TECHNISCHE UNIVERSITÄT DARMSTADT

Solving Mixed-Integer Semidefinite Programs

Marc Pfetsch, TU Darmstadt

Discrete Optimization

Overview

Basic Definitions

TECHNISCHE UNIVERSITÄT DARMSTADT

Recall:

- Let $M \in \mathbb{R}^{n \times n}$ be a symmetric matrix.
- Note: All matrices in this talk will be symmetric.
- M is positive semidefinite (psd), denoted $M \succeq 0$, if $x^\top M x \geq 0$ for all $x \in \mathbb{R}^n$.
- This is equivalent to M having nonnegative eigenvalues.
- $M \succeq 0$ if and only if there exists $S \in \mathbb{R}^{r \times n}$ with $M = S^\top S$.
- $S^n_+ \coloneqq \{M \in \mathbb{R}^{n \times n} : M \text{ symmetric and } M \succeq 0\}$ is a closed convex cone.
- For two matrices $A, B \in \mathbb{R}^{n \times n}$, $\langle A, B \rangle \coloneqq \sum_{ij} A_{ij} B_{ij}$ is the inner product.
- If $A, B \succeq 0$ then $\langle A, B \rangle \geq 0$.
- M is positive definite, denoted $A \succ 0$, if $x^\top M x > 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$.
- $[m] \coloneqq \{1, \ldots, m\}$ for $m \in \mathbb{N}$.

From Linear to Semidefinite Programs

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Let $a^0, \dots, a^m \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$.

Primal/dual pair of linear program:

$$
(P)\quad \max \quad (a^0)^\top x \quad \text{s.t.} \quad (a^k)^\top x = b_k \quad \forall k \in [m], \quad x \geq 0.
$$

$$
(D)\quad \min \quad b^\top y \quad \text{s.t.} \quad \sum_{k \in [m]} a^k y_k - a^0 \geq 0, \quad y \in \mathbb{R}^m.
$$

From Linear to Semidefinite Programs

TECHNISCHE UNIVERSITÄT DARMSTADT

Let $a^0, \ldots, a^m \in \mathbb{R}^n$ and $b \in \mathbb{R}^m$.
Primal/dual pair of linear program:

$$(P) \quad \max \quad (a^0)^\top x \quad \text{s.t.} \quad (a^k)^\top x = b_k \quad \forall k \in [m], \quad x \geq 0.$$

$$(D) \quad \min \quad b^\top y \quad \text{s.t.} \quad \sum_{k \in [m]} a^k y_k - a^0 \geq 0, \quad y \in \mathbb{R}^m.$$

Generalize to matrices: Let $A^0, \ldots, A^m \in \mathbb{R}^{n \times n}$ be (symmetric) matrices, $b \in \mathbb{R}^m$.
Primal/dual Pair of semidefinite program (SDP):

$$(P) \quad \sup \quad \langle A^0, X \rangle \quad \text{s.t.} \quad \langle A^k, X \rangle = b_k \quad \forall k \in [m], \quad X \succeq 0.$$

$$(D) \quad \inf \quad b^\top y \quad \text{s.t.} \quad \sum_{k \in [m]} A^k y_k - A^0 \succeq 0, \quad y \in \mathbb{R}^m.$$

LPs are a special case of SDPs for diagonal matrices.

Duality for Semidefinite Programs

$$\text{primal } (P): \quad \sup\{\langle A^0, X\rangle \,:\, \langle A^k, X\rangle = b_k,\ \forall k \in [m],\ X \succeq 0\},$$

$$\text{dual } (D): \quad \inf\{b^\top y \,:\, \sum_{k\in[m]} A^k y_k - A^0 \succeq 0,\ y \in \mathbb{R}^m\}.$$

- Weak duality: $\hat{X}$ and $\hat{y}$ feasible for (P) and (D), resp. Then

$$0 \leq \langle \sum_{k\in[m]} A^k \hat{y}_k - A^0, \hat{X}\rangle = \sum_{k\in[,]}^{m} b_k \hat{y}_k - \langle A^0, \hat{X}\rangle = b^\top \hat{y} - \langle A^0, \hat{X}\rangle,$$

  which is equivalent to $\langle A^0, \hat{X}\rangle \leq b^\top \hat{y}$.
- Strong Duality holds if Slater condition holds for (P) or (D):
  $\exists\, X \succ 0$ feasible for (P) or y such that $\sum_{k\in[m]} A^k y_k - A^0 \succ 0$ in (D).
- If Slater condition holds for (P), optimal objective of (D) is attained and vice versa.

Solving Semidefinite Programs

TECHNISCHE UNIVERSITÄT DARMSTADT

- SDPs can be solved in polynomial time up to a given accuracy $\varepsilon > 0$, e.g., by interior point solvers.
- Existence of a KKT-point is guaranteed if Slater condition holds for (P) and (D). This is assumed by most interior-point SDP-solvers.
- No “combinatorial algorithm” is known for SDPs.
- Restarting interior point solvers is notoriously hard as compared to hot starting the simplex algorithm.
- As a consequence, the solution of SDPs is much more time consuming (currently a factor of 10 to 100 slower).

Mixed-Integer Semidefinite Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

Mixed-Integer Semidefinite Program (MISDP)

$$\begin{aligned}
\inf \quad & b^\top y \\
\text{s.t.} \quad & \sum_{k=1}^{m} A^k y_k - A^0 \succeq 0, \\
& \ell_i \leq y_i \leq u_i && \forall i \in [m], \\
& y_i \in \mathbb{Z} && \forall i \in I,
\end{aligned}$$

- symmetric matrices $A^k \in \mathbb{R}^{n \times n}$ for $k \in [m]_0 := \{0, \dots, m\}$, $b \in \mathbb{R}^m$,
- bounds: $\ell_i \in \mathbb{R} \cup \{-\infty\}$, $u_i \in \mathbb{R} \cup \{\infty\}$ for all $i \in [m]$,
- integer variables: $I \subseteq [m]$.

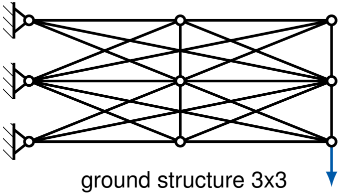
Mixed Integer Programs (MIPs) are a special case.

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Robust Truss Topology Design

- n nodes $V \subset \mathbb{R}^d$
- n_f free nodes $V_f \subset V$
- m possible bars E
- force $f \in \mathbb{R}^{d_f}$ for $d_f = d \cdot n_f$



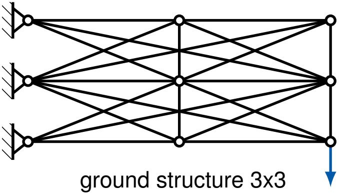
ground structure 3x3

Robust Truss Topology Design

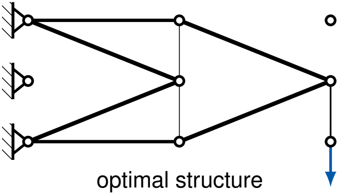
TECHNISCHE UNIVERSITÄT DARMSTADT

- n nodes $V \subset \mathbb{R}^d$
- n_f free nodes $V_f \subset V$
- m possible bars E
- force $f \in \mathbb{R}^{d_f}$ for $d_f = d \cdot n_f$

- Cross-sectional areas $x \in \mathbb{R}^m_+$ for bars minimizing 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



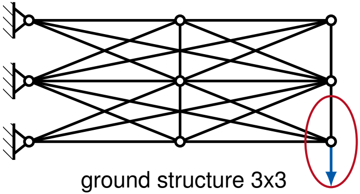
optimal structure

Robust Truss Topology Design

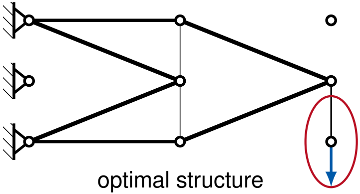
TECHNISCHE UNIVERSITÄT DARMSTADT

- n nodes $V \subset \mathbb{R}^d$
- n_f free nodes $V_f \subset V$
- m possible bars E
- force $f \in \mathbb{R}^{d_f}$ for $d_f = d \cdot n_f$

- Cross-sectional areas $x \in \mathbb{R}^m_+$ for bars minimizing 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

- Use uncertainty set $\{f \in \mathbb{R}^{d_f} \;:\; f = Qg : \|g\|_2 \leq 1\}$ instead of single force f.
- Instead of arbitrary cross-sections $x \in \mathbb{R}^m_+$ restrict them to discrete set $\mathcal{A}$.

Robust Truss Topology Design

TECHNISCHE UNIVERSITÄT DARMSTADT

Elliptic Robust Discrete TTD [Ben-Tal/Nemirovski 1997; Mars 2013]

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

with bar lengths ℓ_e, upper bound $C_{\max}$ on compliance and stiffness matrix

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

for positive semidefinite, rank-one single bar stiffness matrices A_e.

Cardinality Constrained Least Squares

- Sample points as rows of $A \in \mathbb{R}^{m \times d}$ with measurements $b_1, \ldots, b_m \in \mathbb{R}$
- Find $x \in \mathbb{R}^d$ minimizing $\frac{1}{2}\|Ax - b\|_2^2 + \frac{\rho}{2}\|x\|_2^2$ for a regularization parameter ρ.
- Further restrict x to at most k non-zero components.

Cardinality Constrained Least Squares [Pilanci/Wainwright/El Ghaoui 2015]

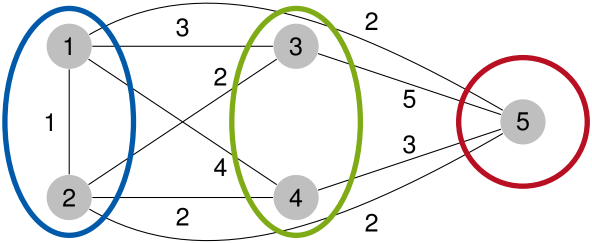
$$
\begin{aligned}
\inf \quad & \tau \\
\text{s.t.} \quad & \begin{pmatrix} I + \frac{1}{\rho} A \operatorname{Diag}(z) A^\top & b \\ b^\top & \tau \end{pmatrix} \succeq 0, \\
& \sum_{j=1}^{d} z_j \leq k, \quad z \in \{0,1\}^d.
\end{aligned}
$$

Minimum k-Partitioning

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Given undirected graph $G = (V, E)$, edge costs c and number of parts $k \in \mathbb{N}$.
- Find partitioning of V into k disjoint sets $V_1, \ldots, V_k$ minimizing the total cost within the parts

$$\sum_{i=1}^{k} \sum_{e \in E[V_i]} c(e).$$



- Applications in, e.g., frequency planning and layout of electronic circuits.

Minimum k-Partitioning [Eisenblätter 2001]

$$\begin{aligned} \inf \quad & \sum_{1 \le i < j \le n} c_{ij} \, Y_{ij} \\ \text{s.t.} \quad & \tfrac{-1}{k-1} \, J + \tfrac{k}{k-1} \, Y \succeq 0, \\ & Y_{ii} = 1, \; Y_{ij} \in \{0, 1\}, \end{aligned}$$

where J is the all-one matrix.

Constraints on the size of the partitions can be added as

$$\ell \le \sum_{j=1}^{n} w_j \, Y_{ij} \le u \qquad \forall i \le n,$$

with w_j weight of node j and ℓ and u bounds on total weight of each partition.

Further Applications

- Computing restricted isometry constants in compressed sensing
- Optimal transmission switching problem in AC power flow
- Robustification of physical parameters in gas networks
- Subset selection for eliminating multicollinearity
- ...

Overview

Solving Methods for MISDPs

TECHNISCHE UNIVERSITÄT DARMSTADT

1. SDP-based branch-and-bound: Solve SDP-relaxations (special case of NLP-based B&B [Dakin 1965])
2. LP-based branch-and-bound: Cutting plane method based on LP-relaxations [Sherali and Fraticelli 2002]; [Krishnan and Mitchell 2006]
3. Outer approximation: Solve MIP-relaxations [Duran and Grossmann 1986].

Implementations:

1. YALMIP [Löfberg 2004] and SCIP-SDP
2. YALMIP and SCIP-SDP
3. Pajarito [Coey, Lubin, and Vielma 2020]

SDP-based Branch-and-Bound

TECHNISCHE UNIVERSITÄT DARMSTADT

- Relax integrality.
- Branch on integral y-variables.
- 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 solvers.
- Convergence assumptions of SDP-solvers should be satisfied.
- Usually much slower than solving LPs and no warmstart.

LP-based Approach

TECHNISCHE UNIVERSITÄT DARMSTADT

For LP-based approach and outer approximation:

- Usual approach for convex MINLP:
  $g(Z) = -\lambda_{\min}(Z)$. Then $Z \succeq 0 \Leftrightarrow g(Z) \leq 0$. Use gradient cuts

$$g(\overline{Z}) + \nabla g(\overline{Z})^\top (Z - \overline{Z}) \leq 0.$$

- But function of smallest eigenvalue is not differentiable everywhere.
- Instead use characterization $Z \succeq 0 \quad \Leftrightarrow \quad u^\top Z\, u \geq 0$ for all $u \in \mathbb{R}^n$.
- If $Z \coloneqq \sum_{k=1}^m A^k\, y_k^* - A^0 \not\succeq 0$, compute eigenvector v to smallest eigenvalue. Then

$$v^\top Z\, v = \sum_{k=1}^m v^\top A^k v\, y_k - v^\top A^0 v \geq 0$$

  is valid and cuts off y^* $\quad\rightarrow$ Eigenvector cut.

Cutting Planes: MISOCP vs. MISDP

TECHNISCHE UNIVERSITÄT DARMSTADT

- Cutting planes are often used by solvers for mixed-integer second-order cone problems (MISOCPs).
- Approximation for SOCPs possible with polynomial number of cuts [Ben-Tal/Nemirovski 2001].
- Approximation for SDPs needs exponential number of cuts:

Theorem ([Braun, Fiorini, Pokutta, Steurer 2015])

There are SDPs of dimension $n \times n$ for which any polyhedral approximation is of size $2^{\Omega(n)}$.

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

SCIP-SDP

TECHNISCHE UNIVERSITÄT DARMSTADT

Our solver: SCIP-SDP

- Based on SCIP (`www.scipopt.org`)
- Supports both SDP-based B&B and LP-based branch-and-cut.
- Introduced by [Mars 2013], continued by [Gally 2019] and Matter [2022], ...
- Apache 2.0 license.
- Current version 4.3: `wwwopt.mathematik.tu-darmstadt.de/scipsdp`, `https://github.com/scipopt/SCIP-SDP`
- Approximately 50 000 lines of C-code
- SDP-solvers: interfaces to Mosek, DSDP, SDPA
- Matlab-Interface: `github.com/scipopt/MatlabSCIPInterface`
- File formats: SDPA-format and CBF
- Parallelized version available as UG-MISDP.
- Supports rank 1 constraints.

Components (Plugins) of SCIP-SDP

TECHNISCHE UNIVERSITÄT DARMSTADT

- Constraint handler for dual form: $\sum_{k\in[m]}^{m} A^k y_k - A^0 \succeq 0$.
- Two MISDP heuristics: SDP-based diving, SDP-based randomized rounding
- Several presolving methods: Add linear constraints implied by SDP-constraint during presolving, e.g., non-negativity of diagonal entries. (See below.)
- Several MISDP propagators: dual fixing, minor propagation. (See below.)
- Relaxator solves trivial relaxations (e.g., all variables fixed), otherwise calls SDP interface (SDPI).
- Upper level SDPI does some local presolving important for SDP-solvers, e.g., removing fixed variables and removing zero rows/columns.
- Lower level SDPI brings SDP into the form needed by the solver (e.g., primal instead of dual SDP for MOSEK) and solves it.
- In case SDP-solver failed to converge (e.g., because of failure of constraint qualification), upper level SDPI can apply penalty formulation and call lower level SDPI for adjusted problem.

For Computations . . .

TECHNISCHE UNIVERSITÄT DARMSTADT

- Use SCIP developer version (8.0.3).
- Use Mosek 9.2.40 for solving SDP-relaxations.
- Linux cluster with 3.5 GHz Intel Xeon E5-1620 Quad-Core CPUs.
- Nodes and times are shifted geometric means, time limit 1 h.

Comparison of SDP and LP-based Approach

TECHNISCHE UNIVERSITÄT DARMSTADT

Testset: 185 instances from different sources.

type	# solved	# nodes	time
SDP	167	1066.1	132.2
LP	109	419.2	336.5
all optimal (106):			
SDP		605.0	93.2
LP		507.0	63.2

Conclusions:

- ▷ LP-based approach solves significantly less instances.
- ▷ On the instances solved by both, it is faster by 32 % and uses less nodes.
- ▷ Open question: Predict which method is faster and explain why.

Comparison of MISDP Solvers I

TECHNISCHE UNIVERSITÄT DARMSTADT

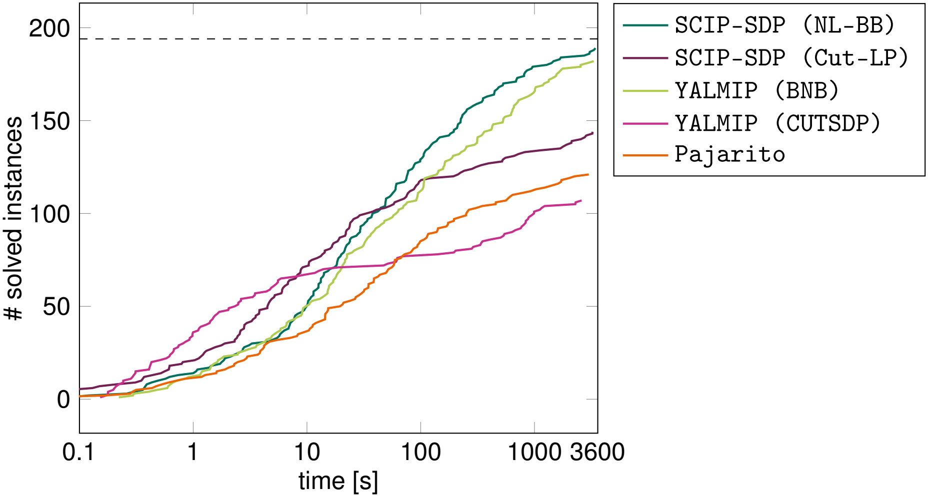
A bit older comparison:

solver	TTD		CLS		M*k*-P		Total	
	opt	time	opt	time	opt	time	opt	time
`SCIP-SDP (NL-BB)`	57	64.4	63	94.3	69	36.4	189	60.4
`SCIP-SDP (Cut-LP)`	44	143.6	65	9.0	35	640.3	144	117.5
`YALMIP (BNB)`	52	203.0	62	132.0	68	25.2	182	88.1
`YALMIP (CUTSDP)`	22	1026.8	58	33.1	27	657.2	107	295.5
`Pajarito`	43	190.9	65	54.3	13	1503.5	121	271.2

run on 8-core Intel i7-4770 CPU with 3.4 GHz and 16GB memory over 196 instances of CBLIB; time limit of 3600 seconds, times as shifted geometric means, SDPs solved using MOSEK 8.1.0.54, MIPs/LPs using CPLEX 12.6.1; all solvers single-threaded; SCIP-SDP 3.1.1 (LP-based cutting planes), YALMIP-CUTSDP R20180926, Pajarito 0.5.0

Comparison of MISDP Solvers I

TECHNISCHE UNIVERSITÄT DARMSTADT



Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Dual Fixing

TECHNISCHE UNIVERSITÄT DARMSTADT

- Extension of reduced-cost fixing to general MINLPs by [Ryoo and Sahinidis 1996] and primal MISDPs by [Helmberg 2000].
- Our approach uses conic duality and only requires feasibility.

Theorem [Gally, P., Ulbrich 2018]

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

Then for every optimal solution $y^\star$ of the MISDP

$$y_j^\star \leq \ell_j + \frac{U - f}{W_{jj}} \text{ if } \ell_j > -\infty \quad \text{and} \quad y_j^\star \geq u_j - \frac{U - f}{V_{jj}} \text{ if } u_j < \infty.$$

- For binary y_j: If $U - f < W_{jj}$, then $y_j^\star = 0$, if $U - f < V_{jj}$, then $y_j^\star = 1$.
- 9% reduction of B&B-nodes, 23% speedup.

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Bound Tightening

TECHNISCHE UNIVERSITÄT DARMSTADT

For an index $k \in [m]$, define

$$P_k \coloneqq \{i \in [m] \setminus \{k\} \,:\, A^i \succeq 0\}, \qquad N_k \coloneqq \{i \in [m] \setminus \{k\} \,:\, A^i \preceq 0\},$$

as well as

$$\underline{\mu}_k \coloneqq \inf \Big\{\mu \,:\, A^k \mu + \sum_{i \in P_k} A^i u_i + \sum_{j \in N_k} A^j \ell_j - A^0 \succeq 0\Big\},$$

$$\overline{\mu}_k \coloneqq \sup \Big\{\mu \,:\, A^k \mu + \sum_{i \in P_k} A^i u_i + \sum_{j \in N_k} A^j \ell_j - A^0 \succeq 0\Big\}$$

or $\pm\infty$ if $\pm\infty$ occurs in bounds (ℓ, u).

Lemma (Tighten Bounds (TB))

Let all matrices be (positive or negative) semidefinite. Then, $\underline{\mu}_k \leq y_k \leq \overline{\mu}_k$ is valid for all $k \in [m]$. We can round bounds for integral variables.

One-Variable SDPs

- For computing bound tightenings, need to solve one-variable SDPs.

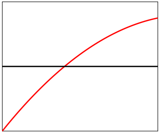
$$\inf \{\mu \,:\, \mu A - B \succeq 0,\ \ell \leq \mu \leq u\}.$$

  for symmetric $A, B \in \mathbb{R}^{n \times n}$.
- Can easily see: $\mu \mapsto \lambda_{\min}(\mu A - B)$ is concave.
- Let $\hat{v}$ be a unit eigenvector for $\lambda_{\min}(\hat{\mu} A - B)$ for $\hat{\mu} \in \mathbb{R}$. Then $\hat{v}^\top A \hat{v}$ is a supergradient, i.e.,

$$\lambda_{\min}(\mu A - B) \leq \lambda_{\min}(\hat{\mu} A - B) + (\mu - \hat{\mu})\, \hat{v}^\top A \hat{v}$$

  for all $\mu \in \mathbb{R}$.

- Goal: Want increase μ from ℓ until $\lambda_{\min}(\mu A - B) = 0$.
- Yields semismooth Newton algorithm …

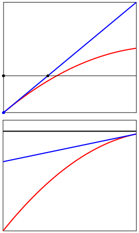


One-Variable SDPs

TECHNISCHE UNIVERSITÄT DARMSTADT

v_k = eigenvector for $\lambda_k \coloneqq \lambda_{\min}(A\mu_k - B)$

$\mu_{k+1} = \mu_k - \frac{\lambda_k}{(v^k)^\top A v^k}$



Handle easy cases, e.g., infeasible if $\lambda_{\min}(A\,u - B) < 0$, supergradient positive.

- Always converges.
- Converges Q-superlinearly to a zero $\mu^\star$ of $f(\mu) = \lambda_{\min}(\mu\,A - B)$, given that $\partial f(\mu^\star)$ is nonsingular and the starting point lies near $\mu^\star$ [Qi and Sun, 1993].
- Very fast in practice; bottleneck: eigenvector computation . . .

Condensed Computational Results

TECHNISCHE UNIVERSITÄT DARMSTADT

Testset with 185 instances, results from [Matter and P. 2023]:

Setting	# solved	# nodes	time
nopresol	168	1405.3	180.23
bound tightening	167	1297.6	152.43
MIX	167	1085.2	139.52

- Bound tightening applied in every node produces a speed-up of about 7 %.
- MIX includes bound tightening and several other methods.
  It produces a speed-up of about 22 %.

- Some techniques do not do anything on some instances.
- The methods are effective if they can be applied and induce a small time overhead.

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

1 Preliminaries
2 Applications
3 Solution Methods
4 SCIP-SDP
5 Dual Fixing
6 Presolving MISDPs
7 **Conflict Analysis**
8 Symmetry
9 Conclusion & Outlook

Conflict Analysis I

TECHNISCHE UNIVERSITÄT DARMSTADT

- The original idea is to learn from infeasible nodes in a branch-and-bound-tree.
- Idea transferred from SAT-solving to MIPs by [Achterberg 2007].
- More generally, can be seen as a way to learn cuts from solutions of the duals $\rightarrow$ similar to "dual ray/solution analysis" [Witzig et al. 2017, Witzig 2021].

Conflict Analysis II

TECHNISCHE UNIVERSITÄT DARMSTADT

Consider

$$\inf \{b^\top y \;:\; A(y) \coloneqq \sum_{k=1}^{m} A^k y_k - A^0 \succeq 0,\; Dy \geq d,\; \ell \leq y \leq u\}$$

and $\hat{X} \succeq 0$, $\hat{z} \geq 0$. Aggregation yields:

$$\langle A(y), \hat{X} \rangle + \hat{z}^\top Dy \;\geq\; \hat{z}^\top d.$$

Idea: Do not add this (redundant) inequality, but perform bound propagation, taking integrality conditions into account.

Conflict Analysis III

TECHNISCHE UNIVERSITÄT DARMSTADT

The dual can provide $(\hat{X}, \hat{z}, \hat{r}^\ell, \hat{r}^u)$:

$$\begin{aligned} \sup \quad & \langle A^0, X\rangle + z^\top d + \ell^\top r^\ell - u^\top r^u \\ \text{s.t.} \quad & \langle A^j, X\rangle + (D^\top z)_j + r^\ell_j - r^u_j = b_j \quad \forall j \in [m], \\ & X \succeq 0,\ z,\ r^\ell,\ r^u \geq 0. \end{aligned}$$

Similarly for a primal ray satisfying:

$$\begin{aligned} & \langle A^j, X\rangle + (D^\top z)_j + r^\ell_j - r^u_j = 0 \qquad \forall j \in [m], \\ & \langle A^0, X\rangle + d^\top z + \ell^\top r^\ell - u^\top r^u > 0, \\ & X \succeq 0,\ z,\ r^\ell,\ r^u \geq 0. \end{aligned}$$

Lemma

Let $(\hat{X}, \hat{z}, \hat{r}^\ell, \hat{r}^u)$ be a primal ray. Then the aggregated inequality is infeasible with respect to the local bounds ℓ and u.

Conflict Analysis – Computations

TECHNISCHE UNIVERSITÄT DARMSTADT

Generate a conflict constraint for each feasible or infeasible node. Store them as constraints and perform bound propagation.

type	# solved	# nodes	time
default	167	1066.1	132.2
conflicts	168	989.6	122.2
all optimal (167):			
default		788.7	94.2
conflicts		726.3	86.4

- Using conflicts provides a speed-up and node-reduction of about 8 %.
- Average number of conflict constraints per node: 1.25
  (Note that we also run in heuristics and we do not count nodes of heuristics).

Overview

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Symmetry Detection

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Goal: apply known symmetry handling methods.

For a permutation σ of $[n]$:

$$\sigma(A)_{ij} = A_{\sigma^{-1}(i),\sigma^{-1}(j)} \quad \forall i, j \in [n].$$

Definition

A permutation $\pi \in \mathcal{S}_m$ of variables is a formulation symmetry if there exists a permutation $\sigma \in \mathcal{S}_n$ such that

1. $\pi(I) = I$, $\pi(\ell) = \ell$, $\pi(u) = u$, and $\pi(b) = b$
   (π leaves integer variables, variable bounds, and the objective coefficients invariant),
2. $\sigma(A^0) = A^0$ and, for all $i \in [m]$, $\sigma(A^i) = A^{\pi^{-1}(i)}$.

Such symmetries can be detected by using graph automorphism algorithms.

Symmetry: Computed Symmetries

TECHNISCHE UNIVERSITÄT DARMSTADT

instance	symmetry group
0+-115305C_MISDPld000010	$\mathcal{S}_2$
0+-115305C_MISDPrd000010	$\mathcal{S}_2$
band60605D_MISDPld000010	$\mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_{10} \times \mathcal{S}_3 \times \mathcal{S}_4$
band60605D_MISDPrd000010	$\mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_{10} \times \mathcal{S}_3 \times \mathcal{S}_4$
band70704A_MISDPld000010	$\mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_3 \times \mathcal{S}_3$
band70704A_MISDPrd000010	$\mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_3 \times \mathcal{S}_3$
clique_60_k10_6_6, clique_60_k15_4_4, clique_60_k20_3_3, clique_60_k4_15_15, clique_60_k5_12_12, clique_60_k6_10_10, clique_60_k7_8_9, clique_60_k8_7_8, clique_60_k9_6_7, clique_70_k3_23_24	$\mathcal{S}_2$
diw_34	$\mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{D}_4 \times \mathcal{S}_4 \times \mathcal{S}_4$
diw_37	$\mathcal{S}_2 \times \mathcal{S}_4 \times \mathcal{S}_3 \times \mathcal{S}_4$
diw_38	$\mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_2 \times \mathcal{S}_3$
diw_43	$\mathcal{S}_3$
diw_44	$\mathcal{S}_3$

$\mathcal{S}_k$ = full symmetric group on k elements; $\mathcal{D}_k$ = dihedral group.

Symmetry: Computational Results

TECHNISCHE UNIVERSITÄT DARMSTADT

Results from [Hojny and P. 2023]:

	all (184)			all optimal (168)		only symmetric (21)
	time (s)	symtime (s)	# gens	time (s)	#nodes	time (s)
without	130.6	–	–	95.0	778.3	45.07
with	125.3	0.44	99	90.8	760.6	29.84

- Speed-up of about 4 % for all instances;
- Speed-up of about 34 % for the 21 instances that contain symmetry.
- Number of generators is quite small.
- Note that we do not exploit symmetries in the solutions of the SDPs (yet).

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Conclusion & Outlook

TECHNISCHE UNIVERSITÄT DARMSTADT

- Framework for solving general MISDPs
- Several methods help to improve performance.
- Solving SDPs is still one bottleneck, but often yields strong bounds.
- Future: follow development path for MIP-solvers for MISDP-solvers as well.

SCIP-SDP is available in source code at

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

Thank you for your attention!

References

- T. Gally, M. E. Pfetsch, and S. Ulbrich.
  A framework for solving mixed-integer semidefinite programs.
  Optimization Methods and Software, 33(3):594–632, 2018.
- C. Hojny and M. E. Pfetsch.
  Handling symmetries in mixed-integer semidefinite programs.
  In A. A. Cire, editor, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 69–78, Cham, 2023. Springer.
- F. Matter and M. E. Pfetsch.
  Presolving for mixed-integer semidefinite optimization.
  INFORMS J. Opt., 5(2):131–154, 2022.
- M. E. Pfetsch.
  Dual conflict analysis for mixed-integer semidefinite programs.
  Technical report, Optimization Online, 2023.