{S}^{k-1}$;

while *not yet converged* **do**
  for $i \leftarrow 1$ **to** n **do**
    $v_i \leftarrow \frac{V\cdot c_i}{\|V\cdot c_i\|}$;

Algorithm: Mixing Method

Algorithm 1: Mixing Method (Wang et al., 2018)

Input: $C = (c_1|\ldots|c_n) \in \mathbb{R}^{n\times n}$ with $\operatorname{diag}(C) = 0,\ k \in \mathbb{N}_{\geq 1}$
Output: approximate solution $V = (v_1|\ldots|v_n) \in \mathbb{R}^{k\times n}$ of (SDP-vec)

for $i \leftarrow 1$ **to** n **do**
    $v_i \leftarrow$ random vector on the unit sphere $\mathcal{S}^{k-1}$;

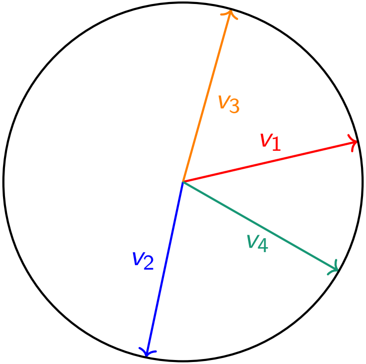
while *not yet converged* **do**
    for $i \leftarrow 1$ **to** n **do**
        $v_i \leftarrow \frac{V\cdot c_i}{\|V\cdot c_i\|}$;

Theorem (Wang et al., 2018)

The Mixing Method converges linearly to the global optimum under a non-degeneracy assumption.

Example with $n = 4$ and $k = 2$

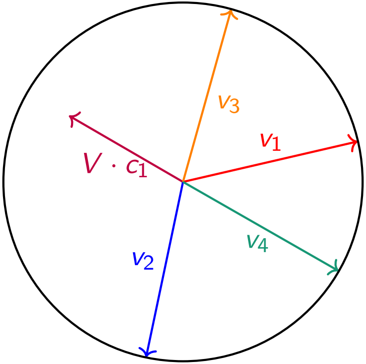
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V\rangle = -2.469151715641014$$

Example with $n = 4$ and $k = 2$

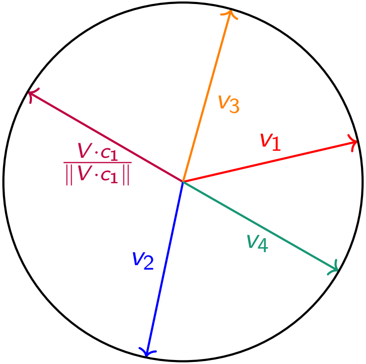
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = -2.469151715641014$$

Example with $n = 4$ and $k = 2$

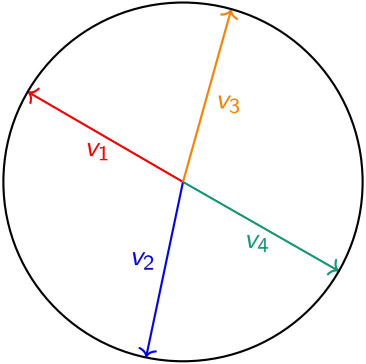
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = -2.469151715641014$$

Example with $n = 4$ and $k = 2$

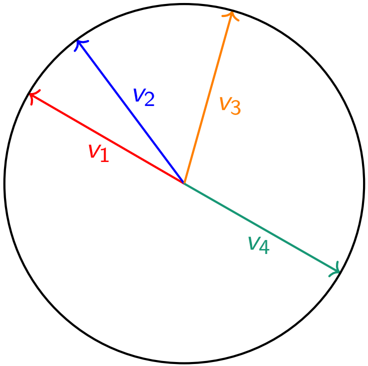
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 0.0701836938398076$$

Example with $n = 4$ and $k = 2$

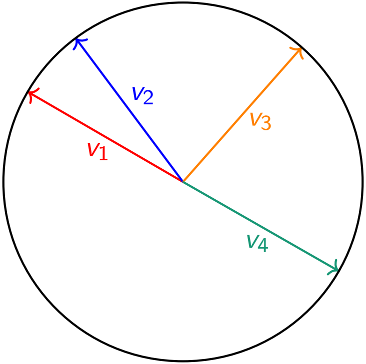
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.1042821481042009$$

Example with $n = 4$ and $k = 2$

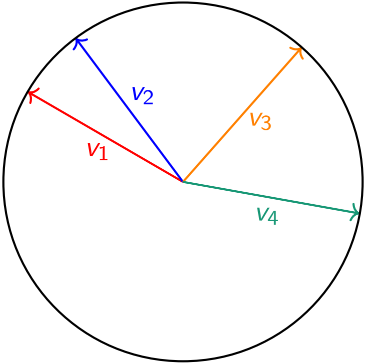
$$C = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.1248497956082537$$

Example with $n = 4$ and $k = 2$

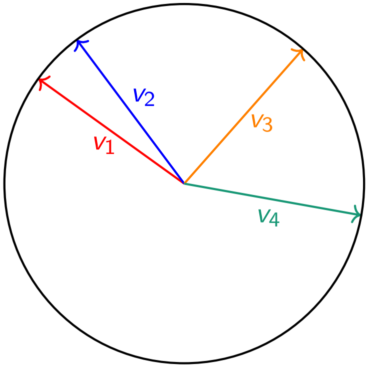
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V\rangle = 2.2584781813631301$$

Example with $n = 4$ and $k = 2$

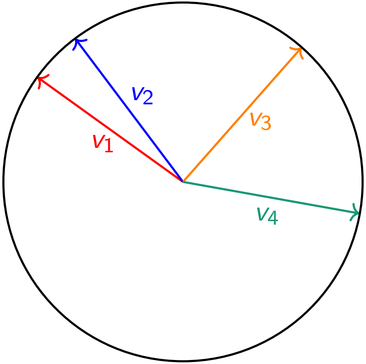
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2669613535505473$$

Example with $n = 4$ and $k = 2$

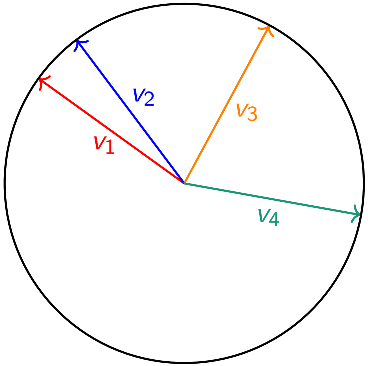
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2669669930002718$$

Example with $n = 4$ and $k = 2$

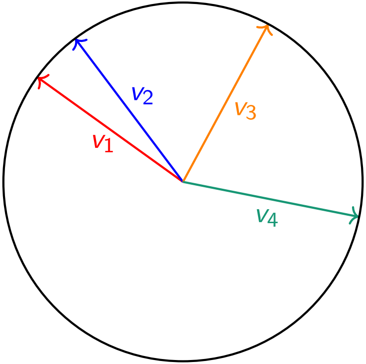
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2820426702215686$$

Example with $n = 4$ and $k = 2$

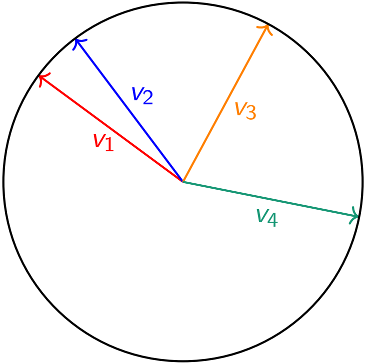
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V\rangle = 2.2824146853764495$$

Example with $n = 4$ and $k = 2$

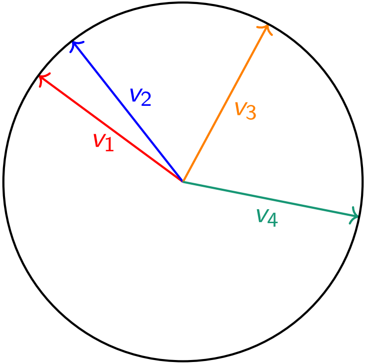
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2825485984904232$$

Example with $n = 4$ and $k = 2$

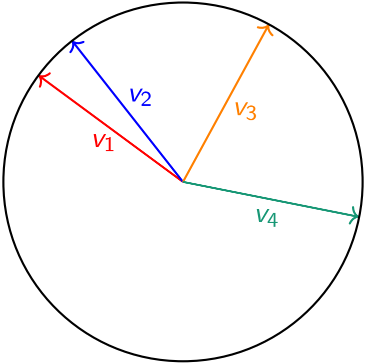
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2827921992397187$$

Example with $n = 4$ and $k = 2$

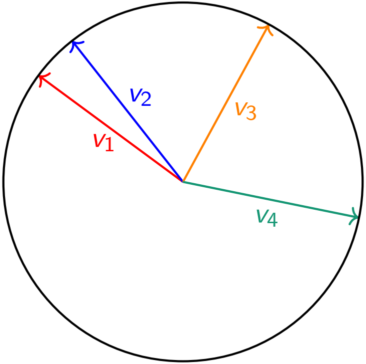
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2827965824488148$$

Example with $n = 4$ and $k = 2$

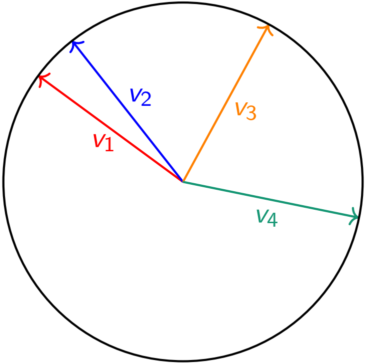
$$C = \frac{1}{4} \begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2828175664597827$$

Example with $n = 4$ and $k = 2$

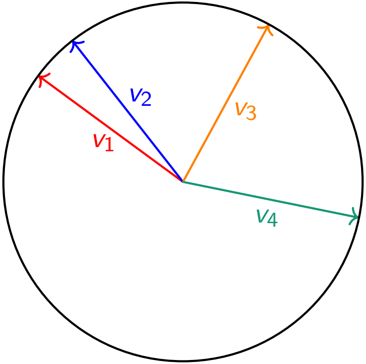
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2828214514872149$$

Example with $n = 4$ and $k = 2$

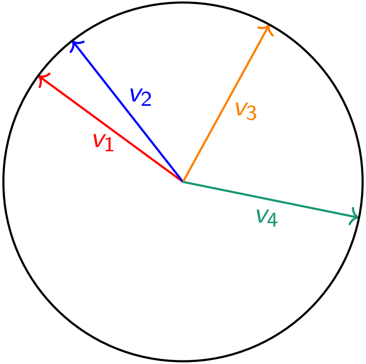
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V\rangle = 2.2828225671023645$$

Example with $n = 4$ and $k = 2$

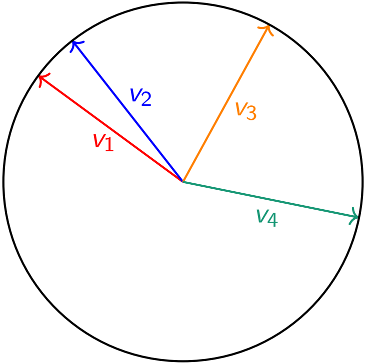
$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V\rangle = 2.2828245614424776$$

Example with $n = 4$ and $k = 2$

$$C = \frac{1}{4}\begin{pmatrix} 1 & 1 & 1 & -3 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -4 & 2 \\ -3 & -1 & 2 & 2 \end{pmatrix}$$



$$\langle C, V^\top V \rangle = 2.2828250454404815$$

Properties of the Mixing Method

Observations

- parameter-free and easy to implement
- objective value is strictly increasing
- produces primal feasible iterates
- warm start possible

Properties of the Mixing Method

Observations

- parameter-free and easy to implement
- objective value is strictly increasing
- produces primal feasible iterates
- warm start possible

But when do we stop?

Properties of the Mixing Method

Observations

- parameter-free and easy to implement
- objective value is strictly increasing
- produces primal feasible iterates
- warm start possible

But when do we stop?

Stopping criterion: relative step tolerance

- stop **if** $\frac{\|V_{\text{old}} - V_{\text{new}}\|_F}{1 + \|V_{\text{old}}\|_F} < \varepsilon$

Properties of the Mixing Method

Observations

- parameter-free and easy to implement
- objective value is strictly increasing
- produces primal feasible iterates
- warm start possible

But when do we stop?

Stopping criterion: relative step tolerance

- stop **if** $\frac{\|V_{\text{old}} - V_{\text{new}}\|_F}{1 + \|V_{\text{old}}\|_F} < \varepsilon$
- we use $\varepsilon = 0.013$

Properties of the Mixing Method

Observations

- parameter-free and easy to implement
- objective value is strictly increasing
- produces primal feasible iterates
- warm start possible

But when do we stop?

Stopping criterion: relative step tolerance

- stop **if** $\frac{\|V_{\text{old}} - V_{\text{new}}\|_F}{1 + \|V_{\text{old}}\|_F} < \varepsilon$
- we use $\varepsilon = 0.013$

How do we get an upper bound?

Upper Bounds via Weak Duality

Duality

$$\begin{aligned} \max \quad & \langle C, X\rangle \\ \text{s.t.} \quad & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{aligned} \tag{SDP}$$

$$\begin{aligned} \min \quad & e^\top y \\ \text{s.t.} \quad & \operatorname{Diag}(y) - C = Z \\ & Z \succeq 0,\ y \in \mathbb{R}^n \end{aligned} \tag{DSDP}$$

Upper Bounds via Weak Duality

Duality

$$\begin{array}{rl} \max & \langle C, X \rangle \\ \text{s. t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{array} \qquad \text{(SDP)} \qquad\qquad \begin{array}{rl} \min & e^\top y \\ \text{s. t.} & \operatorname{Diag}(y) - C = Z \\ & Z \succeq 0,\ y \in \mathbb{R}^n \end{array} \qquad \text{(DSDP)}$$

Proposition [Wang et al., 2018]

If V and $X = V^\top V$ are optimal for (SDP-vec) and (SDP), then the vector $y \in \mathbb{R}^n$ with entries $y_i = \|V \cdot c_i\|_2$ is optimal for (DSDP).

Upper Bounds via Weak Duality

Duality

$$\begin{array}{rl} \max & \langle C, X\rangle \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{array} \qquad \text{(SDP)}$$

$$\begin{array}{rl} \min & e^\top y \\ \text{s.t.} & \operatorname{Diag}(y) - C = Z \\ & Z \succeq 0, \; y \in \mathbb{R}^n \end{array} \qquad \text{(DSDP)}$$

Proposition [Wang et al., 2018]

If V and $X = V^\top V$ are optimal for (SDP-vec) and (SDP), then the vector $y \in \mathbb{R}^n$ with entries $y_i = \|V \cdot c_i\|_2$ is optimal for (DSDP).

After stopping the Mixing Method with approximate $\tilde{V}$:

- approximate but non-feasible dual variables: $\tilde{y}_i = \|\tilde{V} \cdot c_i\|_2$

Upper Bounds via Weak Duality

Duality

$$\begin{array}{ll} \max & \langle C, X\rangle \\ \text{s. t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{array} \qquad \text{(SDP)}$$

$$\begin{array}{ll} \min & e^\top y \\ \text{s. t.} & \operatorname{Diag}(y) - C = Z \\ & Z \succeq 0, \; y \in \mathbb{R}^n \end{array} \qquad \text{(DSDP)}$$

Proposition [Wang et al., 2018]

If V and $X = V^\top V$ are optimal for (SDP-vec) and (SDP), then the vector $y \in \mathbb{R}^n$ with entries $y_i = \|V \cdot c_i\|_2$ is optimal for (DSDP).

After stopping the Mixing Method with approximate $\tilde{V}$:

- approximate but non-feasible dual variables: $\tilde{y}_i = \|\tilde{V} \cdot c_i\|_2$
- feasible dual variables: $y = \tilde{y} - \lambda_{\min}\left(\operatorname{Diag}(\tilde{y}) - C\right) e$

Other Possibility

We use the dual bound

$$e^\top \tilde{y} - n\lambda_{\min}\left(\mathrm{Diag}(\tilde{y}) - C\right).$$

Other Possibility

We use the dual bound

$$e^\top \tilde{y} - n\lambda_{\min}\left(\mathrm{Diag}(\tilde{y}) - C\right).$$

Better upper bound [Jansson et al., 2007]

Let $\tilde{y} \in \mathbb{R}^n$ and $\bar{x}$ such that $\lambda_{\max}(X) \leq \bar{x}$ for some optimal X of (SDP). Then

$$e^\top \tilde{y} - \sum_{\lambda_k(\mathrm{Diag}(\tilde{y}) - C) < 0} \lambda_k \bar{x}$$

is an upper bound on (SDP).

Other Possibility

We use the dual bound

$$e^\top \tilde{y} - n\lambda_{\min}\left(\mathrm{Diag}(\tilde{y}) - C\right).$$

Better upper bound [Jansson et al., 2007]

Let $\tilde{y} \in \mathbb{R}^n$ and $\bar{x}$ such that $\lambda_{\max}(X) \leq \bar{x}$ for some optimal X of (SDP). Then

$$e^\top \tilde{y} - \sum_{\lambda_k(\mathrm{Diag}(\tilde{y}) - C) < 0} \lambda_k \bar{x}$$

is an upper bound on (SDP).

- slightly better bounds
- more expensive

Primal Heuristic

Algorithm 2: Goemans-Williamson hyperplane rounding

Input: $V = (v_1 | \dots | v_n) \in \mathbb{R}^{k \times n}$ (such that $V^\top V = X$)
Output: $x \in \{-1, 1\}^n$

$h \leftarrow$ random vector on the unit sphere $\mathcal{S}^{k-1}$;
for $i \leftarrow 1$ **to** n **do**

$$x_i \leftarrow \begin{cases} +1, & \text{if } h^\top v_i \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

return x;

Primal Heuristic

Algorithm 2: Goemans-Williamson hyperplane rounding

Input: $V = (v_1 | \dots | v_n) \in \mathbb{R}^{k \times n}$ (such that $V^\top V = X$)
Output: $x \in \{-1, 1\}^n$

$h \leftarrow$ random vector on the unit sphere $\mathcal{S}^{k-1}$;
for $i \leftarrow 1$ **to** n **do**

$$x_i \leftarrow \begin{cases} +1, & \text{if } h^\top v_i \geq 0 \\ -1, & \text{otherwise} \end{cases}$$

return x;

- local search to improve the solution (one-opt and two-opt)
- detect reasonable candidates for local search
- use a 'good'/biased hyperplane

Branch-and-Bound Algorithm

Branching:

- branching on products $X_{ij} \in \{-1, 1\}$
- branch on (i, j) where sum of dual variables is large
- best-first search (largest upper bound)

Branch-and-Bound Algorithm

Branching:

- branching on products $X_{ij} \in \{-1, 1\}$
- branch on (i, j) where sum of dual variables is large
- best-first search (largest upper bound)

Bounding:

- primal (lower) bounds via heuristics
- dual (upper) bounds via weak duality and postprocessing

Branch-and-Bound Algorithm

Branching:

- branching on products $X_{ij} \in \{-1, 1\}$
- branch on (i, j) where sum of dual variables is large
- best-first search (largest upper bound)

Bounding:

- primal (lower) bounds via heuristics
- dual (upper) bounds via weak duality and postprocessing

Features:

- early branching
- variable fixing

Branching Example

$$C = \begin{pmatrix} 2 & -1 & 3 & -2 \\ -1 & -1 & 1 & 2 \\ 3 & 1 & 1 & -1 \\ -2 & 2 & -1 & 1 \end{pmatrix}$$

Branching on $(2,3)$ with $X_{23} = x_2 \cdot x_3 = 1$:

$$\begin{pmatrix} 2 & -1+3 & 3 & -2 \\ -1+3 & -1+1+2\cdot 1 & 1 & 2-1 \\ 3 & 1 & 1 & -1 \\ -2 & 2-1 & -1 & 1 \end{pmatrix} \xRightarrow[\text{row/column 3}]{\text{remove}} C' = \begin{pmatrix} 2 & 2 & -2 \\ 2 & 2 & 1 \\ -2 & 1 & 1 \end{pmatrix}$$

Branching on $(2,3)$ with $X_{23} = x_2 \cdot x_3 = -1$:

$$\begin{pmatrix} 2 & -1-3 & 3 & -2 \\ -1-3 & -1+1-2\cdot 1 & 1 & 2+1 \\ 3 & 1 & 1 & -1 \\ -2 & 2+1 & -1 & 1 \end{pmatrix} \xRightarrow[\text{row/column 3}]{\text{remove}} C' = \begin{pmatrix} 2 & -4 & -2 \\ -4 & -2 & 3 \\ -2 & 3 & 1 \end{pmatrix}$$

Branching Decision

- SDP approaches in literature only use X for branching decision
  - often: branching on most fractional variable
  - some solvers branch in first row/column only

Branching Decision

- SDP approaches in literature only use X for branching decision
  - often: branching on most fractional variable
  - some solvers branch in first row/column only

Branching decision based on dual variables

We determine the branching decision (i, j) in $\mathcal{O}(n)$:

Branching Decision

- SDP approaches in literature only use X for branching decision
  - often: branching on most fractional variable
  - some solvers branch in first row/column only

Branching decision based on dual variables

We determine the branching decision (i, j) in $\mathcal{O}(n)$:

1. Find $i = \operatorname{argmax}_k \{y_k\}$.

Branching Decision

- SDP approaches in literature only use X for branching decision
  - often: branching on most fractional variable
  - some solvers branch in first row/column only

Branching decision based on dual variables

We determine the branching decision (i, j) in $\mathcal{O}(n)$:

1. Find $i = \text{argmax}_k \{y_k\}$.
2. Find $j = \text{argmax}_k \{(y_i + y_k) \cdot f(X_{ik}) \colon |X_{ik}| \leq 0.875\}$.

- where $f \colon \{-1, 1\} \to [0, 1]$ decreasing in $|X_{ik}|$

Feature: Early Branching

Assumption

Finding an optimal solution with heuristics is easy.

Observation

The Mixing Method produces primal feasible iterates for (SDP).

Feature: Early Branching

Assumption

Finding an optimal solution with heuristics is easy.

Observation

The Mixing Method produces primal feasible iterates for (SDP).

Stopping criteria have an impact on:

- solutions found by heuristics (important for pruning)
- branching decision (important for overall efficiency)
- upper bound (important for pruning and best-first search)

Feature: Early Branching

Assumption

Finding an optimal solution with heuristics is easy.

Observation

The Mixing Method produces primal feasible iterates for (SDP).

Stopping criteria have an impact on:

- solutions found by heuristics (important for pruning)
- branching decision (important for overall efficiency)
- upper bound (important for pruning and best-first search)

Early branching

Immediately branch if we have done at least 4 iterations of the **while** loop and we know that the optimal value of (SDP) will be larger than the best known lower bound found by heuristics.

Feature: Variable Fixing

Given: Dual feasible solution $\mathrm{Diag}(y) - C \succeq 0$ for $C \in \mathbb{R}^{n \times n}$.

Notation

- $C_{/j}$ denotes matrix C without row j and column j.
- $y_{/j}$ denotes vector y without entry j.

Feature: Variable Fixing

Given: Dual feasible solution $\mathrm{Diag}(y) - C \succeq 0$ for $C \in \mathbb{R}^{n \times n}$.

Notation

- $C_{/j}$ denotes matrix C without row j and column j.
- $y_{/j}$ denotes vector y without entry j.

Branching on $(1, j)$ would yield cost matrix $\tilde{C} \in \mathbb{R}^{(n-1)\times(n-1)}$ with $C_{/j} - \tilde{C} = \begin{pmatrix} 0 & \delta^\top \\ \delta & 0 \end{pmatrix}$ for some $\delta \in \mathbb{R}^{n-2}$.

Feature: Variable Fixing

Given: Dual feasible solution $\mathrm{Diag}(y) - C \succeq 0$ for $C \in \mathbb{R}^{n \times n}$.

Notation

- $C_{/j}$ denotes matrix C without row j and column j.
- $y_{/j}$ denotes vector y without entry j.

Branching on $(1, j)$ would yield cost matrix $\tilde{C} \in \mathbb{R}^{(n-1)\times(n-1)}$ with $C_{/j} - \tilde{C} = \begin{pmatrix} 0 & \delta^\top \\ \delta & 0 \end{pmatrix}$ for some $\delta \in \mathbb{R}^{n-2}$.

Lemma

$\tilde{y} := y_{/j} + \begin{pmatrix} \|\delta\|_1 \\ |\delta_1| \\ \vdots \\ |\delta_{n-2}| \end{pmatrix}$ is dual feasible, i.e., $\mathrm{Diag}(\tilde{y}) - \tilde{C} \succeq 0$.

Proof.

$$\mathrm{Diag}(\tilde{y}) - \tilde{C} = \mathrm{Diag}\left(y_{/j} + \begin{pmatrix} \|\delta\|_1 \\ |\delta_1| \\ \vdots \\ |\delta_{n-2}| \end{pmatrix}\right) - \left(C_{/j} - \begin{pmatrix} 0 & \delta^\top \\ \delta & 0 \end{pmatrix}\right)$$

$$= \mathrm{Diag}\left(y_{/j}\right) + \mathrm{Diag}\left(\begin{pmatrix} \|\delta\|_1 \\ |\delta_1| \\ \vdots \\ |\delta_{n-2}| \end{pmatrix}\right) - C_{/j} + \begin{pmatrix} 0 & \delta^\top \\ \delta & 0 \end{pmatrix}$$

$$= \underbrace{\mathrm{Diag}\left(y_{/j}\right) - C_{/j}}_{\succeq 0} + \underbrace{\mathrm{Diag}\left(\begin{pmatrix} \|\delta\|_1 \\ |\delta_1| \\ \vdots \\ |\delta_{n-2}| \end{pmatrix}\right) + \begin{pmatrix} 0 & \delta^\top \\ \delta & 0 \end{pmatrix}}_{\succeq 0} \succeq 0$$

□

Variable Fixing

- bound at current node: $e^\top y$

'Free' dual bound if we would branch

Dual bound after branching on (i, j): $e^\top \tilde{y} + 2\|\delta\|_1 \pm 2c_{ij}$.

- difference of bounds: $-y_j + 2\sum_{k \neq i,j} |c_{jk}| \pm 2c_{ij}$

Variable Fixing

- bound at current node: $e^\top y$

'Free' dual bound if we would branch

Dual bound after branching on (i, j): $e^\top \tilde{y} + 2\|\delta\|_1 \pm 2c_{ij}$.

- difference of bounds: $-y_j + 2\sum_{k \neq i,j} |c_{jk}| \pm 2c_{ij}$
- best scenario: 'free' dual bound worse than best known primal bound

Variable Fixing

- bound at current node: $e^\top y$

'Free' dual bound if we would branch

Dual bound after branching on (i,j): $e^\top \tilde{y} + 2\|\delta\|_1 \pm 2c_{ij}$.

- difference of bounds: $-y_j + 2\sum_{k\neq i,j} |c_{jk}| \pm 2c_{ij}$
- best scenario: 'free' dual bound worse than best known primal bound

How we use it

- check all $\mathcal{O}(n^2)$ candidates in $\mathcal{O}(n^2)$ time
- do usual branching step + additional fixation(s)

Variable Fixing

- bound at current node: $e^\top y$

'Free' dual bound if we would branch

Dual bound after branching on (i,j): $e^\top \tilde{y} + 2\|\delta\|_1 \pm 2c_{ij}$.

- difference of bounds: $-y_j + 2\sum_{k\neq i,j} |c_{jk}| \pm 2c_{ij}$
- best scenario: 'free' dual bound worse than best known primal bound

How we use it

- check all $\mathcal{O}(n^2)$ candidates in $\mathcal{O}(n^2)$ time
- do usual branching step + additional fixation(s)

Issue

Conflict with early branching (no dual feasible solution)!

Preliminary Results

- C implementation using Intel MKL
- tested on instances from the *BiqMac Library* with $n \leq 100$

Results

- 100–1000 times more subproblems than other approaches
- 2–10 times faster than the best approach in the literature

Preliminary Results

- C implementation using Intel MKL
- tested on instances from the *BiqMac Library* with $n \leq 100$

Results

- 100–1000 times more subproblems than other approaches
- 2–10 times faster than the best approach in the literature

Current goal: including *triangle inequalities*

$$
\begin{aligned}
X_{ij} + X_{ik} + X_{jk} &\geq -1, \quad i < j < k \\
X_{ij} - X_{ik} - X_{jk} &\geq -1, \quad i < j < k \\
-X_{ij} + X_{ik} - X_{jk} &\geq -1, \quad i < j < k \\
-X_{ij} - X_{ik} + X_{jk} &\geq -1, \quad i < j < k
\end{aligned}
$$

Lagrangian Relaxation

(SDP) with triangle inequalities $\langle A_i, X \rangle \leq b_i,\ i = 1, \ldots, m$:

$$\begin{aligned} \max \quad & \langle C, X \rangle \\ \text{s.t.} \quad & \operatorname{diag}(X) = e \\ & \mathcal{A}(X) \leq b \\ & X \succeq 0. \end{aligned}$$

Lagrangian Relaxation

(SDP) with triangle inequalities $\langle A_i, X\rangle \le b_i,\ i = 1, \ldots, m$:

$$\begin{aligned} \max \quad & \langle C, X\rangle \\ \text{s. t.} \quad & \operatorname{diag}(X) = e \\ & \mathcal{A}(X) \le b \\ & X \succeq 0. \end{aligned}$$

After dualizing the constraints $\mathcal{A}(X) \le b$, we have to solve

$$\min_{y \ge 0} \left\{ b^\top y + \max_{\substack{\operatorname{diag}(X)=e \\ X \succeq 0}} \left\{ \langle C - \mathcal{A}^\top(y), X\rangle \right\} \right\}. \qquad (*)$$

Lagrangian Relaxation

(SDP) with triangle inequalities $\langle A_i, X \rangle \leq b_i, \; i = 1, \ldots, m$:

$$\begin{array}{rl} \max & \langle C, X \rangle \\ \text{s. t.} & \operatorname{diag}(X) = e \\ & \mathcal{A}(X) \leq b \\ & X \succeq 0. \end{array}$$

After dualizing the constraints $\mathcal{A}(X) \leq b$, we have to solve

$$\min_{y \geq 0} \left\{ b^\top y + \max_{\substack{\operatorname{diag}(X)=e \\ X \succeq 0}} \left\{ \langle C - \mathcal{A}^\top(y), X \rangle \right\} \right\}. \qquad (*)$$

- Mixing Method can be used for the inner problem
- problem $(*)$ is nonsmooth
- $b - \mathcal{A}(X^*)$ is subgradient

Lagrangian Relaxation

(SDP) with triangle inequalities $\langle A_i, X\rangle \le b_i,\ i = 1, \ldots, m$:

$$\begin{aligned} \max \quad & \langle C, X\rangle \\ \text{s. t.} \quad & \operatorname{diag}(X) = e \\ & \mathcal{A}(X) \le b \\ & X \succeq 0. \end{aligned}$$

After dualizing the constraints $\mathcal{A}(X) \le b$, we have to solve

$$\min_{y\ge 0} \left\{ b^\top y + \max_{\substack{\operatorname{diag}(X)=e \\ X\succeq 0}} \left\{ \langle C - \mathcal{A}^\top(y), X\rangle \right\} \right\}. \qquad (*)$$

- Mixing Method can be used for the inner problem
- problem $(*)$ is nonsmooth
- $b - \mathcal{A}(X^*)$ is subgradient

Thank you!