October 7, 2022

A Mixing Method based Branch-and-Bound Solver for QUBO Problems

Joint work with Valentin Durante

FWF Der Wissenschaftsfonds.

UNIVERSITÄT KLAGENFURT

Modeling - Analysis - Optimization · DOC.FUNDS DOCTORAL SCHOOL

Jan Schwiddessen
University of Klagenfurt, Department of Mathematics

Quadratic Unconstrained Binary Optimization (QUBO)

Optimization Problem (QUBO)

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

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

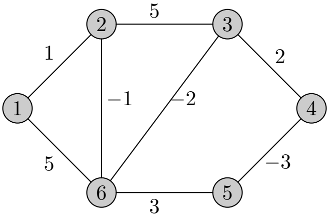
- $\mathcal{NP}$-hard problem
- LP approaches exist only for sparse C

Example:

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

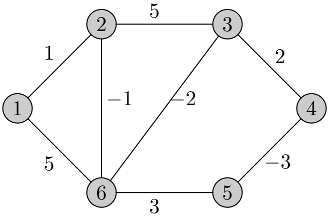
The (Weighted) Max-Cut Problem

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 $w \in \mathbb{R}^E$

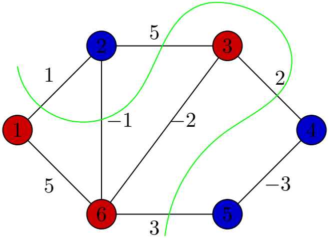


Goal: find a maximum cut in G, i.e., an optimal solution of

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

The (Weighted) Max-Cut Problem

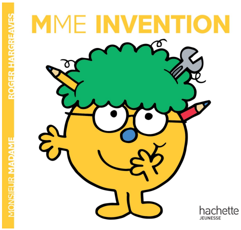
Given: undirected graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^E$



Goal: find a maximum cut in G, i.e., an optimal solution of

$$\max_{S \subseteq V} \sum_{i \in S,\ j \in V \setminus S} w_{ij} \qquad \text{(MC)}$$

Solvers for dense C using Semidefinite Programming

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

Semidefinite 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

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

Semidefinite Relaxation

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

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

- all mentioned solvers: additional ‘clique’ inequalities
- but competitive implementations possible without inequalities

Low-rank Factorization $X = V^\top V$

Factorization of $X \succeq 0$

$$X = V^\top V$$

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

Low-rank Factorization $X = V^\top V$

Factorization of $X \succeq 0$

$$X = V^\top V$$

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

- $X_{ij} = v_i^\top v_j \quad \Rightarrow \quad \langle C, X \rangle = \sum_{i,j=1}^n C_{ij} X_{ij} = \sum_{i,j=1}^n C_{ij} v_i^\top v_j$
- $\operatorname{diag}(X) = e \quad \Leftrightarrow \quad \|v_i\|^2 = 1, \; i = 1, \dots, n$

Low-rank Factorization $X = V^\top V$

Factorization of $X \succeq 0$

$$X = V^\top V$$

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

- $X_{ij} = v_i^\top v_j \quad \Rightarrow \quad \langle C, X \rangle = \sum_{i,j=1}^n C_{ij} X_{ij} = \sum_{i,j=1}^n C_{ij} v_i^\top v_j$
- $\operatorname{diag}(X) = e \quad \Leftrightarrow \quad \|v_i\|^2 = 1, \; i = 1, \dots, n$

Optimization Problem (SDP-vec)

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

Low-rank Factorization $X = V^\top V$

Factorization of $X \succeq 0$

$$X = V^\top V$$

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

- $X_{ij} = v_i^\top v_j \quad \Rightarrow \quad \langle C, X \rangle = \sum_{i,j=1}^n C_{ij} X_{ij} = \sum_{i,j=1}^n C_{ij} v_i^\top v_j$
- $\mathrm{diag}(X) = e \quad \Leftrightarrow \quad \|v_i\|^2 = 1, \; i = 1, \dots, n$

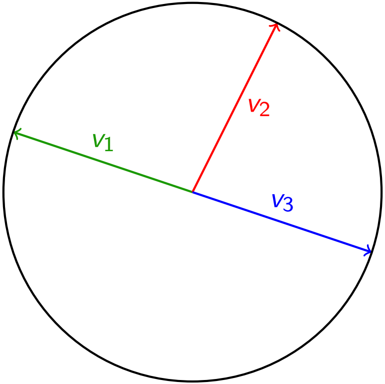
Optimization Problem (SDP-vec)

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

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

Geometric Interpretation

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



Coordinate Ascent Method

Optimization Problem (SDP-vec)

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

Coordinate Ascent

We fix all but one vector 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)

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

Coordinate Ascent

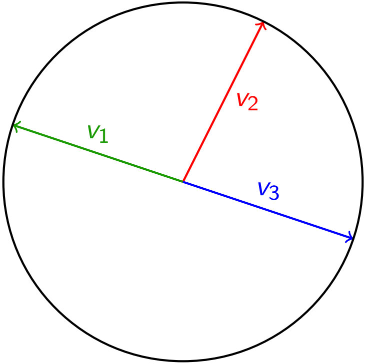
We fix all but one vector v_i. (SDP-vec) reduces to

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

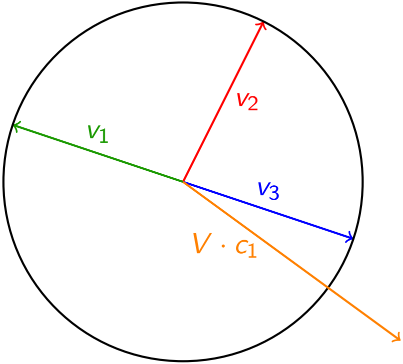
where $g = \sum_j^n c_{ij} v_j = V \cdot c_i$

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

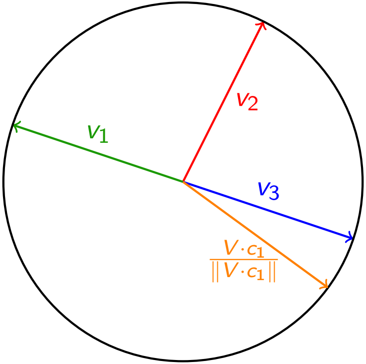
Mixing Method (Wang el al., 2018)



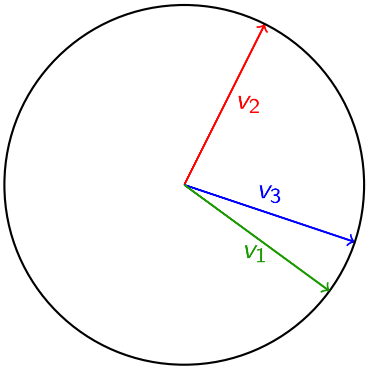
Mixing Method (Wang el al., 2018)



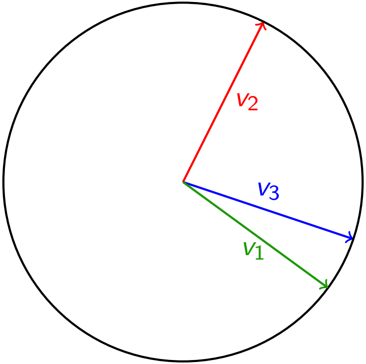
Mixing Method (Wang el al., 2018)



Mixing Method (Wang el al., 2018)



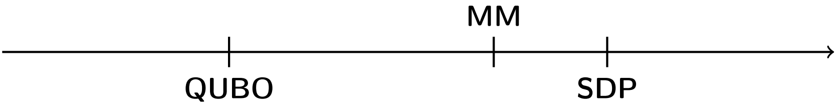
Mixing Method (Wang el al., 2018)



Mixing Method

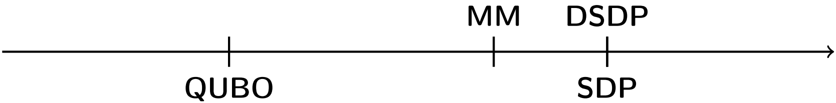
- repeat for $v_1, v_2, \ldots, v_n$ again and again
- initialize V randomly on the unit sphere
- converges to optimal solution with linear rate

Exact Approach for QUBO



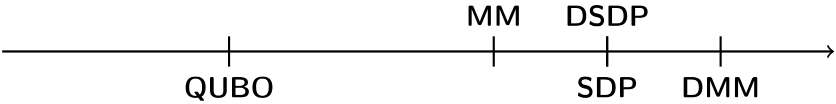
- we have to stop the Mixing Method at some point
- we cannot guarantee that QUBO $\leq$ MM

Exact Approach for QUBO



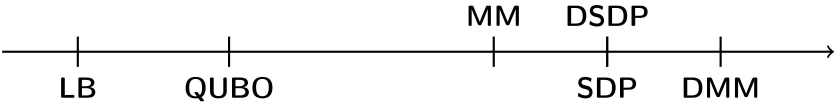
- we have to stop the Mixing Method at some point
- we cannot guarantee that QUBO $\leq$ MM

Exact Approach for QUBO



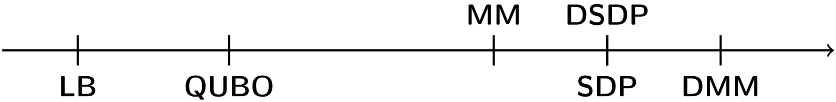
- we have to stop the Mixing Method at some point
- we cannot guarantee that QUBO $\leq$ MM

Exact Approach for QUBO



- we have to stop the Mixing Method at some point
- we cannot guarantee that QUBO $\leq$ MM
- postprocessing recovers upper bound: QUBO $\leq$ DMM
- heuristics provide lower bound: LB $\leq$ QUBO

Exact Approach for QUBO



- we have to stop the Mixing Method at some point
- we cannot guarantee that QUBO $\leq$ MM
- postprocessing recovers upper bound: QUBO $\leq$ DMM
- heuristics provide lower bound: LB $\leq$ QUBO

Branch-and-Bound

If DMM $>$ LB, partition QUBO into two smaller subproblems and proceed recursively.

Results and Future Work

- C implementation using Intel MKL
- tested on many instances with $n \leq 100$

Results

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

Results and Future Work

- C implementation using Intel MKL
- tested on many instances with $n \leq 100$

Results

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

Future Work

- tackle larger problems
- use clique inequalities via Lagrangian duality

Results and Future Work

- C implementation using Intel MKL
- tested on many instances with $n \leq 100$

Results

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

Future Work

- tackle larger problems
- use clique inequalities via Lagrangian duality

Thank you!