July 25, 2024

Exploiting low-rank SDP methods for solving Max-Cut

Joint work with Valentin Durante, Federal University of Toulouse

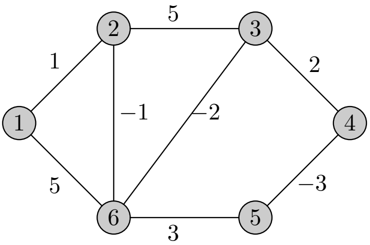
FWF Der Wissenschaftsfonds.

UNIVERSITÄT KLAGENFURT

Modeling - Analysis - Optimization DOC.FUNDS DOCTORAL SCHOOL

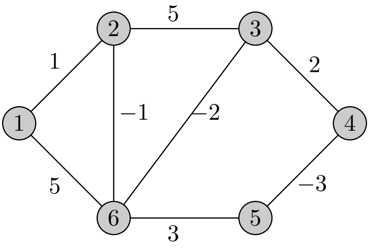
The (weighted) Max-Cut Problem

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



The (weighted) Max-Cut Problem

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



Definition: induced cut

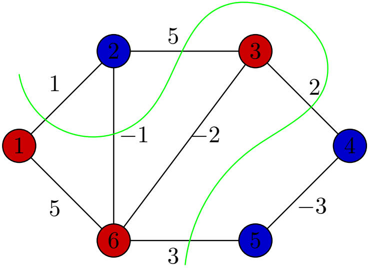
For $S \subseteq V$, the set of edges

$$\delta(S) \coloneqq \{ij \in E \colon i \in S,\ j \notin S\}$$

is called the *cut* induced by S.

The (weighted) Max-Cut Problem

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



Definition: induced cut

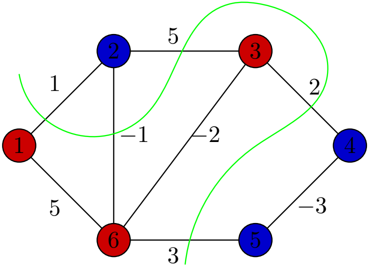
For $S \subseteq V$, the set of edges

$$\delta(S) := \{ij \in E \colon i \in S,\ j \notin S\}$$

is called the *cut* induced by S.

The (weighted) Max-Cut Problem

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



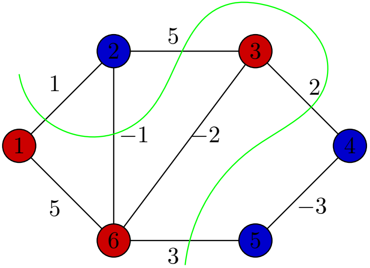
Max-Cut Problem

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

$$\max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij}. \tag{MC}$$

The (weighted) Max-Cut Problem

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



Max-Cut Problem

- $\mathcal{NP}$-hard
- polynomial time solvable in special cases (e.g., planar graphs)
- 0.878-approximation algorithm for $a \geq 0$ (Goemans & Williamson, 1995) (Mahajan & Ramesh, 1995)
- LP-based approaches efficient for sparse graphs

Quadratic unconstrained binary optimization (QUBO)

- Laplacian matrix $L := \mathrm{Diag}(Ae) - A$
  - weighted adjacency matrix $A = (a_{ij})_{ij}$
  - all-ones vector e

Formulation of Max-Cut

$$(\mathrm{MC}) \Leftrightarrow \begin{array}{ll} \max & \frac{1}{4} x^\top L x \\ \text{s.t.} & x \in \{-1, 1\}^n \end{array}$$

Quadratic unconstrained binary optimization (QUBO)

- Laplacian matrix $L := \mathrm{Diag}(Ae) - A$
  - weighted adjacency matrix $A = (a_{ij})_{ij}$
  - all-ones vector e

Formulation of Max-Cut

$$(\text{MC}) \Leftrightarrow \begin{array}{ll} \max & \frac{1}{4} x^\top L x \\ \text{s. t.} & x \in \{-1, 1\}^n \end{array}$$

Quadratic unconstrained binary optimization

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

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

Quadratic unconstrained binary optimization (QUBO)

- Laplacian matrix $L := \mathrm{Diag}(Ae) - A$
  - weighted adjacency matrix $A = (a_{ij})_{ij}$
  - all-ones vector e

Formulation of Max-Cut

$$(\mathrm{MC}) \Leftrightarrow \quad \begin{array}{ll} \max & \frac{1}{4} x^\top L x \\ \text{s.t.} & x \in \{-1, 1\}^n \end{array}$$

Quadratic unconstrained binary optimization

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

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

Goal: branch-and-cut solver for (MC) and (QUBO)

(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 $x^\top Q x + q^\top x$
- variables in $\{0,1\}^n \leftrightarrow \{-1,1\}^n$
- linear constraints $Ax = b$

Linearly constrained binary quadratic problems

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

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! (Lasserre, 2016)

Semidefinite programming relaxation

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

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

Semidefinite programming relaxation

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 (Laurent & Poljak, 1995)

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

Semidefinite programming relaxation

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 programming relaxation

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

Semidefinite programming relaxation

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 programming relaxation

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

Optimal value of SDP relaxation is at most...

- 57% larger if $C \succeq 0$. (Nesterov, 1998)
- 13.83% larger for (MC) if $a \geq 0$. (Goemans & Williamson, 1995)

Branch-and-cut approaches

- SDP-based solvers in the literature:
  - `BiqMac` (2010)
  - `MADAM` (2021)
  - `BiqCrunch` (2016)
  - `BiqBin` (2022)

Branch-and-cut approaches

- SDP-based solvers in the literature:
  - BiqMac (2010)
  - BiqCrunch (2016)
  - MADAM (2021)
  - BiqBin (2022)

- $\mathcal{O}(n^3)$ triangle inequalities:

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

Branch-and-cut approaches

- SDP-based solvers in the literature:
  - `BiqMac` (2010)
  - `MADAM` (2021)
  - `BiqCrunch` (2016)
  - `BiqBin` (2022)
- $\mathcal{O}(n^3)$ triangle inequalities:

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

- `MADAM` & `BiqBin`: $\mathcal{O}(n^5)$ pentagonal, $\mathcal{O}(n^7)$ heptagonal cuts
- exact separation only for triangle inequalities

Branch-and-cut approaches

- SDP-based solvers in the literature:
  - `BiqMac` (2010)
  - `MADAM` (2021)
  - `BiqCrunch` (2016)
  - `BiqBin` (2022)
- $\mathcal{O}(n^3)$ triangle inequalities:

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

- `MADAM` & `BiqBin`: $\mathcal{O}(n^5)$ pentagonal, $\mathcal{O}(n^7)$ heptagonal cuts
- exact separation only for triangle inequalities

Today: new solver called `MixCut`

Lagrangian relaxation

SDP with a subset of m triangle inequalities $\langle A_i, X \rangle \leq b_i$:

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

Lagrangian relaxation

SDP with a subset of m triangle inequalities $\langle A_i, X\rangle \leq b_i$:

$$
\begin{aligned}
f^* \;:=\; \max \quad & \langle C, X\rangle \\
\text{s. t.} \quad & X \in \mathcal{E} \quad (\Leftrightarrow \operatorname{diag}(X) = e,\; X \succeq 0) \\
& \mathcal{A}(X) \leq b
\end{aligned}
$$

Dualizing $\mathcal{A}(X) \leq b$ yields:

partial Lagrangian: $\mathcal{L}(X, \gamma) := \langle C, X\rangle - \gamma^\top(\mathcal{A}(X) - b)$

dual function: $f(\gamma) := \max_{X\in\mathcal{E}} \mathcal{L}(X, \gamma) = b^\top\gamma + \max_{X\in\mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X\rangle$

- adjoint operator: $\mathcal{A}^\top(\gamma) := \sum_{i=1}^m \gamma_i A_i$

Lagrangian relaxation

SDP with a subset of m triangle inequalities $\langle A_i, X\rangle \leq b_i$:

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

Dualizing $\mathcal{A}(X) \leq b$ yields:

partial Lagrangian: $\mathcal{L}(X,\gamma) := \langle C, X\rangle - \gamma^\top(\mathcal{A}(X) - b)$

dual function: $f(\gamma) := \max\limits_{X\in\mathcal{E}} \mathcal{L}(X,\gamma) = b^\top\gamma + \max\limits_{X\in\mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X\rangle$

- adjoint operator: $\mathcal{A}^\top(\gamma) := \sum_{i=1}^m \gamma_i A_i$
- weak duality: $f^* \leq f(\gamma)$ for all $\gamma \in \mathbb{R}^m_+$

Lagrangian relaxation

SDP with a subset of m triangle inequalities $\langle A_i, X \rangle \leq b_i$:

$$
\begin{aligned}
f^* \;:=\; \max \quad & \langle C, X \rangle \\
\text{s.t.} \quad & X \in \mathcal{E} \quad (\Leftrightarrow \operatorname{diag}(X) = e,\; X \succeq 0) \\
& \mathcal{A}(X) \leq b
\end{aligned}
$$

Dualizing $\mathcal{A}(X) \leq b$ yields:

partial Lagrangian: $\mathcal{L}(X, \gamma) := \langle C, X \rangle - \gamma^\top (\mathcal{A}(X) - b)$

dual function: $f(\gamma) := \max_{X \in \mathcal{E}} \mathcal{L}(X, \gamma) = b^\top \gamma + \max_{X \in \mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X \rangle$

- adjoint operator: $\mathcal{A}^\top(\gamma) := \sum_{i=1}^m \gamma_i A_i$
- weak duality: $f^* \leq f(\gamma)$ for all $\gamma \in \mathbb{R}^m_+$
- dual problem:

$$f^* = \min_{\gamma \geq 0} f(\gamma)$$

Evaluating f

$$f(\gamma) = b^\top \gamma + \max_{X \in \mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X \rangle$$

- for $\tilde{C} = C - \mathcal{A}^\top(\gamma)$, we have to solve

$$\begin{aligned} \max \quad & \langle \tilde{C}, X \rangle \\ \text{s.t.} \quad & X \in \mathcal{E} \end{aligned} \qquad (*)$$

- `BiqMac` & `BiqBin` use interior-point methods

Evaluating f

$$f(\gamma) = b^\top \gamma + \max_{X \in \mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X \rangle$$

- for $\tilde{C} = C - \mathcal{A}^\top(\gamma)$, we have to solve

$$\begin{aligned} \max \quad & \langle \tilde{C}, X \rangle \\ \text{s.t.} \quad & X \in \mathcal{E} \end{aligned} \qquad (*)$$

- `BiqMac` & `BiqBin` use interior-point methods

Burer-Monteiro factorization for SDPs (Burer & Monteiro, 2003)

Factorize $X = V^\top V \succeq 0, \ V = (v_1 | \dots | v_n) \in \mathbb{R}^{k \times n}, \ k \leq n$, and solve

$$\begin{aligned} \max \quad & \langle \tilde{C}, V^\top V \rangle \\ \text{s.t.} \quad & V^\top V \in \mathcal{E}. \end{aligned} \qquad \text{(SDP-vec)}$$

Evaluating f

$$f(\gamma) = b^\top \gamma + \max_{X \in \mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X \rangle$$

- for $\tilde{C} = C - \mathcal{A}^\top(\gamma)$, we have to solve

$$\begin{aligned} \max \quad & \langle \tilde{C}, X \rangle \\ \text{s. t.} \quad & X \in \mathcal{E} \end{aligned} \qquad (*)$$

- `BiqMac` & `BiqBin` use interior-point methods

Burer-Monteiro factorization for SDPs (Burer & Monteiro, 2003)

Factorize $X = V^\top V \succeq 0,\ V = (v_1 | \ldots | v_n) \in \mathbb{R}^{k \times n},\ k \leq n$, and solve

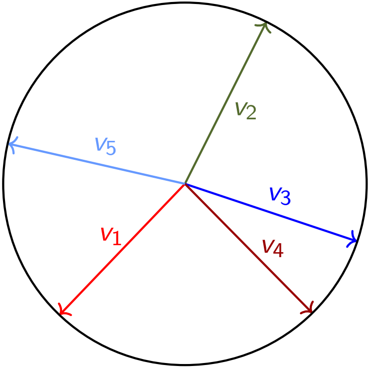
$$\begin{aligned} \max \quad & \langle \tilde{C}, V^\top V \rangle \\ \text{s. t.} \quad & V^\top V \in \mathcal{E}. \end{aligned} \qquad \text{(SDP-vec)}$$

- $V^\top V \in \mathcal{E} \Leftrightarrow \|v_i\| = 1,\ i = 1, \ldots, n$
- $(*) \Leftrightarrow$ (SDP-vec) for $k = \lceil \sqrt{2n} \rceil$ (Barvinok, 1995; Pataki, 1998)

Geometric interpretation

Optimization problem (SDP-vec)

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



$$\begin{aligned} v_i^\top v_j &= \|v_i\| \cdot \|v_j\| \cdot \cos \measuredangle(v_i, v_j) \\ &= \cos \measuredangle(v_i, v_j) \end{aligned}$$

The Mixing Method (Wang et al., 2018)

Optimization problem (SDP-vec)

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

The Mixing Method (Wang et al., 2018)

Optimization problem (SDP-vec)

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

Coordinate ascent

We fix all columns except v_i.

The Mixing Method (Wang et al., 2018)

Optimization problem (SDP-vec)

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

Coordinate ascent

We fix all columns except v_i. (SDP-vec) reduces to

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

where $g = \sum_{j=1, j\neq i}^{n} \tilde{C}_{ij} v_j = V \cdot \tilde{C}_{(i)} - \tilde{C}_{ii} v_i$.

The Mixing Method (Wang et al., 2018)

Optimization problem (SDP-vec)

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

Coordinate ascent

We fix all columns except 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 $g = \sum_{j=1, j\neq i}^{n} \tilde{C}_{ij} v_j = V \cdot \tilde{C}_{(i)} - \tilde{C}_{ii} v_i$.

The Mixing Method (Wang et al., 2018)

Optimization problem (SDP-vec)

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

Coordinate ascent

We fix all columns except 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 $g = \sum_{j=1, j \neq i}^{n} \tilde{C}_{ij} v_j = V \cdot \tilde{C}_{(i)} - \tilde{C}_{ii} v_i$.

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

Low-rank methods

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

Input: $\tilde{C} \in \mathbb{R}^{n \times n}$ with $\operatorname{diag}(\tilde{C}) = 0$, $k \in \mathbb{N}_{\geq 1}$
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**
    $v_i \leftarrow \frac{V \cdot \tilde{C}_{(i)}}{\| V \cdot \tilde{C}_{(i)} \|}$;

Low-rank methods

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

Input: $\tilde{C} \in \mathbb{R}^{n \times n}$ with $\operatorname{diag}(\tilde{C}) = 0$, $k \in \mathbb{N}_{\geq 1}$
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**
    $v_i \leftarrow \frac{V \cdot \tilde{C}_{(i)}}{\| V \cdot \tilde{C}_{(i)} \|}$;

Theorem: Local linear convergence (Wang et al., 2018)

Let $k > \sqrt{2n}$. If the iterates do not degenerate, then the Mixing Method converges locally to the global optimum of (SDP-vec) at a linear rate.

Low-rank methods

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

Input: $\tilde{C} \in \mathbb{R}^{n \times n}$ with $\mathrm{diag}(\tilde{C}) = 0$, $k \in \mathbb{N}_{\geq 1}$
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**
    $v_i \leftarrow \frac{V \cdot \tilde{C}_{(i)}}{\| V \cdot \tilde{C}_{(i)} \|}$;

Theorem: Local linear convergence (Wang et al., 2018)

Let $k > \sqrt{2n}$. If the iterates do not degenerate, then the Mixing Method converges locally to the global optimum of (SDP-vec) at a linear rate.

- block-coordinate maximization (Erdogdu et al, 2022)
- momentum-based acceleration (Kim et al., 2021)
- bilinear decomposition, ADMM (Chen & Goulart, 2023)

Approximately solving the dual problem

Dual problem

$$\min_{\gamma \geq 0} f(\gamma) = \min_{\gamma \geq 0} \left\{ b^\top \gamma + \max_{X \in \mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X \rangle \right\}$$

- f is nonsmooth
- evaluation of f at $\gamma \in \mathbb{R}^m_+$ yields
  - function value $f(\gamma)$
  - subgradient $g(\gamma) = b - \mathcal{A}(X^*)$ of f at γ
- dynamic bundle approach for SDPs by Fischer et al. (2003)

Approximately solving the dual problem

Dual problem

$$\min_{\gamma \geq 0} f(\gamma) = \min_{\gamma \geq 0} \left\{ b^\top \gamma + \max_{X \in \mathcal{E}} \langle C - \mathcal{A}^\top(\gamma), X \rangle \right\}$$

- f is nonsmooth
- evaluation of f at $\gamma \in \mathbb{R}^m_+$ yields
  - function value $f(\gamma)$
  - subgradient $g(\gamma) = b - \mathcal{A}(X^*)$ of f at γ
- dynamic bundle approach for SDPs by Fischer et al. (2003)

Model of f using trial points $\gamma_i \in \mathbb{R}^m_+$ for $i = 1, \ldots, k$:

(Proximal) cutting plane model

$$\hat{f}_k(\gamma) = \max_{1 \leq i \leq k} \{ f(\gamma_i) + \langle g(\gamma_i), \gamma - \gamma_i \rangle \} + \frac{1}{2t} \|\gamma - \hat{\gamma}\|^2$$

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $\Delta_k = \langle \tilde{C}, V_k^\top V_k - V_{k-1}^\top V_{k-1} \rangle$, improvement in iteration k
- $r_k = \frac{\Delta_k}{\Delta_{k-1}}$, $k \geq 2$, ratio of improvements

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $\Delta_k = \langle \tilde{C}, V_k^\top V_k - V_{k-1}^\top V_{k-1} \rangle$, improvement in iteration k
- $r_k = \frac{\Delta_k}{\Delta_{k-1}}$, $k \geq 2$, ratio of improvements

Relative step tolerance

- stop **if** $\frac{\|V_{k-1} - V_k\|_F}{1 + \|V_{k-1}\|_F} < 0.01$

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $\Delta_k = \langle \tilde{C}, V_k^\top V_k - V_{k-1}^\top V_{k-1} \rangle$, improvement in iteration k
- $r_k = \frac{\Delta_k}{\Delta_{k-1}}$, $k \geq 2$, ratio of improvements

Relative step tolerance

- stop **if** $\frac{\|V_{k-1} - V_k\|_F}{1 + \|V_{k-1}\|_F} < 0.01$

Practical observation

- $(\Delta_k)_{k \in \mathbb{N}_{\geq 2}}$ is strictly decreasing
- $(r_k)_{k \in \mathbb{N}_{\geq 2}}$ is strictly increasing: $0 < r_{k-1} < r_k < 1$

Gap estimation

Assuming that $(r_k)_{k\in\mathbb{N}_{\geq 2}}$ is strictly increasing, we have

$$\frac{\Delta_p}{\Delta_k} = \prod_{i=k}^{p-1} \frac{\Delta_{i+1}}{\Delta_i} \geq r_k^{p-k}, \qquad \forall p > k,$$

Gap estimation

Assuming that $(r_k)_{k\in\mathbb{N}_{\geq 2}}$ is strictly increasing, we have

$$\frac{\Delta_p}{\Delta_k} = \prod_{i=k}^{p-1} \frac{\Delta_{i+1}}{\Delta_i} \geq r_k^{p-k}, \qquad \forall p > k,$$

$$\sum_{p=k+1}^{\infty} \Delta_p = \Delta_k \sum_{p=k+1}^{\infty} \frac{\Delta_p}{\Delta_k} \geq \Delta_k \left(\frac{1}{1 - r_k} - 1 \right) = \frac{\Delta_k^2}{\Delta_{k-1} - \Delta_k}.$$

Gap estimation

Assuming that $(r_k)_{k\in\mathbb{N}_{\geq 2}}$ is strictly increasing, we have

$$\frac{\Delta_p}{\Delta_k} = \prod_{i=k}^{p-1} \frac{\Delta_{i+1}}{\Delta_i} \geq r_k^{p-k}, \qquad \forall p > k,$$

$$\sum_{p=k+1}^{\infty} \Delta_p = \Delta_k \sum_{p=k+1}^{\infty} \frac{\Delta_p}{\Delta_k} \geq \Delta_k \left(\frac{1}{1 - r_k} - 1 \right) = \frac{\Delta_k^2}{\Delta_{k-1} - \Delta_k}.$$

Lower bound

$$f_k + \frac{\Delta_k^2}{\Delta_{k-1} - \Delta_k} \leq f^*$$

Gap estimation

Assuming that $(r_k)_{k\in\mathbb{N}_{\geq 2}}$ is strictly increasing, we have

$$\frac{\Delta_p}{\Delta_k} = \prod_{i=k}^{p-1} \frac{\Delta_{i+1}}{\Delta_i} \geq r_k^{p-k}, \qquad \forall p > k,$$

$$\sum_{p=k+1}^{\infty} \Delta_p = \Delta_k \sum_{p=k+1}^{\infty} \frac{\Delta_p}{\Delta_k} \geq \Delta_k \left(\frac{1}{1-r_k} - 1 \right) = \frac{\Delta_k^2}{\Delta_{k-1} - \Delta_k}.$$

Lower bound

$$f_k + \frac{\Delta_k^2}{\Delta_{k-1} - \Delta_k} \leq f^*$$

Estimated upper bound (see MIXSAT solver, Wang & Kolter, 2019)

$$f_k + \frac{\Delta_{k-1}\Delta_k}{\Delta_{k-1} - \Delta_k} \approx f^*$$

Upper bounds via weak duality

Primal-dual pair

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

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

Upper bounds via weak duality

Primal-dual pair

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

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

Proposition (see Wang et al., 2018)

Let $V^* = \lim_{k \to \infty} V_k$. Then $y_i = \|V^* \cdot \tilde{C}_{(i)}\|_2$ is optimal for (DSDP).

Upper bounds via weak duality

Primal-dual pair

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

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

Proposition (see Wang et al., 2018)

Let $V^* = \lim_{k \to \infty} V_k$. Then $y_i = \|V^* \cdot \tilde{C}_{(i)}\|_2$ is optimal for (DSDP).

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

- approximate but non-feasible dual variables: $\tilde{y}_i = \|\tilde{V} \cdot \tilde{C}_{(i)}\|_2$

Upper bounds via weak duality

Primal-dual pair

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

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

Proposition (see Wang et al., 2018)

Let $V^* = \lim_{k \to \infty} V_k$. Then $y_i = \|V^* \cdot \tilde{C}_{(i)}\|_2$ is optimal for (DSDP).

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

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

Solver features

Speed:

- fast approximate function and subgradient evaluation
- at most one eigenvalue computation per B&B node
- tradeoff between number of B&B nodes and overall time spent

Solver features

Speed:

- fast approximate function and subgradient evaluation
- at most one eigenvalue computation per B&B node
- tradeoff between number of B&B nodes and overall time spent

Branching:

- decision based on dual information
- active cutting planes are passed down in B&B tree

Solver features

Speed:

- fast approximate function and subgradient evaluation
- at most one eigenvalue computation per B&B node
- tradeoff between number of B&B nodes and overall time spent

Branching:

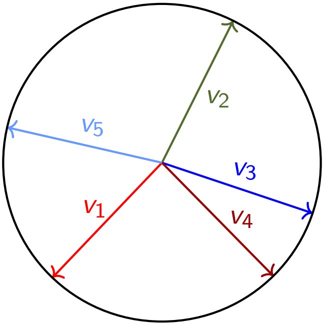
- decision based on dual information
- active cutting planes are passed down in B&B tree

Primal heuristic:

- Goemans-Williamson hyperplane rounding
  - one-opt and two-opt local search
  - ‘biased’ hyperplanes

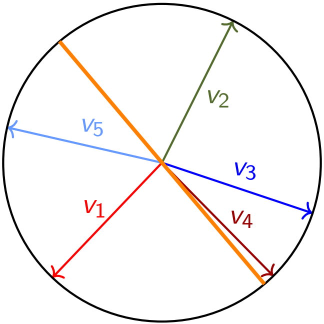
Goemans-Williamson hyperplane rounding

- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \text{sign}(h^\top v_i)$



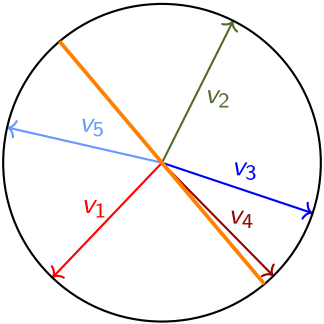
Goemans-Williamson hyperplane rounding

- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \text{sign}(h^\top v_i)$



Goemans-Williamson hyperplane rounding

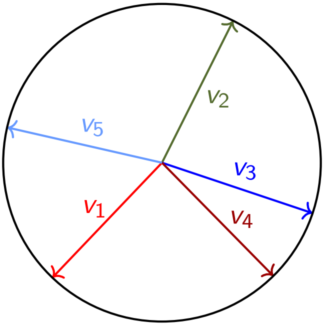
- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \text{sign}(h^\top v_i)$



- local search to improve solution (one-opt and two-opt)
- good candidates are close to hyperplane, i.e., $|h^\top v_i|$ small

Goemans-Williamson hyperplane rounding

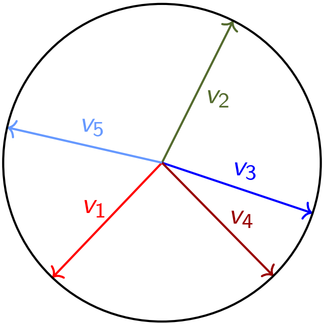
- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \text{sign}(h^\top v_i)$



- local search to improve solution (one-opt and two-opt)
- good candidates are close to hyperplane, i.e., $|h^\top v_i|$ small
- construct a 'biased hyperplane' $h^* \in \arg\max_{\|h\|=1} \|V^\top h\|^2$

Goemans-Williamson hyperplane rounding

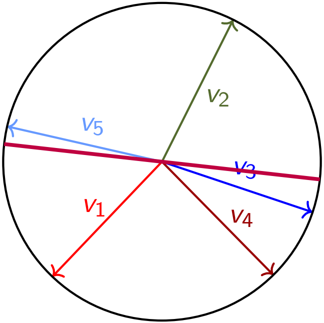
- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \text{sign}(h^\top v_i)$



- local search to improve solution (one-opt and two-opt)
- good candidates are close to hyperplane, i.e., $|h^\top v_i|$ small
- construct a 'biased hyperplane' $h^* \in \arg\max_{\|h\|=1} \|V^\top h\|^2$
- h^* is eigenvector to largest eigenvalue of $VV^\top$

Goemans-Williamson hyperplane rounding

- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \text{sign}(h^\top v_i)$



- local search to improve solution (one-opt and two-opt)
- good candidates are close to hyperplane, i.e., $|h^\top v_i|$ small
- construct a 'biased hyperplane' $h^* \in \arg\max_{\|h\|=1} \|V^\top h\|^2$
- h^* is eigenvector to largest eigenvalue of $VV^\top$

Computational results I

- implementation in C, linked against Intel MKL
- all solvers are run in single-threaded mode on same hardware

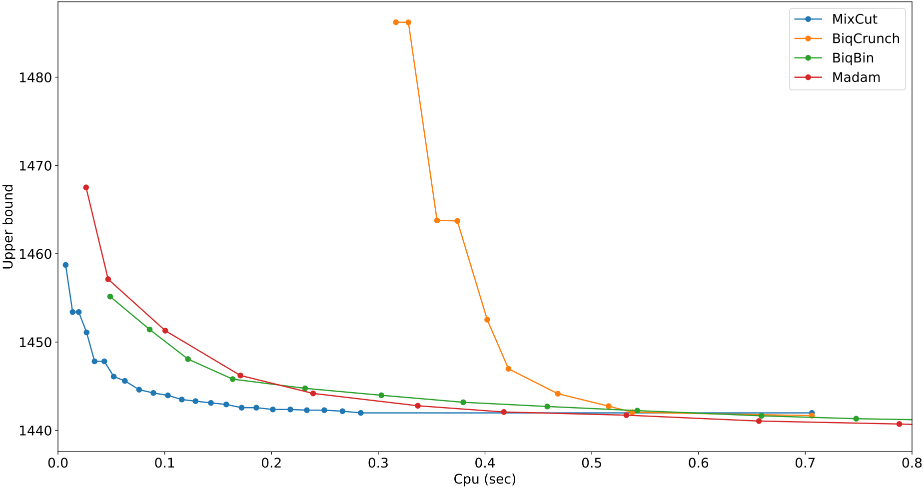
Computational results I

- implementation in C, linked against Intel MKL
- all solvers are run in single-threaded mode on same hardware

Instance	BiqCrunch		BiqBin		MADAM		MixCut	
	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes
g05_100.0	253.89	325	107.48	99	88.09	195	**14.20**	743
g05_100.1	1447.89	1779	554.74	465	522.70	863	**66.06**	3615
g05_100.2	92.26	97	33.03	29	36.99	55	**4.19**	193
g05_100.3	454.63	659	195.09	209	127.56	389	**24.39**	1253
g05_100.4	31.85	31	12.43	7	12.85	11	**2.05**	87
g05_100.5	103.74	93	30.42	19	24.48	25	**4.77**	219
g05_100.6	99.20	99	36.37	25	43.01	33	**5.42**	245
g05_100.7	212.91	205	86.21	65	84.37	85	**9.27**	453
g05_100.8	143.52	165	60.57	41	48.29	79	**7.22**	361
g05_100.9	169.25	237	61.87	57	52.47	155	**8.02**	393

- Erdős–Rényi graphs $G_{100,\frac{1}{2}}$ (unweighted)
- time in seconds

Root node bounds for g05_100.1

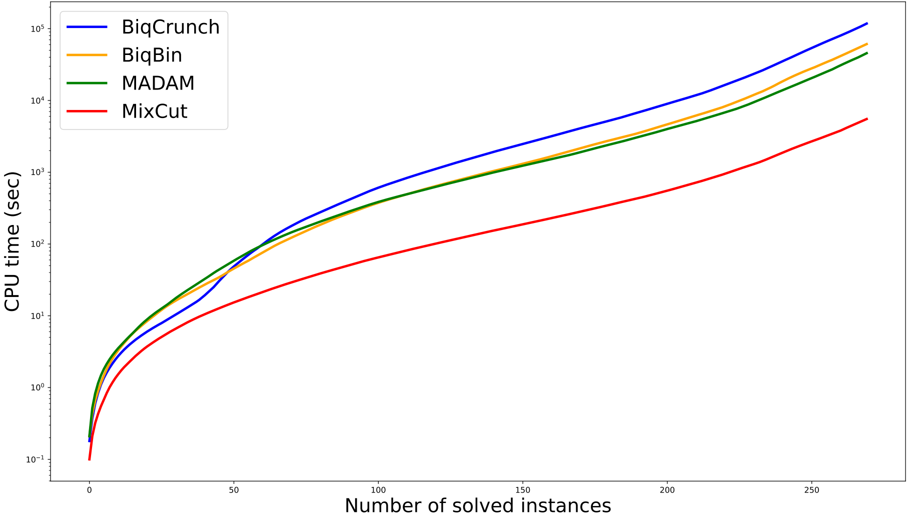


Computational results II

- average CPU times (s) and B&B nodes for $n = 120$

Instance	BiqCrunch		BiqBin		MADAM		MixCut	
	Time	Nodes	Time	Nodes	Time	Nodes	Time	Nodes
g05	1102.90	2039	750.59	1419	477.77	1390	**70.38**	3109
pm1d	1998.67	2792	1105.18	1772	652.46	1471	**93.63**	4373
pm1s	110.63	202	77.20	148	64.70	179	**6.00**	219
pw01	49.02	44	25.88	34	18.75	23	**2.95**	97
pw05	2040.15	2533	1013.91	1907	743.24	1369	**84.98**	3721
pw09	2064.74	2446	1009.16	1691	757.77	1268	**103.71**	4300
w01	26.87	22	17.95	23	16.68	15	**2.02**	64
w05	1199.89	1339	623.85	974	532.44	730	**44.50**	1909
w09	1832.45	1897	812.77	1754	743.84	1188	**65.28**	2882

Comparison on 270 instances with $n \in \{80, 100, 120\}$



Conclusion and future work

Conclusion:

- significant speedup of ‘traditional’ approaches by recent low-rank methods
- further improvements by
  - branching decision based on dual information
  - reducing number of cutting plane iterations

Conclusion and future work

Conclusion:

- significant speedup of ‘traditional’ approaches by recent low-rank methods
- further improvements by
  - branching decision based on dual information
  - reducing number of cutting plane iterations

Future work:

- use parallelization
- apply to general binary quadratic problems

Conclusion and future work

Conclusion:

- significant speedup of 'traditional' approaches by recent low-rank methods
- further improvements by
  - branching decision based on dual information
  - reducing number of cutting plane iterations

Future work:

- use parallelization
- apply to general binary quadratic problems

Thank you!