Mixed-Integer Semidefinite Programming

- Mixed-integer semidefinite program

\[
\begin{align*}
\sup & \quad b^Ty \\
\text{s.t.} & \quad C - \sum_{i=1}^{m} A_iy_i \succeq 0, \\
& \quad y_i \in \mathbb{Z} \quad \forall \ i \in \mathcal{I}
\end{align*}
\]

for symmetric matrices \(A_i, C\)

- Linear constraints, bounds, multiple blocks possible within SDP-constraint
Mixed-Integer Semidefinite Programming

- Mixed-integer semidefinite program

\[
\begin{align*}
\sup & \quad b^T y \\
\text{s.t.} & \quad C - \sum_{i=1}^{m} A_i y_i \succeq 0, \\
& \quad y_i \in \mathbb{Z} \quad \forall \ i \in \mathcal{I}
\end{align*}
\]

for symmetric matrices \(A_i, C\)

- Linear constraints, bounds, multiple blocks possible within SDP-constraint

- Efficient solvers for specific applications, but few solvers (and theory) for the general case
Contents

Applications

Solution Approaches
  Outer Approximation
  SDP-based Branch-and-Bound

Warmstarts

Dual Fixing

Solvers for MISDP

Conclusion & Outlook
Applications

Solution Approaches
  Outer Approximation
  SDP-based Branch-and-Bound

Warmstarts

Dual Fixing

Solvers for MISDP

Conclusion & Outlook
Classical Example: Max-Cut

Find Cut $\delta(S)$, with $S \subseteq V$ and $\{i, j\} \in \delta(S)$ iff $i \in S, j \in V \setminus S$, that maximizes

$$\sum_{\{i, j\} \in \delta(S)} c_{ij}.$$
Classical Example: Max-Cut

Max-Cut

Find Cut $\delta(S)$, with $S \subseteq V$ and $\{i, j\} \in \delta(S)$ iff $i \in S, j \in V \setminus S$, that maximizes

$$\sum_{\{i, j\} \in \delta(S)} c_{ij}.$$ 

Using variables $(x_i)_{i \in V} \in \{-1, 1\}^n$ with $x_i = 1 \iff i \in S$, this is equivalent to

Max-Cut MIQP

$$\max \sum_{i < j} c_{ij} \frac{1 - x_i x_j}{2}$$

s.t. $x_i \in \{-1, 1\} \forall i \leq n$
Classical Example: Max-Cut

\[
\sum_{i<j} c_{ij} \frac{1 - x_i x_j}{2} = \frac{1}{4} \sum_{i=1}^{n} \left( \sum_{j=1}^{n} c_{ij} x_i x_i - \sum_{j=1}^{n} c_{ij} x_i x_j \right)
\]

\[
= \frac{1}{4} x^T (\text{Diag}(C \mathbb{1}) - C) x
\]
Classical Example: Max-Cut

\[ \sum_{i<j} c_{ij} \frac{1 - x_i x_j}{2} = \frac{1}{4} \sum_{i=1}^{n} \left( \sum_{j=1}^{n} c_{ij} x_i x_j - \sum_{j=1}^{n} c_{ij} x_i x_j \right) \]

\[ = \frac{1}{4} x^T (\text{Diag}(C \mathbb{1}) - C)x \]

With \( X := xx^T \) (and notation \( A \bullet B := \text{Tr}(AB) = \sum_{ij} A_{ij} B_{ij} \)), this is equivalent to

Max-Cut Rk1-MISDP [Poljak, Rendl 1995]

\[
\begin{align*}
\text{max} & \quad \frac{1}{4} (\text{Diag}(C \mathbb{1}) - C) \bullet X \\
\text{s.t.} \quad & \quad \text{diag}(X) = \mathbb{1} \\
& \quad \text{Rank}(X) = 1 \\
& \quad X \succeq 0 \\
& \quad X_{ij} \in \{-1, 1\}
\end{align*}
\]
Classical Example: Max-Cut

Max-Cut Rk1-MISDP

\[
\begin{align*}
\text{max} & \quad \frac{1}{4} (\text{Diag}(C \mathbb{1}) - C) \cdot X \\
\text{s.t.} & \quad \text{diag}(X) = \mathbb{1} \\
& \quad \text{Rank}(X) = 1 \\
& \quad X \succeq 0 \\
& \quad X_{ij} \in \{-1, 1\}
\end{align*}
\]

- Relaxation still non-convex because of rank constraint
Classical Example: Max-Cut

Max-Cut MISDP

\[
\begin{align*}
\text{max} \quad & \frac{1}{4} \left( \text{Diag}(C \mathbb{1}) - C \right) \bullet X \\
\text{s.t.} \quad & \text{diag}(X) = \mathbb{1} \\
& \text{Rank}(X) = 1 \\
& X \succeq 0 \\
& X_{ij} \in \{-1, 1\}
\end{align*}
\]

- Relaxation still non-convex because of rank constraint

Theorem [Laurent, Poljak 1995]

Every integral solution satisfies \( \text{Rank}(X) = 1 \).
Compressed Sensing

- Task: find sparsest solution to underdetermined system of linear equations, i.e. a solution of

\[
\min_{x} \| x \|_0 \\
\text{s.t.} \quad Ax = b \\
x \in \mathbb{R}^n
\]

where \( \| x \|_0 := |\text{supp}(x)| \).
Compressed Sensing

- Task: find sparsest solution to underdetermined system of linear equations, i.e. a solution of

\[
\begin{align*}
\ell_0\text{-Minimization} \\
\min & \quad \|x\|_0 \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in \mathbb{R}^n
\end{align*}
\]

where \(\|x\|_0 := |\text{supp}(x)|\).

- Under certain conditions on \(A\), this is equivalent to

\[
\ell_1\text{-Minimization} \\
\min & \quad \|x\|_1 \\
\text{s.t.} & \quad Ax = b \\
& \quad x \in \mathbb{R}^n
\]
Compressed Sensing

One such condition is the (asymmetric) restricted isometry property (RIP):

$$\alpha_k^2 \|x\|_2^2 \leq \|Ax\|_2^2 \leq \beta_k^2 \|x\|_2^2 \quad \forall x : \|x\|_0 \leq k$$
Compressed Sensing

One such condition is the (asymmetric) restricted isometry property (RIP):

\[ \alpha_k^2 \| x \|_2^2 \leq \| Ax \|_2^2 \leq \beta_k^2 \| x \|_2^2 \quad \forall x : \| x \|_0 \leq k \]

**Theorem [Foucart, Lai 2008]**

If \( Ax = b \) has a solution \( x \) with \( \| x \|_0 \leq k \) and the RIP of order \( 2k \) holds for

\[ \frac{\beta_{2k}^2}{\alpha_{2k}^2} < 4\sqrt{2} - 3 \approx 2.6569, \]

then \( x \) is the unique solution of both the \( \ell_0 \)- and the \( \ell_1 \)-optimization problem.
Compressed Sensing

The optimal constant $\alpha_k^2$ (and similarly $\beta_k^2$) for
\[
\alpha_k^2 \| x \|_2^2 \leq \| Ax \|_2^2 \leq \beta_k^2 \| x \|_2^2 \quad \forall x : \| x \|_0 \leq k
\]
can be computed via the following non-convex rank-constrained MISDP:

**RIP-Rk1-MISDP**

\[
\begin{align*}
\text{min} & \quad \text{Tr}(A^TAX) \\
\text{s.t.} & \quad \text{Tr}(X) = 1 \\
& \quad -z_j \leq X_{jj} \leq z_j \quad \forall j \leq n \\
& \quad \sum_{j=1}^n z_j \leq k \\
& \quad \text{Rank}(X) = 1 \\
& \quad X \succeq 0 \\
& \quad z \in \{0, 1\}^n
\end{align*}
\]
Compressed Sensing

**RIP-MISDP**

\[
\begin{align*}
\text{min} & \quad \text{Tr}(A^T AX) \\
\text{s.t.} & \quad \text{Tr}(X) = 1 \\
& \quad -z_j \leq X_{jj} \leq z_j \quad \forall j \leq n \\
& \quad \sum_{j=1}^{n} z_j \leq k \\
& \quad \text{Rank}(X) = 1 \\
& \quad X \succeq 0 \\
& \quad z \in \{0, 1\}^n
\end{align*}
\]

**Theorem [G., Pfetsch 2016]**

There always exists an optimal solution for (RIP-MISDP) with Rank(X) = 1.
Truss Topology Design

- \( n \) nodes \( V = \{ v_i \in \mathbb{R}^d : i = 1, \ldots, n \} \)
- \( n_f \) free nodes \( V_f \subset V \)
- \( m \) possible bars
  \[ E \subseteq \{ \{ v_i, v_j \} : i \neq j \}, |E| = m \]
- Force \( f \in \mathbb{R}^{d_f} \) for \( d_f = d \cdot n_f \)
Truss Topology Design

- $n$ nodes $V = \{v_i \in \mathbb{R}^d : i = 1, \ldots, n\}$
- $n_f$ free nodes $V_f \subset V$
- $m$ possible bars $E \subseteq \{\{v_i, v_j\} : i \neq j\}$, $|E| = m$
- Force $f \in \mathbb{R}^{d_f}$ for $d_f = d \cdot n_f$
- Cross-sectional areas $x \in \mathbb{R}_{+}^m$ for bars that minimize the volume while creating a “stable” truss
- Stability is measured by the compliance $\frac{1}{2} f^T u$ with node displacements $u$

**ground structure 3x3**

**optimal structure**
Truss Topology Design

TTD-SDP [Ben-Tal, Nemirovski 1997]

\[
\min \sum_{e \in E} \ell_e x_e \\
\text{s.t.} \quad \begin{pmatrix} 2C_{\text{max}} & f^T \\ f & A(x) \end{pmatrix} \succeq 0 \\
x_e \geq 0 \quad \forall e \in E
\]

- \( E \): set of possible bars
- \( \ell_e \): length of bar \( e \)
- \( x \): cross-sectional areas
- \( f \): external force
- \( C_{\text{max}} \): upper bound on compliance
- \( A_e \): bar stiffness matrices

with stiffness matrix \( A(x) = \sum_{e \in E} A_e \ell_e x_e \).
In practice, we won’t be able to produce/buy bars of any desired size.

⇒ Only allow cross-sectional areas from a discrete set \( \mathcal{A} \).
In practice, we won’t be able to produce/buy bars of any desired size. Only allow cross-sectional areas from a discrete set $\mathcal{A}$.

**TTD-MISDP [Kočvara 2010, Mars 2013]**

$$
\begin{align*}
\min & \sum_{e \in E} \ell_e \sum_{a \in \mathcal{A}} a x_e^a \\
\text{s.t.} & \begin{pmatrix} 2C_{\text{max}} & f^T \\ f & A(x) \end{pmatrix} \succeq 0 \\
& \sum_{a \in \mathcal{A}} x_e^a \leq 1 \quad \forall e \in E \\
& \sum_{a \in \mathcal{A}} x_e^a \in \{0, 1\} \quad \forall e \in E, a \in \mathcal{A},
\end{align*}
$$

where $A(x) = \sum_{e \in E} \sum_{a \in \mathcal{A}} A_e \ell_e a x_e^a$. 

January 11, 2018 | Applications and Solution Approaches for Mixed-Integer Semidefinite Programming | Tristan Gally | 14
Further Applications

- AC power flow
  - Transmission switching problems
  - Unit commitment problems
- Cardinality-constrained least-squares
- Minimum $k$-partitioning
- Quadratic assignment problems (including TSP as special case)
- Robustification of physical parameters in gas networks
- Subset selection for eliminating multicollinearity
- ...
Applications

Solution Approaches
  Outer Approximation
  SDP-based Branch-and-Bound

Warmstarts

Dual Fixing

Solvers for MISDP

Conclusion & Outlook
Idea: Solve LP/MIP and enforce SDP-constraint via linear cuts
Outer Approximation / Cutting Planes

- Idea: Solve LP/MIP and enforce SDP-constraint via linear cuts

- Cutting plane approach (Kelley 1960):
  - Solve a single MIP
  - In each node add cuts to enforce nonlinear constraints and resolve LP
Outer Approximation / Cutting Planes

- Idea: Solve LP/MIP and enforce SDP-constraint via linear cuts

- Cutting plane approach (Kelley 1960):
  - Solve a single MIP
  - In each node add cuts to enforce nonlinear constraints and resolve LP

- Outer Approximation (Quesada/Grossmann 1992):
  - Solve MIP (without nonlinear constraints) to optimality
  - Solve continuous relaxation for fixed integer variables
  - If objectives do not agree, update polyhedral approximation
  - Resolve MIP and continue iterating
Enforcing the SDP-Constraint

- For convex MINLP one usually uses gradient cuts

\[ g_j(x) + \nabla g_j(x)^\top (x - \bar{x}) \leq 0. \]

- But function of smallest eigenvalue is not differentiable everywhere.
Enforcing the SDP-Constraint

▶ For convex MINLP one usually uses gradient cuts

\[ g_j(x) + \nabla g_j(x)^\top (x - \bar{x}) \leq 0. \]

▶ But function of smallest eigenvalue is not differentiable everywhere.

⇒ Instead use characterization \( X \succeq 0 \iff u^\top X u \geq 0 \) for all \( u \in \mathbb{R}^n \)

▶ If \( Z := C - \sum_{i=1}^m A_i y_i^* \not\succeq 0 \), compute eigenvector \( v \) to smallest eigenvalue. Then

\[ v^\top Z v \geq 0 \]

is a valid linear cut that cuts off \( y^* \).
Cutting Plane Approach: MISOCP vs. MISDP

- Successfully used by many commercial solvers for mixed-integer second-order cone

- Outer approximation for SOCPs possible with polynomial number of cuts (Ben-Tal/Nemirovski 2001)

- Outer approximation for SDPs needs exponential number of cuts (Braun et al. 2015)
SDP-based Branch-and-Bound

- Relax integrality instead of SDP-constraint
- Need to solve a continuous SDP in each branch-and-bound node
- Relaxations can be solved by problem-specific approaches (e.g. conic bundle or low-rank methods) or interior-point
- Need to satisfy convergence assumptions of SDP-solvers
Strong Duality in SDP

Dual SDP (D)

\[
\sup b^T y \\
\text{s.t. } C - \sum_{i=1}^{m} A_i y_i \succeq 0 \\
y \in \mathbb{R}^m
\]

Primal SDP (P)

\[
\inf C \bullet X \\
\text{s.t. } A_i \bullet X = b_i \quad \forall i \leq m \\
X \succeq 0
\]

Strong Duality holds if Slater condition holds for (P) or (D), i.e., there exists a feasible \( X \succ 0 \) for (P) or \( y \) such that \( C - \sum_{i=1}^{m} A_i y_i \succ 0 \) in (D).

If Slater holds for (P), optimal objective of (D) is attained and vice versa.

Existence of a KKT-point is guaranteed if Slater holds for both, usual assumption of interior-point algorithms for SDP.

Need to assume this for root node. But is this enough or can these assumptions be lost through branching?
Strong Duality in SDP

**Dual SDP (D)**

\[
\sup \quad b^T y \\
\text{s.t.} \quad C - \sum_{i=1}^{m} A_i y_i \succeq 0 \\
y \in \mathbb{R}^m
\]

**Primal SDP (P)**

\[
\inf \quad C \cdot X \\
\text{s.t.} \quad A_i \cdot X = b_i \quad \forall \ i \leq m \\
X \succeq 0
\]

Strong Duality holds if Slater condition holds for (P) or (D), i.e., there exists a feasible \( X \succeq 0 \) for (P) or \( y \) such that \( C - \sum_{i=1}^{m} A_i y_i \succeq 0 \) in (D).
Strong Duality in SDP

- **Strong Duality holds if Slater condition holds for (P) or (D), i.e., there exists a feasible \( X \succ 0 \) for (P) or \( y \) such that \( C - \sum_{i=1}^{m} A_i y_i \succ 0 \) in (D).**

- **If Slater holds for (P), optimal objective of (D) is attained and vice versa.**
Strong Duality in SDP

**Dual SDP (D)**

\[
\begin{align*}
\sup & \quad b^T y \\
\text{s.t.} & \quad C - \sum_{i=1}^{m} A_i y_i \succeq 0 \\
& \quad y \in \mathbb{R}^m
\end{align*}
\]

- Strong Duality holds if Slater condition holds for (P) or (D), i.e., there exists a feasible \( X \succ 0 \) for (P) or \( y \) such that \( C - \sum_{i=1}^{m} A_i y_i \succ 0 \) in (D).
- If Slater holds for (P), optimal objective of (D) is attained and vice versa.
- Existence of a KKT-point is guaranteed if Slater holds for both, usual assumption of interior-point algorithms for SDP.

**Primal SDP (P)**

\[
\begin{align*}
\inf & \quad C \cdot X \\
\text{s.t.} & \quad A_i \cdot X = b_i \quad \forall \ i \leq m \\
& \quad X \succeq 0
\end{align*}
\]
Strong Duality in SDP

**Dual SDP (D)**

\[
\sup \ b^T y \\
\text{s.t.} \ C - \sum_{i=1}^{m} A_i y_i \succeq 0 \\
y \in \mathbb{R}^m
\]

**Primal SDP (P)**

\[
\inf \ C \cdot X \\
\text{s.t.} \ A_i \cdot X = b_i \quad \forall \ i \leq m \\
X \succeq 0
\]

- Strong Duality holds if Slater condition holds for (P) or (D), i.e., there exists a feasible \(X \succ 0\) for (P) or \(y\) such that \(C - \sum_{i=1}^{m} A_i y_i \succ 0\) in (D).
- If Slater holds for (P), optimal objective of (D) is attained and vice versa.
- Existence of a KKT-point is guaranteed if Slater holds for both, usual assumption of interior-point algorithms for SDP.
- Need to assume this for root node. But is this enough or can these assumptions be lost through branching?
### Theorem [G., Pfetsch, Ulbrich 2016]

Let \( (D_+) \) be the problem formed by adding a linear constraint to \( (D) \). If

- strong duality holds for \( (P) \) and \( (D) \),
- the set of optimal \( Z := C - \sum_{i=1}^{m} A_i y_i \) in \( (D) \) is compact and nonempty,
- the problem \( (D_+) \) is feasible,

then strong duality also holds for \( (D_+) \) and \( (P_+) \) and the set of optimal \( Z \) for \( (D_+) \) is compact and nonempty.
Let \( (D_+) \) be the problem formed by adding a linear constraint to \( (D) \). If

- strong duality holds for \( (P) \) and \( (D) \),
- the set of optimal \( Z := C - \sum_{i=1}^{m} A_i y_i \) in \( (D) \) is compact and nonempty,
- the problem \( (D_+) \) is feasible,

then strong duality also holds for \( (D_+) \) and \( (P_+) \) and the set of optimal \( Z \) for \( (D_+) \) is compact and nonempty.

- Compactness of set of optimal \( Z \) also necessary for strong duality (Friberg 2016)
Strong Duality in Branch-and-Bound

Let \((D_+)\) be the problem formed by adding a linear constraint to \((D)\). If

- strong duality holds for \((P)\) and \((D)\),
- the set of optimal \(Z := C - \sum_{i=1}^{m} A_i y_i\) in \((D)\) is compact and nonempty,
- the problem \((D_+)\) is feasible,

then strong duality also holds for \((D_+)\) and \((P_+)\) and the set of optimal \(Z\) for \((D_+)\) is compact and nonempty.

- Compactness of set of optimal \(Z\) also necessary for strong duality (Friberg 2016)

- Equivalent result for adding linear constraints to \((P)\) with set of optimal \(X\) compact and nonempty and \((P_+)\) feasible
Proposition [G., Pfetsch, Ulbrich 2016]

After adding a linear constraint $\sum_{i=1}^{m} a_i y_i \geq c$ (or $\leq$ or $=$) to (D), if (P) satisfies the Slater condition and the coefficient vector $a$ satisfies $a \in \text{Range}(A)$, for $A : S_n \rightarrow \mathbb{R}^m$, $X \mapsto (A_i \cdot X)_{i \in [m]}$, then the Slater condition also holds for $(P_+)$. 

$\Rightarrow a \in \text{Range}(A)$ is implied by linear independence of $A_i$. 

Dual Slater condition is preserved after adding linear constraint to (P) (without additional assumptions on the coefficients).
Slater Condition in Branch-and-Bound

Proposition [G., Pfetsch, Ulbrich 2016]

After adding a linear constraint $\sum_{i=1}^{m} a_i y_i \geq c$ (or $\leq$ or $=$) to (D), if (P) satisfies the Slater condition and the coefficient vector $a$ satisfies $a \in \text{Range}(A)$, for $A : S_n \rightarrow \mathbb{R}^m$, $X \mapsto (A_i \bullet X)_{i \in [m]}$, then the Slater condition also holds for $(P_+)$.  

▶ $a \in \text{Range}(A)$ is implied by linear independence of $A_i$.  

January 11, 2018 | Applications and Solution Approaches for Mixed-Integer Semidefinite Programming | Tristan Gally | 23
Slater Condition in Branch-and-Bound

Proposition [G., Pfetsch, Ulbrich 2016]

After adding a linear constraint $\sum_{i=1}^{m} a_i y_i \geq c$ (or $\leq$ or $=$) to (D), if (P) satisfies the Slater condition and the coefficient vector $a$ satisfies $a \in \text{Range}(\mathcal{A})$, for $\mathcal{A} : S_n \rightarrow \mathbb{R}^m$, $X \mapsto (A_i \cdot X)_{i \in [m]}$, then the Slater condition also holds for $(P_+)$.

- $a \in \text{Range}(\mathcal{A})$ is implied by linear independence of $A_i$.
- Dual Slater condition is preserved after adding linear constraint to (P) (without additional assumptions on the coefficients).
KKT-points may get lost after branching:

\[(D)\]
\[
\begin{align*}
\sup & 
2y_1 - y_2 \\
\text{s.t.} & 
\begin{pmatrix}
0.5 & -y_1 \\
-y_1 & y_2
\end{pmatrix} \succeq 0,
\end{align*}
\]

\[(P)\]
\[
\begin{align*}
\inf & 
0.5X_{11} \\
\text{s.t.} & 
\begin{pmatrix}
X_{11} & 1 \\
1 & 1
\end{pmatrix} \succeq 0,
\end{align*}
\]

- Strictly feasible solutions given by \(y = (0, 0.5), X_{11} = 2\)
- Optimal objective of 0.5 attained (only) for \(y = (0.5, 0.5), X_{11} = 1\)
KKT-condition in Branch-and-Bound

After branching on $y_2$ and adding cut $y_2 \leq 0$:

$$(D_+)\sup 2y_1 - y_2$$
\[\begin{array}{ccc}
0.5 & -y_1 & 0 \\
-y_1 & y_2 & 0 \\
0 & 0 & -y_2 \\
\end{array}\] \succeq 0,

$$(P_+)\inf 0.5X_{11}$$
\[\begin{array}{ccc}
X_{11} & 1 & X_{13} \\
1 & X_{22} & X_{23} \\
X_{13} & X_{23} & X_{22} - 1 \\
\end{array}\] \succeq 0,

- Optimal objective 0 attained for $y = (0, 0)$
- Relative interior of $(D_+)$ is empty
- $(P_+)$ still has strictly feasible solution $X_{11} = X_{22} = 2$, $X_{13} = X_{23} = 0$
- $(P_+)$ has minimizing sequence $X_{11} = 1/k$, $X_{22} = k$, $X_{13} = X_{23} = 0$
- No longer satisfies assumptions for convergence of interior-point solvers
## Slater Condition in Practice

<table>
<thead>
<tr>
<th>Problem</th>
<th>Dual Slater</th>
<th>Primal Slater</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td>CLS</td>
<td>55.23%</td>
<td>3.26%</td>
</tr>
<tr>
<td>Mk-P</td>
<td>3.66%</td>
<td>65.49%</td>
</tr>
<tr>
<td>TTD</td>
<td>81.99%</td>
<td>5.96%</td>
</tr>
<tr>
<td>Overall</td>
<td>45.16%</td>
<td>26.23%</td>
</tr>
</tbody>
</table>

run on cluster of 40 Intel Xeon E5-1620 3.5 GHz processors with 4 cores and 32GB memory; using SCIP-SDP 3.0.0 and DSDP 5.8; on testset of 194 CBLIB instances
If interior-point solver did not converge for original formulation, solve

\[
\begin{align*}
\inf \quad r \\
\text{s.t.} \quad C - \sum_{i=1}^{m} A_i y_i + l r \succeq 0.
\end{align*}
\]

If optimum \( r^* > 0 \), original problem is infeasible and node can be cut off.
If problem is not infeasible, solve

**Penalty Formulation [Benson, Ye 2008]**

\[
\begin{align*}
\sup & \quad b^\top y - \Gamma r \\
\text{s.t.} & \quad C - \sum_{i=1}^{m} A_i y_i + lr \succeq 0, \\
& \quad r \geq 0
\end{align*}
\]

for sufficiently large \( \Gamma \) to compute an upper bound.

- If optimal \( r^* = 0 \), then solution is also optimal for original problem.
- Adds constraint \( \text{Tr}(X) \leq \Gamma \) to primal problem, for large enough \( \Gamma \) also preserves primal Slater condition.
### Behavior if Slater condition holds for (P) and (D)

<table>
<thead>
<tr>
<th>solver</th>
<th>default</th>
<th>penalty</th>
<th>bound</th>
<th>unsucc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDPA</td>
<td>90.78 %</td>
<td>5.50 %</td>
<td>0.00 %</td>
<td>3.73 %</td>
</tr>
<tr>
<td>DSDP</td>
<td>99.68 %</td>
<td>0.32 %</td>
<td>0.00 %</td>
<td>0.00 %</td>
</tr>
<tr>
<td>MOSEK</td>
<td>99.51 %</td>
<td>0.49 %</td>
<td>0.00 %</td>
<td>0.00 %</td>
</tr>
</tbody>
</table>

### Behavior if Slater condition fails for (P) or (D)

<table>
<thead>
<tr>
<th>solver</th>
<th>default</th>
<th>penalty</th>
<th>bound</th>
<th>unsucc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDPA</td>
<td>56.15 %</td>
<td>1.14 %</td>
<td>13.00 %</td>
<td>29.71 %</td>
</tr>
<tr>
<td>DSDP</td>
<td>99.81 %</td>
<td>0.13 %</td>
<td>0.00 %</td>
<td>0.05 %</td>
</tr>
<tr>
<td>MOSEK</td>
<td>99.20 %</td>
<td>0.79 %</td>
<td>0.01 %</td>
<td>0.00 %</td>
</tr>
</tbody>
</table>

### Behavior if problem is infeasible

<table>
<thead>
<tr>
<th>solver</th>
<th>default</th>
<th>feas check</th>
<th>bound</th>
<th>unsucc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDPA</td>
<td>46.99 %</td>
<td>39.46 %</td>
<td>4.88 %</td>
<td>8.67 %</td>
</tr>
<tr>
<td>DSDP</td>
<td>92.44 %</td>
<td>2.23 %</td>
<td>1.39 %</td>
<td>3.94 %</td>
</tr>
<tr>
<td>MOSEK</td>
<td>88.42 %</td>
<td>10.36 %</td>
<td>1.22 %</td>
<td>0.00 %</td>
</tr>
</tbody>
</table>
Contents

Applications

Solution Approaches
  Outer Approximation
  SDP-based Branch-and-Bound

Warmstarts

Dual Fixing

Solvers for MISDP

Conclusion & Outlook
Warmstarts

- MIP: Large savings by starting dual simplex from optimal basis of parent node.

- Interior-point solvers: Need $X \succ 0$ and $Z := C - \sum_{i=1}^{m} A_i y_i \succ 0$ for initial point.

- Not satisfied by optimal solution of parent node, which will be on boundary.

- Infeasible interior-point methods update $Z$ and $y$ separately, so $Z$ doesn’t necessarily need to be updated after branching, but has to be positive definite.

$\Rightarrow$ Cannot easily warmstart with unadjusted solution of parent node.
Warmstarting Techniques

- Starting from Earlier Iterates
- Convex Combination with Strictly Feasible Solution
- Projection onto Positive Definite Cone
- Rounding Problems
Starting from Earlier Iterates

- Proposed by Gondzio for MIP.

- Store earlier iterate further away from optimum but still sufficiently interior.

- First solve relaxation to sufficiently large gap $\varepsilon_1$ (e.g., $10^{-2}$), then save current iterate and continue solving until original tolerance $\varepsilon_2$ (e.g., $10^{-5}$) is reached.
Convex Combination with Strictly Feasible Solution

- First proposed by Helmberg and Rendl, recently revisited by Skajaa, Andersen and Ye for MIP.

- Take convex combination between optimal solution \((X^*, y^*, Z^*)\) and strictly feasible \((X^0, y^0, Z^0)\).

- Choose \((X^0, y^0, Z^0)\) as default initial point like \((I, 0, I)\), possibly scaled either by maximum entry of primal/dual matrix or maximum of both.

- Also possible to compute analytic center of feasible region once in root node and use this as strictly feasible solution.
Projection onto Positive Definite Cone

- Project optimal solution of parent node onto set of positive definite matrices with $\lambda_{\min} \geq \lambda > 0$.

- For given optimal solution $X^*$ (equivalently $Z^*$) of parent node let $VD\text{diag}(\lambda)V^T = X^*$ be an eigenvector decomposition. Then compute

$$VD\text{diag}((\max\{\lambda_i, \Delta\})_{i \leq n})V^T.$$
Rounding Problems

- Proposed by Çay, Pólik and Terlaky for MISOCOP based on Jordan Frames
- Fix EV decomposition $V \text{Diag}(\lambda^*) V^\top = X^*$ and optimize over eigenvalues
Rounding Problems

- Proposed by Çay, Pólik and Terlaky for MISOCP based on Jordan Frames
- Fix EV decomposition $V \text{Diag}(\lambda^*) V^\top = X^*$ and optimize over eigenvalues
- First solve the linear primal rounding problem

Primal SDP (P)

\[
\begin{align*}
\text{inf} & \quad C \bullet X \\
\text{s.t.} & \quad A_i \bullet X = b_i \quad \forall i \leq m \\
& \quad X \succeq 0
\end{align*}
\]

Primal Rounding Problem (P-R)

\[
\begin{align*}
\text{inf} & \quad C \bullet (V \text{Diag}(\lambda) V^\top) \\
\text{s.t.} & \quad A_i \bullet (V \text{Diag}(\lambda) V^\top) = b_i \quad \forall i \leq m \\
& \quad \lambda_i \geq 0 \quad \forall i \leq n
\end{align*}
\]
Rounding Problems

- Proposed by Çay, Pólik and Terlaky for MISOCP based on Jordan Frames
- Fix EV decomposition $V\text{Diag}(\lambda^*)V^\top = X^*$ and optimize over eigenvalues
- First solve the **linear** primal rounding problem

**Primal SDP (P)**

$$\inf \ C \cdot X$$  
$$\text{s.t.} \quad A_i \cdot X = b_i \quad \forall \ i \leq m$$  
$$X \succeq 0$$

**Primal Rounding Problem (P-R)**

$$\inf \ C \cdot (V\text{Diag}(\lambda)V^\top)$$  
$$\text{s.t.} \quad A_i \cdot (V\text{Diag}(\lambda)V^\top) = b_i \quad \forall \ i \leq m$$  
$$\lambda_i \geq 0 \quad \forall \ i \leq n$$

- $(P-R)$ is restriction of $(P)$ to matrices with same eigenvectors as $X^*$
  $$\Rightarrow \text{optval}(P-R) \geq \text{optval}(P) \geq \text{optval}(D)$$
- $(P-R)$ unbounded $\Rightarrow$ (D) infeasible
- $\text{optval}(P-R) \leq \text{cutoff bound} \Rightarrow$ (D) not optimal
If (D) is not cut off, let $W\text{Diag}(\mu^*)W^\top = Z^*$ be an eigenvector decomposition of the parent node and solve the corresponding linear dual rounding problem:

**Dual SDP (D)**

\[
\begin{align*}
\text{sup} & \quad b^Ty \\
\text{s.t.} & \quad C - \sum_{i=1}^{m} A_iy_i = Z \\
& \quad Z \succeq 0, \quad y \in \mathbb{R}^m
\end{align*}
\]

**Dual Rounding Problem (D-R)**

\[
\begin{align*}
\text{sup} & \quad b^Ty \\
\text{s.t.} & \quad W\text{Diag}(\mu)W^\top + \sum_{i=1}^{m} A_iy_i = C \\
& \quad \mu_i \geq 0 \quad \forall \ i \leq n, \quad y \in \mathbb{R}^m
\end{align*}
\]
Rounding Problems

- If (D) is not cut off, let $W \text{Diag}(\mu^*)W^\top = Z^*$ be an eigenvector decomposition of the parent node and solve the corresponding linear dual rounding problem.

**Dual SDP (D)**

$$\sup \ b^T y$$
$$\text{s.t. } C - \sum_{i=1}^{m} A_i y_i = Z$$
$$Z \succeq 0, \quad y \in \mathbb{R}^m$$

**Dual Rounding Problem (D-R)**

$$\sup \ b^T y$$
$$\text{s.t. } W \text{Diag}(\mu)W^\top + \sum_{i=1}^{m} A_i y_i = C$$
$$\mu_i \geq 0 \quad \forall \ i \leq n, \quad y \in \mathbb{R}^m$$

- Since (D-R) is restriction of (D) to matrices with same eigenvectors as $Z^*$, $\text{optval}(D-R) \leq \text{optval}(D) \leq \text{optval}(P) \leq \text{optval}(P-R)$.

- $\text{optval}(D-R) = \text{optval}(P-R) \quad \Rightarrow \quad$ problem solved to optimality.
If (D) is not cut off, let $W\text{Diag}(\mu^*) W^\top = Z^*$ be an eigenvector decomposition of the parent node and solve the corresponding linear dual rounding problem

**Dual SDP (D)**

\[
\begin{align*}
\sup & \quad b^T y \\
\text{s.t.} & \quad C - \sum_{i=1}^{m} A_i y_i = Z \\
& \quad Z \succeq 0, \quad y \in \mathbb{R}^m
\end{align*}
\]

**Dual Rounding Problem (D-R)**

\[
\begin{align*}
\sup & \quad b^T y \\
\text{s.t.} & \quad W\text{Diag}(\mu) W^\top + \sum_{i=1}^{m} A_i y_i = C \\
& \quad \mu_i \geq 0 \quad \forall \ i \leq n, \quad y \in \mathbb{R}^m
\end{align*}
\]

Since (D-R) is restriction of (D) to matrices with same eigenvectors as $Z^*$,

\[
\text{optval}(D-R) \leq \text{optval}(D) \leq \text{optval}(P) \leq \text{optval}(P-R).
\]

\[
\text{optval}(D-R) = \text{optval}(P-R) \quad \Rightarrow \quad \text{problem solved to optimality}
\]

Otherwise use convex combination to compute strictly feasible initial point.
## Rounding Problems

<table>
<thead>
<tr>
<th>testset</th>
<th>time</th>
<th>roundtime</th>
<th>opt</th>
<th>cutoff</th>
<th>warmstart</th>
<th>pfail</th>
<th>dfail</th>
<th>detected</th>
<th>undetected</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLS</td>
<td>229.38</td>
<td>101.19</td>
<td>0.03</td>
<td>0.68</td>
<td>0.03</td>
<td>0.00</td>
<td>1847.37</td>
<td>310.27</td>
<td>841.17</td>
</tr>
<tr>
<td>MkP</td>
<td>271.18</td>
<td>6.97</td>
<td>0.00</td>
<td>0.40</td>
<td>0.88</td>
<td>0.12</td>
<td>188.18</td>
<td>1.49</td>
<td>459.83</td>
</tr>
<tr>
<td>TT</td>
<td>102.73</td>
<td>17.80</td>
<td>0.02</td>
<td>44.65</td>
<td>284.81</td>
<td>0.00</td>
<td>13,616.42</td>
<td>24.21</td>
<td>1805.33</td>
</tr>
<tr>
<td>CS</td>
<td>166.69</td>
<td>86.72</td>
<td>0.17</td>
<td>6022.54</td>
<td>4794.20</td>
<td>0.00</td>
<td>0.02</td>
<td>0.01</td>
<td>0.37</td>
</tr>
</tbody>
</table>

run on cluster of 40 Intel Xeon E5-1620 3.5 GHz processors with 4 cores and 32GB memory; time limit of 3600 seconds; times as shifted geometric means, SDPs solved using SDPA 7.4.0; $\gamma = 0.5$
## Comparison of Warmstarting Techniques

<table>
<thead>
<tr>
<th>settings</th>
<th>solved</th>
<th>time</th>
<th>sdptier</th>
</tr>
</thead>
<tbody>
<tr>
<td>no warmstart</td>
<td>290</td>
<td>117.85</td>
<td>22,827.93</td>
</tr>
<tr>
<td>unadjusted warmstart</td>
<td>126</td>
<td>821.82</td>
<td>–</td>
</tr>
<tr>
<td>earlier iterate: gap 0.01</td>
<td>172</td>
<td>396.93</td>
<td>–</td>
</tr>
<tr>
<td>earlier iterate: gap 0.5</td>
<td>252</td>
<td>213.88</td>
<td>26,923.91</td>
</tr>
<tr>
<td>convcomb: 0.01 scaled (pdsame) id</td>
<td>288</td>
<td>113.60</td>
<td>19,697.25</td>
</tr>
<tr>
<td>convcomb: 0.5 scaled (pddiff) id</td>
<td>289</td>
<td>108.60</td>
<td>18,307.29</td>
</tr>
<tr>
<td>convcomb: 0.5 scaled (pdsame) id</td>
<td>290</td>
<td>109.92</td>
<td>19,684.70</td>
</tr>
<tr>
<td>convcomb: 0.5 analcent</td>
<td>288</td>
<td>140.21</td>
<td>25,351.48</td>
</tr>
<tr>
<td>projection</td>
<td>289</td>
<td>112.87</td>
<td>20,195.03</td>
</tr>
<tr>
<td>roundingprob 0.5 id</td>
<td>281</td>
<td>180.95</td>
<td>16,955.37</td>
</tr>
<tr>
<td>roundingprob inf only</td>
<td>289</td>
<td>159.66</td>
<td>18,521.50</td>
</tr>
</tbody>
</table>

Run on cluster of 40 Intel Xeon E5-1620 3.5 GHz processors with 4 cores and 32GB memory; time limit of 3600 seconds; times (and iterations) as shifted geometric means (over instances solved by all settings except unadjusted warmstart and preoptimal), SDPs solved using SDPA 7.4.0.
Comparison of Warmstarting Techniques

<table>
<thead>
<tr>
<th>Speedup for conv 0.01 pdsame</th>
<th>Speedup for conv 0.5 pddiff</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>testset</strong></td>
<td><strong>solved</strong></td>
</tr>
<tr>
<td>CLS</td>
<td>0</td>
</tr>
<tr>
<td>MkP</td>
<td>+1</td>
</tr>
<tr>
<td>TT</td>
<td>-3</td>
</tr>
<tr>
<td>CS</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Speedup for conv 0.5 pdsame</th>
<th>Speedup for projection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>testset</strong></td>
<td><strong>solved</strong></td>
</tr>
<tr>
<td>CLS</td>
<td>-1</td>
</tr>
<tr>
<td>MkP</td>
<td>+2</td>
</tr>
<tr>
<td>TT</td>
<td>-1</td>
</tr>
<tr>
<td>CS</td>
<td>0</td>
</tr>
</tbody>
</table>

run on cluster of 40 Intel Xeon E5-1620 3.5 GHz processors with 4 cores and 32GB memory; time limit of 3600 seconds; times (and iterations) as shifted geometric means (over instances solved by all settings except unadjusted warmstart and preoptimal), SDPs solved using SDPA 7.4.0
Contents

Applications

Solution Approaches
  Outer Approximation
  SDP-based Branch-and-Bound

Warmstarts

Dual Fixing

Solvers for MISDP

Conclusion & Outlook
Dual Fixing

- Generalization of reduced-cost fixing for MILPs
- Used for interior-point LP-solvers by Mitchell (1997), primal MISDPs by Helmberg (2000) and general MINLPs by Vigerske (2012)

\[ \text{Theorem } \left[ G., \text{Pfetsch, Ulbrich 2016} \right] \]

\[ (X, W, V): \text{Primal feasible solution, where } W, V \text{ are primal variables corresponding to variable bounds } \ell, u \text{ in the dual} \]

\[ f: \text{Corresponding primal objective value} \]

\[ L: \text{Lower bound on the optimal objective value of the MISDP} \]

Then for every optimal solution of the MISDP

\[ y_j \leq \ell_j + f - L W_j \text{ if } \ell_j > -\infty \] and

\[ y_j \geq u_j - f - L V_j \text{ if } u_j < \infty \]

- If \( f - L < W_j \) for binary \( y_j \), it can be fixed to 0, if \( f - L < V_j \), then \( y_j = 1 \).

6% reduction of B&B-nodes, 26% speedup
Dual Fixing

- Generalization of reduced-cost fixing for MILPs
- Used for interior-point LP-solvers by Mitchell (1997), primal MISDPs by Helmberg (2000) and general MINLPs by Vigerske (2012)

Theorem [G., Pfetsch, Ulbrich 2016]

- \((X, W, V)\): Primal feasible solution, where \(W, V\) are primal variables corresponding to variable bounds \(\ell, u\) in the dual
- \(f\): Corresponding primal objective value
- \(L\): Lower bound on the optimal objective value of the MISDP

Then for every optimal solution of the MISDP

\[
y_j \leq \ell_j + \frac{f - L}{W_{jj}} \quad \text{if } \ell_j > -\infty \quad \text{and} \quad y_j \geq u_j - \frac{f - L}{V_{jj}} \quad \text{if } u_j < \infty
\]

- If \(f - L < W_{jj}\) for binary \(y_j\), it can be fixed to 0, if \(f - L < V_{jj}\), then \(y_j = 1\).
Dual Fixing

- Generalization of reduced-cost fixing for MILPs
- Used for interior-point LP-solvers by Mitchell (1997), primal MISDPs by Helmberg (2000) and general MINLPs by Vigerske (2012)

**Theorem [G., Pfetsch, Ulbrich 2016]**

- \((X, W, V)\): Primal feasible solution, where \(W, V\) are primal variables corresponding to variable bounds \(\ell, u\) in the dual
- \(f\): Corresponding primal objective value
- \(L\): Lower bound on the optimal objective value of the MISDP

Then for every optimal solution of the MISDP

\[
y_j \leq \ell_j + \frac{f - L}{W_{jj}} \quad \text{if} \quad \ell_j > -\infty \quad \text{and} \quad y_j \geq u_j - \frac{f - L}{V_{jj}} \quad \text{if} \quad u_j < \infty
\]

- If \(f - L < W_{jj}\) for binary \(y_j\), it can be fixed to 0, if \(f - L < V_{jj}\), then \(y_j = 1\).
- 6% reduction of B&B-nodes, 26% speedup
Contents

Applications

Solution Approaches
  Outer Approximation
  SDP-based Branch-and-Bound

Warmstarts

Dual Fixing

Solvers for MISDP

Conclusion & Outlook
MISDP-Solvers

Nonlinear branch-and-bound
- SCIP-SDP 3.1 (nonlinear branch-and-bound)
  - Our implementation, using SCIP as B&B-framework
- YALMIP-BNB R20170921
  - MATLAB toolbox for rapid prototyping

Cutting plane / outer approximation approaches
- SCIP-SDP 3.1 (LP-based cutting planes)
- YALMIP-CUTSDP R20170921
- Pajarito 0.5
  - Julia implementation for mixed-integer convex including MISDP
  - MIP-solver-drives version (single MIP with SDP solves for stronger cuts)
Which SDP-Solver to use for the Relaxations?

<table>
<thead>
<tr>
<th>solver</th>
<th>solved</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDPA</td>
<td>161</td>
<td>136.3</td>
</tr>
<tr>
<td>DSDP</td>
<td>175</td>
<td>157.3</td>
</tr>
<tr>
<td>MOSEK</td>
<td>187</td>
<td>64.9</td>
</tr>
</tbody>
</table>
Comparison of MISDP-Solvers

<table>
<thead>
<tr>
<th>solver</th>
<th>CLS solved</th>
<th>CLS time</th>
<th>Mk-P solved</th>
<th>Mk-P time</th>
<th>TTD solved</th>
<th>TTD time</th>
<th>Total solved</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCIP-SDP(NL-BB)</td>
<td>63</td>
<td>104.3</td>
<td>68</td>
<td>38.3</td>
<td>57</td>
<td>65.2</td>
<td>188</td>
<td>63.9</td>
</tr>
<tr>
<td>YALMIP(BNB)</td>
<td>62</td>
<td>195.9</td>
<td>64</td>
<td>61.7</td>
<td>36</td>
<td>537.5</td>
<td>162</td>
<td>181.5</td>
</tr>
<tr>
<td>SCIP-SDP(Cut-LP)</td>
<td>65</td>
<td>8.3</td>
<td>31</td>
<td>614.8</td>
<td>39</td>
<td>230.5</td>
<td>135</td>
<td>132.6</td>
</tr>
<tr>
<td>YALMIP(CUTSDP)</td>
<td>31</td>
<td>525.1</td>
<td>15</td>
<td>1145.7</td>
<td>22</td>
<td>945.9</td>
<td>68</td>
<td>832.0</td>
</tr>
<tr>
<td>Pajarito</td>
<td>65</td>
<td>64.2</td>
<td>13</td>
<td>1577.5</td>
<td>43</td>
<td>220.7</td>
<td>121</td>
<td>303.3</td>
</tr>
</tbody>
</table>

run on 8-core Intel i7-4770 CPU with 3.4 GHz and 16GB memory; time limit of 3600 seconds, times as shifted geometric means, SDPs solved using MOSEK 8.1.0.25, MIPs/LPs using CPLEX 12.6.1
Applications

Solution Approaches
  Outer Approximation
  SDP-based Branch-and-Bound

Warmstarts

Dual Fixing

Solvers for MISDP

Conclusion & Outlook
Conclusion

- MISDPs can be solved by generic framework
- Primal Slater condition inherited in MISDP, Dual Slater may get lost
- Warmstarting is possible and can help for some applications
Conclusion & Outlook

Conclusion

- MISDPs can be solved by generic framework
- Primal Slater condition inherited in MISDP, Dual Slater may get lost
- Warmstarting is possible and can help for some applications

Outlook

- Cutting Planes
  - Chvátal-Gomory / knapsack cuts portable to MISDP, but generation less clear
Conclusion & Outlook

Conclusion

- MISDPs can be solved by generic framework
- Primal Slater condition inherited in MISDP, Dual Slater may get lost
- Warmstarting is possible and can help for some applications

Outlook

- Cutting Planes
  - Chvátal-Gomory / knapsack cuts portable to MISDP, but generation less clear
- Facial Reduction
  - Project on minimal face of psd-cone if Slater condition fails
  - Projection as solution of homogeneous self-dual model
  - Optimization over face of psd-cone can again be formulated as SDP
SCIP-SDP is available in source code at

http://www.opt.tu-darmstadt.de/scipsdp/

Thank you for your attention!
Aharon Ben-Tal and Arkadi Nemirovski.  
Robust truss topology design via semidefinite programming.  

Aharon Ben-Tal and Arkadi Nemirovski.  
On polyhedral approximations of the second-order cone.  

Gábor Braun, Samuel Fiorini, Sebastian Pokutta, and David Steurer.  
Approximation limits of linear programs (beyond hierarchies).  

Mehmet T. Çezik and Garud Iyengar.  
Cuts for mixed 0-1 conic programming.  

Alexander Engau.  

S. Foucart and M.-J. Lai.  
Sparsest solutions of underdetermined linear systems via $\ell_q$-minimization for $0 < q \leq 1$.  

Henrik A. Friberg.  
CBLIB 2014: A benchmark library for conic mixed-integer and continuous optimization.  

Henrik A. Friberg.  
Facial reduction heuristics and the motivational example of mixed-integer conic optimization.  

Tristan Gally and Marc E. Pfetsch.  
Computing restricted isometry constants via mixed-integer semidefinite programming.  
Tristan Gally, Marc E. Pfetsch, and Stefan Ulbrich.
A framework for solving mixed-integer semidefinite programs.
*Optimization Methods and Software, 2017.*
To Appear.

Ambros Gleixner, Leon Eifler, Tristan Gally, Gerald Gamrath, Patrick Gemander, Robert Lion Gottwald, Gregor Hendel, Christopher Hojny, Thorsten Koch, Matthias Miltenberger, Benjamin Müller, Marc E. Pfetsch, Christian Puchert, Daniel Rehfeldt, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Jan Merlin Viernickel, Stefan Vigerske, Dieter Weninger, Jonas T. Witt, and Jakob Witzig.
The SCIP optimization suite 5.0.
ZIB report 17-12, 2017.

Jacek Gondzio.
Warm start of the primal-dual method applied in the cutting plane scheme.

Christoph Helmberg and Franz Rendl.
Solving quadratic (0,1)-problems by semidefinite programs and cutting planes.

IBM.
*IBM ILOG CPLEX Optimization Studio.*

J.E. Kelley Jr.
The cutting-plane method for solving convex programs.

Michal Kočvara.
Truss topology design with integer variables made easy.

N. Krislock, J. Malick, and F. Roupin.

M. Laurent and S. Poljak.
On a positive semidefinite relaxation of the cut polytope.
*Linear Algebra and its Applications, 223/224, 1995.*

J. Löfberg.
YALMIP: a toolbox for modeling and optimization in MATLAB.
Miles Lubin.  
*Mixed-integer convex optimization: outer approximation algorithms and modeling power.*  

Miles Lubin, Emre Yaman, Russell Bent, and Juan Pablo Vielma.  

Sonja Mars.  
*Mixed-Integer Semidefinite Programming with an Application to Truss Topology Design.*  

MOSEK ApS.  

Frank Permenter, Henrik A. Friberg, and Erling D. Andersen.  
Solving conic optimization problems via self-dual embedding and facial reduction: a unified approach.  

S. Poljak and F. Rendl.  
Nonpolyhedral relaxations of graph-bisection problems.  

I. Quesada and I.E. Grossmann.  
An LP/NLP based branch and bound algorithm for convex MINLP optimization problems.  

Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele.  
Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations.  

Anders Skajaa, Erling D. Andersen, and Yinyu Ye.  
Warmstarting the homogeneous and self-dual interior point method for linear and conic quadratic problems.  