June 2, 2023

Efficiently Solving QUBO Problems

Joint work with Valentin Durante, Federal University of Toulouse

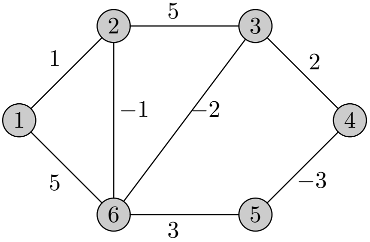
FWF Der Wissenschaftsfonds.

UNIVERSITÄT KLAGENFURT

Modeling - Analysis - Optimization DOC.FUNDS DOCTORAL SCHOOL

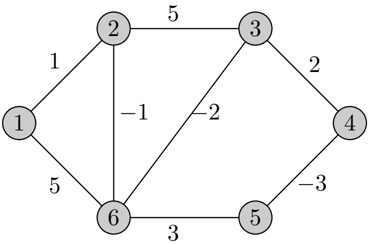
The (weighted) Max-Cut Problem

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

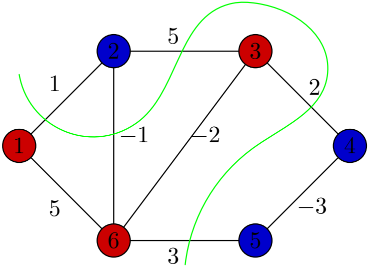
For $S \subseteq V$, the set of edges

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

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

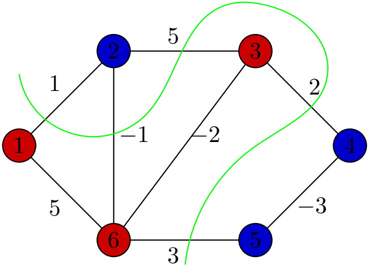
For $S \subseteq V$, the set of edges

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

is called the *cut* induced by S.

The (weighted) Max-Cut Problem

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



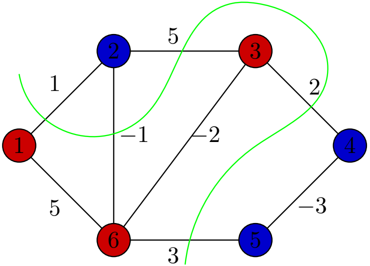
Max-Cut Problem

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

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

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

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

Quadratic unconstrained binary optimization

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

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

Quadratic unconstrained binary optimization

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

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

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 progamming relaxation

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

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

Semidefinite progamming relaxation

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

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

Equivalent formulations (Laurent & Poljak, 1995)

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

Semidefinite progamming 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

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

Semidefinite progamming 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

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

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)
  - `MADAM` (2021)
  - `BiqCrunch` (2016)
  - `BiqBin` (2022)
- $\mathcal{O}(n^3)$ triangle inequalities:

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

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

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

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

Lagrangian relaxation

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

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

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

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

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: $f^* \le f(\gamma)$ for all $\gamma \in \mathbb{R}^m_+$

Lagrangian relaxation

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

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

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

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

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: $f^* \le f(\gamma)$ for all $\gamma \in \mathbb{R}^m_+$
- dual problem:

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

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

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

- `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

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

- `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

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

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

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

- `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

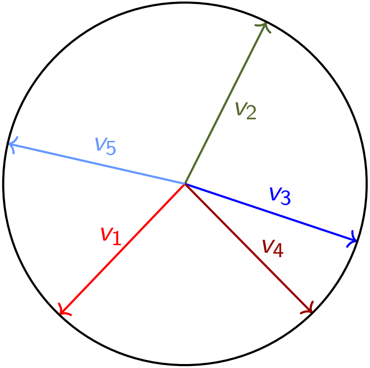
$$\begin{aligned} \max \quad & \langle \tilde{C}, V^\top V \rangle \\ \text{s. t.} \quad & V^\top V \in \mathcal{E}. \end{aligned} \tag{SDP-vec}$$

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

Geometric interpretation

Optimization problem (SDP-vec)

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



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

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)

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

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

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

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

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 $\operatorname{diag}(\tilde{C}) = 0$, $k \in \mathbb{N}_{\geq 1}$
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.

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

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $f_k = \langle \tilde{C}, V_k^\top V_k \rangle$, function value after iteration k
- $\Delta_k = f_k - f_{k-1}$, objective improvement in iteration k

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $f_k = \langle \tilde{C}, V_k^\top V_k \rangle$, function value after iteration k
- $\Delta_k = f_k - f_{k-1}$, objective improvement in iteration k

Stopping criterion: relative step tolerance

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

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $f_k = \langle \tilde{C}, V_k^\top V_k \rangle$, function value after iteration k
- $\Delta_k = f_k - f_{k-1}$, objective improvement in iteration k

Stopping criterion: relative step tolerance

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

Stopping criterion: estimated gap (see `MIXSAT` solver, Wang & Kolter, 2019)

- stop **if** $0 < \varepsilon = \frac{\Delta_{k-1}\Delta_k}{\Delta_{k-1} - \Delta_k}$ small $\Rightarrow f^* \approx f_k + \varepsilon$

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $f_k = \langle \tilde{C}, V_k^\top V_k \rangle$, function value after iteration k
- $\Delta_k = f_k - f_{k-1}$, objective improvement in iteration k

Stopping criterion: relative step tolerance

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

Stopping criterion: estimated gap (see `MIXSAT` solver, Wang & Kolter, 2019)

- stop **if** $0 < \varepsilon = \frac{\Delta_{k-1}\Delta_k}{\Delta_{k-1} - \Delta_k}$ small $\Rightarrow f^* \approx f_k + \varepsilon$
- caveat: the actual optimum can be smaller or larger!

When do we stop the mixing method?

Notation

- V_k: matrix V after iteration k
- $f_k = \langle \tilde{C}, V_k^\top V_k \rangle$, function value after iteration k
- $\Delta_k = f_k - f_{k-1}$, objective improvement in iteration k

Stopping criterion: relative step tolerance

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

Stopping criterion: estimated gap (see `MIXSAT` solver, Wang & Kolter, 2019)

- stop **if** $0 < \varepsilon = \frac{\Delta_{k-1}\Delta_k}{\Delta_{k-1} - \Delta_k}$ small $\Rightarrow f^* \approx f_k + \varepsilon$
- caveat: the actual optimum can be smaller or larger!

How do we bound f^* from above (dualbound)?

Upper bounds via weak duality

Primal-dual pair

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

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

Upper bounds via weak duality

Primal-dual pair

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

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

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

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

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

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

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

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

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: $y = \tilde{y} - \lambda_{\min}\left(\operatorname{Diag}(\tilde{y}) - \tilde{C}\right) e$

Approximately solving the 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

Approximately solving the 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 $g = b - \mathcal{A}(X^*)$ of f at γ

Approximately solving the 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 $g = b - \mathcal{A}(X^*)$ of f at γ
- dynamic bundle approach for SDPs by Fischer et al., 2003
- implementation similar to `BiqMac` and `BiqBin`

Solver features

Speed:

- fast approximate function and subgradient evaluation
- usually a single 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
- usually a single 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
- usually a single eigenvalue computation per B&B node
- tradeoff between number of B&B nodes and overall time spent

Branching:

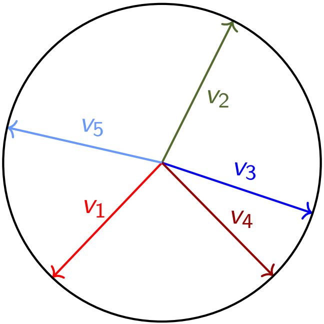
- decision based on dual information
- active cutting planes are passed down in B&B tree

Primal heuristic:

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

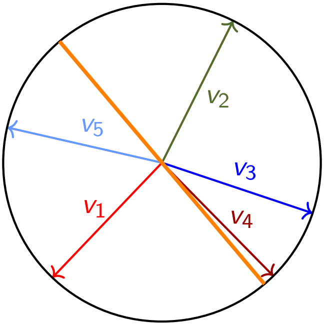
Goemans-Williamson hyperplane rounding

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



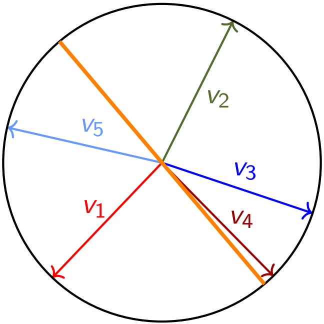
Goemans-Williamson hyperplane rounding

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



Goemans-Williamson hyperplane rounding

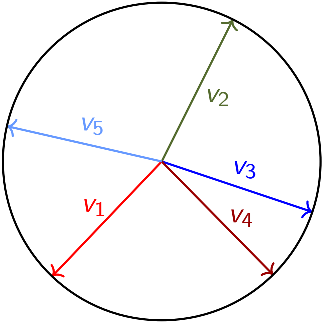
- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \mathrm{sign}(h^\top v_i)$



- 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

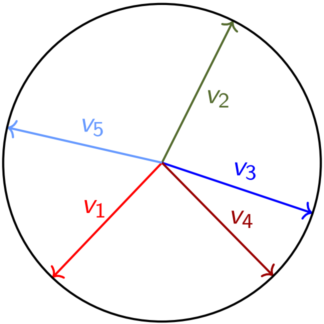
- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \mathrm{sign}(h^\top v_i)$



- 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

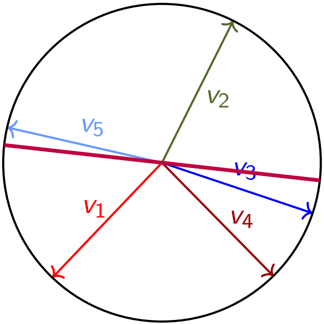
- choose random normal vector $h \in \mathcal{S}^{k-1}$; set $x_i = \text{sign}(h^\top v_i)$



- 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

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

instance	BiqMac		MADAM		our solver	
	time	nodes	time	nodes	time	nodes
g05_100.0	555.16	531	98.33	195	17.19	751
g05_100.1	3547.17	3643	494.10	705	84.78	3888
g05_100.2	115.87	127	40.07	43	5.31	305
g05_100.3	1308.85	1215	129.60	497	29.48	1292
g05_100.4	71.03	69	9.71	11	2.68	99
g05_100.5	116.16	129	28.63	31	5.31	203
g05_100.6	177.22	193	29.52	47	6.52	253
g05_100.7	332.35	337	75.31	73	11.74	495
g05_100.8	291.28	275	35.78	67	8.50	367
g05_100.9	321.10	277	47.34	101	9.57	403

Table: CPU times (s) and B&B nodes for ‘g05’ instances.

Computational results II

- Erdős–Rényi graphs $G_{180,\frac{1}{2}}$ (unweighted)
- MADAM: parallel run on 20 CPUs (240 cores in total)
- our solver: single-threaded
- (∗): $240 \cdot \frac{\text{time}_{\texttt{MADAM}}}{\text{time}_{\text{our solver}}}$

instance	MADAM		our solver		
	time	nodes	time	nodes	(∗)
g05_180.0	671.23	148,617	12164.62	190,859	13.24
g05_180.1	670.63	137,665	12981.40	204,257	12.40
g05_180.2	1116.24	281,215	20693.20	325,851	12.95
g05_180.3	3706.61	786,457	90040.88	1,084,351	9.88
g05_180.4	5209.59	1,556,485	72889.44	987,595	17.15
g05_180.5	8964.00	2,333,997	171576.67	2,803,449	12.54
g05_180.6	10542.41	2,298,681	215391,59	2,926,271	11.75

Table: CPU times (s) and B&B nodes for ‘g05’ instances.

Conclusion and future work

Conclusion:

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

Future work:

- include more cuts
- use parallelization
- apply to QCQPs

Conclusion and future work

Conclusion:

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

Future work:

- include more cuts
- use parallelization
- apply to QCQPs

Thank you!