TECHNISCHE UNIVERSITÄT DARMSTADT

Solving Mixed-Integer SDPs

Marc Pfetsch, TU Darmstadt

based on work together with
Tristan Gally and Stefan Ulbrich

Main source: Dissertation of Tristan Gally, 2019

Discrete Optimization

Mixed-Integer Semidefinite Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

Mixed-integer semidefinite program (MISDP)

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

where A_i, $C \in \mathbb{R}^{n \times n}$ are symmetric, $b \in \mathbb{R}^m$, $\mathcal{I} \subseteq \{1, \dots, m\}$.

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

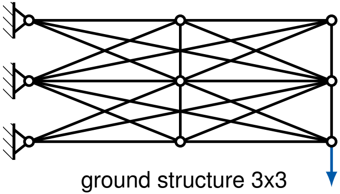
Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

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 $d_f = d \cdot n_f$

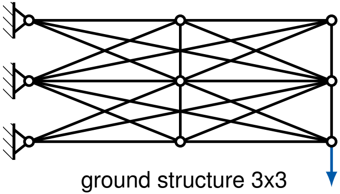


ground structure 3x3

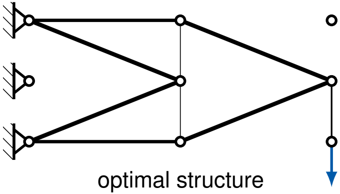
Robust Truss Topology Design

- 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 $d_f = d \cdot n_f$

- 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

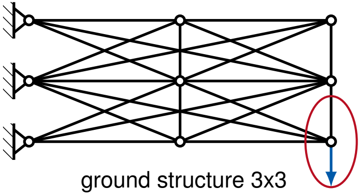


optimal structure

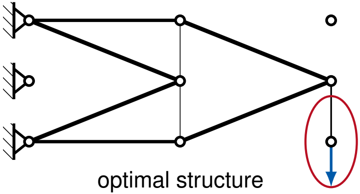
Robust Truss Topology Design

- 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

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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.

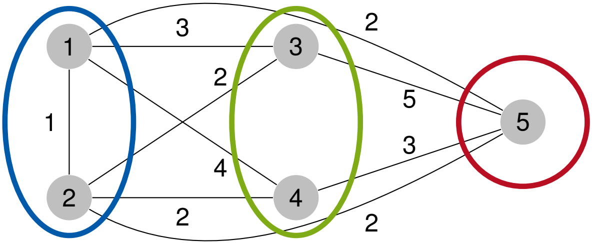
Cardinality Constrained Least Squares [Pilanci/Wainwright/El Ghaoui 2015]

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

Minimum k-Partitioning

- 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 $V_1, \dots, V_k$ 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.

TECHNISCHE UNIVERSITÄT DARMSTADT

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

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- 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

TECHNISCHE UNIVERSITÄT DARMSTADT

Outer Approximation / Cutting Planes

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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

TECHNISCHE UNIVERSITÄT DARMSTADT

- For convex MINLP one usually uses gradient cuts

$$g_j(\overline{x}) + \nabla g_j(\overline{x})^\top (x - \overline{x}) \leq 0.$$

- But function of smallest eigenvalue is not differentiable everywhere.
- Instead use characterization $X \succeq 0 \quad \Leftrightarrow \quad u^\top X\, u \geq 0$ 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 valid and cuts off y^*.

Cutting Planes: MISOCP vs. MISDP

- Cutting planes often used by solvers for mixed-integer second-order cone problems.
- 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

TECHNISCHE UNIVERSITÄT DARMSTADT

- Relax integrality instead of SDP-constraint.
- Branch on 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.
- Need to satisfy convergence assumptions of SDP-solvers.

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Strong Duality in SDP

TECHNISCHE UNIVERSITÄT DARMSTADT

Dual SDP (D)

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

Primal SDP (P)

$$\begin{aligned} \inf \quad & C \bullet X \\ \text{s.t.} \quad & A_i \bullet X = b_i \quad \forall\, i \leq m, \\ & X \succeq 0. \end{aligned}$$

where $A \bullet B = \mathrm{Tr}(AB) = \sum_{ij} A_{ij} B_{ij}$.

Strong Duality in SDP

TECHNISCHE UNIVERSITÄT DARMSTADT

Dual SDP (D)

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

Primal SDP (P)

$$
\begin{aligned}
\inf \quad & C \bullet X \\
\text{s.t.} \quad & A_i \bullet X = b_i \qquad \forall\, i \le m, \\
& X \succeq 0.
\end{aligned}
$$

where $A \bullet B = \mathrm{Tr}(AB) = \sum_{ij} A_{ij} B_{ij}$.

- Strong Duality holds if Slater condition holds for (P) or (D): $\exists\, X \succ 0$ feasible 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, this is assumed by most interior-point SDP-solvers.
- Can these assumptions be lost through branching?

Strong Duality in Branch-and-Bound

TECHNISCHE UNIVERSITÄT DARMSTADT

Theorem [Gally, P., 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,
- 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].
- Analogous result for adding linear constraints to (P) with set of optimal X compact and nonempty and (P_+) feasible.

Slater Condition in Branch-and-Bound

TECHNISCHE UNIVERSITÄT DARMSTADT

Proposition [Gally, P., 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 $\mathcal{A} : S_n \to \mathbb{R}^m$, $X \mapsto (A_i \bullet X)_{i \in [m]}$, 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-condition in Branch-and-Bound

TECHNISCHE UNIVERSITÄT DARMSTADT

KKT-points may get lost after branching, for example:

(D)

$$\begin{aligned} \sup \quad & 2\,y_1 - y_2 \\ \text{s.t.} \quad & \begin{pmatrix} 0.5 & -y_1 \\ -y_1 & y_2 \end{pmatrix} \succeq 0. \end{aligned}$$

(P)

$$\begin{aligned} \inf \quad & 0.5\,X_{11} \\ \text{s.t.} \quad & \begin{pmatrix} X_{11} & 1 \\ 1 & 1 \end{pmatrix} \succeq 0. \end{aligned}$$

- 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

TECHNISCHE UNIVERSITÄT DARMSTADT

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

(D_+)

$$\begin{array}{ll} \sup & 2\,y_1 - y_2 \\ \text{s.t.} & \begin{pmatrix} 0.5 & -y_1 & 0 \\ -y_1 & y_2 & 0 \\ 0 & 0 & -y_2 \end{pmatrix} \succeq 0, \end{array}$$

(P_+)

$$\begin{array}{ll} \inf & 0.5\,X_{11} \\ \text{s.t.} & \begin{pmatrix} X_{11} & 1 & X_{13} \\ 1 & X_{22} & X_{23} \\ X_{13} & X_{23} & X_{22} - 1 \end{pmatrix} \succeq 0. \end{array}$$

- 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

	Dual Slater				Primal Slater		
application	✓	✗	infeas	?	✓	✗	?
TTD	83.22 %	5.82 %	10.96 %	0.00 %	99.99 %	0.00 %	0.01 %
CLS	56.26 %	3.44 %	40.30 %	0.00 %	100.00 %	0.00 %	0.00 %
M*k*P	3.66 %	62.93 %	33.41 %	0.00 %	100.00 %	0.00 %	0.00 %
overall	45.89 %	25.33 %	28.78 %	0.00 %	100.00 %	0.00 %	0.00 %

run on cluster of 64-bit Intel Xeon E5-1620 CPUs running at 3.50 GHz with 32 GB RAM using SCIP-SDP 3.1.1, SCIP 6.0.0, and MOSEK 8.1.0.54 on test set of 194 CBLIB instances

Checking Infeasibility

TECHNISCHE UNIVERSITÄT DARMSTADT

If interior-point solver did not converge for original formulation, solve

Feasibility Check [Mars 2013]

$$\begin{aligned} \inf \quad & r \\ \text{s.t.} \quad & C - \sum_{i=1}^{m} A_i y_i + I\, r \succeq 0. \end{aligned}$$

If optimum $r^* > 0$, original problem is infeasible and node can be cut off.

Handling Failure of the Dual Slater Condition

TECHNISCHE UNIVERSITÄT DARMSTADT

If problem is not infeasible, solve

Penalty Formulation [Benson/Ye 2008]

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

for sufficiently large Γ to compute an upper bound.

- If optimal $r^* = 0$, then solution is also optimal for original problem.
- Adds constraint $\mathrm{Tr}(X) \leq \Gamma$ to primal problem, for large enough Γ also preserves primal Slater condition.

SDP-Solvers depending on Slater Condition

TECHNISCHE UNIVERSITÄT DARMSTADT

Behavior if Slater condition holds for (P) and (D)

solver	default	penalty	bound	unsucc
SDPA	90.78 %	5.50 %	0.00 %	3.73 %
DSDP	99.68 %	0.32 %	0.00 %	0.00 %
MOSEK	99.51 %	0.49 %	0.00 %	0.00 %

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

solver	default	penalty	bound	unsucc
SDPA	56.15 %	1.14 %	13.00 %	29.71 %
DSDP	99.81 %	0.13 %	0.00 %	0.05 %
MOSEK	99.20 %	0.79 %	0.01 %	0.00 %

Behavior if problem is infeasible

solver	default	penalty	bound	unsucc
SDPA	46.99 %	39.46 %	4.88 %	8.67 %
DSDP	92.44 %	2.23 %	1.39 %	3.94 %
MOSEK	88.42 %	10.36 %	1.22 %	0.00 %

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

SCIP-SDP

- Based on SCIP framework.
- Supports both nonlinear B&B and LP-based branch-and-cut.
- Two file-readers
  - CBF
  - SDPA with added integrality information
- Constraint handler for SDP-constraints
- Interfaces to three SDP solvers
  - DSDP
  - SDPA
  - MOSEK
- Two additional heuristics
  - SDP-based diving, SDP-based randomized rounding
- Two additional propagators
  - SDP-based OBBT, SDP-based dual fixing
- Parallelized version available as UG-MISDP.
- Supports rank 1 constraints (implemented together with Frederic Matter).

Constraint Handler

- Handles SDP-constraints in dual form

$$C - \sum_{i=1}^{m} A_i y_i \succeq 0.$$

- For branch & cut separate eigenvector cuts.
- Adds linear constraints implied by SDP-constraint during presolving (e.g., non-negativity of diagonal entries).
  - Redundant for nonlinear branch-and-bound, but can be used by SCIP during presolving for fixing variables.
  - Still lead to speedup of 6% even for nonlinear branch-and-bound.

Relaxator and SDPI

- 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,
  - 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.

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,
- L: Lower bound on the optimal objective value of the MISDP.

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

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

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

Overview

TECHNISCHE
UNIVERSITÄT
DARMSTADT

Warmstarts

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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 does not 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
  - Proposed by [Gondzio 1998] for MIP.
  - Store previous iterate, further away from optimum but still sufficiently interior.
- Convex combination with strictly feasible solution
  - Due to [Helmberg and Rendl 1998]
  - Convex combination with scaled identity matrix or analytic center of root node.
- Projection onto positive definite matrices
  - Project onto set of matrices with $\lambda_{\text{min}} \geq \underline{\lambda} > 0$.
  - Can be computed explicitly from eigendecomposition.
- Rounding problems
  - Proposed by [Çay, Pólik and Terlaky 2017] for MISOCP.
  - Compute feasible solutions for adjusted problems by fixing eigenvectors of parent node and optimizing over eigenvalues as LP.
  - Can prove optimality/suboptimality/infeasibility by linear programming only.
  - Solution still needs to be adjusted for strict feasibility.

Comparison of Warmstarting Techniques

TECHNISCHE UNIVERSITÄT DARMSTADT

settings	solved	time	sdpiter
`no warmstart`	288	117.47	18,957.04
`simple warmstart`	127	794.32	–
`preoptgap 0.01`	191	349.88	–
`preoptgap 0.5`	243	232.49	22,830.56
`0.01 id pdsame`	288	110.63	16,172.51
`0.5 id pddiff`	288	105.36	15,125.79
`0.5 id pdsame`	290	105.56	16,362.67
`0.5 anacent`	286	140.88	20,463.24
`proj minev 0.1`	285	112.74	16,277.19
`roundingprob 0.5 id`	282	174.83	13,952.38
`roundingprob inf only`	287	155.19	15,282.34

run on cluster of 64-bit Intel Xeon E5-1620 CPUs with 3.50 GHz and 32 GB RAM using SCIP-SDP 3.1.1, SCIP 6.0.0, and SDPA 7.4.0 on test set of 194 CBLIB instances and 126 compressed sensing instances; times (and iterations) as shifted geometric means (over instances solved by all settings except unadjusted warmstart and preoptimal)

Comparison of Warmstarting Techniques

TECHNISCHE UNIVERSITÄT DARMSTADT

Speedup for conv 0.01 pdsame

application	solved	time	sdpiter
TTD	−1	+22.4	+10.6
CLS	0	−9.1	−13.0
M*k*P	+1	−16.0	−21.0
RIP	0	−9.7	−18.1

Speedup for conv 0.5 pddiff

application	solved	time	sdpiter
TTD	-1	+15.7	−10.4
CLS	0	−5.2	−4.0
M*k*P	+1	+0.1	−9.8
RIP	0	−27.9	−31.2

Speedup for conv 0.5 pdsame

application	solved	time	sdpiter
TTD	+1	−8.7	−26.3
CLS	−1	−8.7	−11.1
M*k*P	+2	−8.6	+0.6
RIP	0	−12.6	−17.4

Speedup for projection

application	solved	time	sdpiter
TTD	−3	+10.4	−21.7
CLS	−1	+0.7	−5.5
M*k*P	+1	+5.7	+12.1
RIP	0	−17.3	−24.1

run on cluster of 64-bit Intel Xeon E5-1620 CPUs running at 3.50 GHz with 32 GB RAM using SCIP-SDP 3.1.1, SCIP 6.0.0, and SDPA 7.4.0 on test set of 194 CBLIB instances and 126 compressed sensing instances; times (and iterations) as shifted geometric means (over instances solved by all settings except unadjusted warmstart and preoptimal)

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

MISDP Solvers

TECHNISCHE UNIVERSITÄT DARMSTADT

Nonlinear branch-and-bound

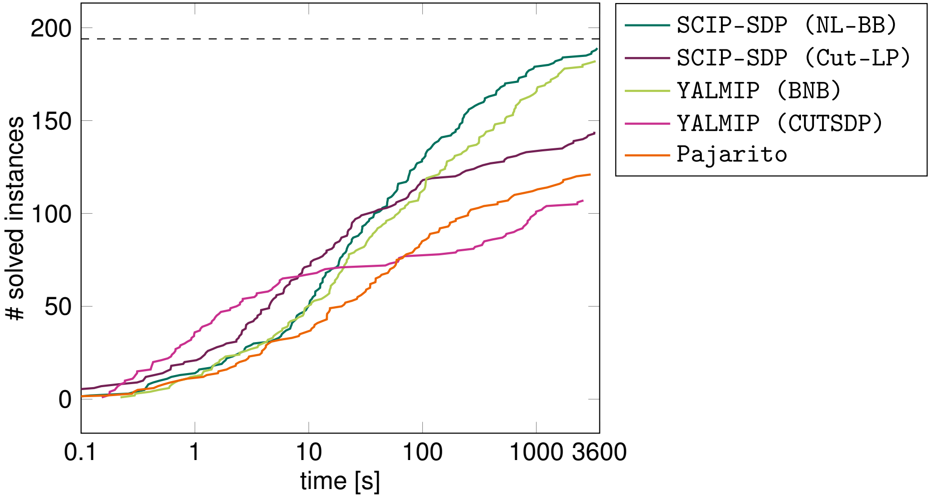
- SCIP-SDP 3.1.1 (nonlinear branch-and-bound)
  Our implementation, using SCIP as B&B-framework
- YALMIP-BNB R20180926
  MATLAB toolbox for rapid prototyping

Cutting plane / outer approximation approaches

- SCIP-SDP 3.1.1 (LP-based cutting planes)
- YALMIP-CUTSDP R20180926
- Pajarito 0.5.0
  - Julia implementation for mixed-integer convex including MISDP
  - MIP-solver-drives version (single MIP with SDP solves for stronger cuts)

Comparison of MISDP Solvers I

TECHNISCHE UNIVERSITÄT DARMSTADT



Comparison of MISDP Solvers II

solver	TTD		CLS		Mk-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

Overview

TECHNISCHE UNIVERSITÄT DARMSTADT

Parallelization

TECHNISCHE UNIVERSITÄT DARMSTADT

- ▷ Based on UG framework.
- ▷ Can either parallelize on SDP or on MIP side.
- ▷ Parallel Cholesky for SDPs depends on sparsity pattern and usually only efficient for larger SDPs.
- ▷ Solving subtrees of branch and bound tree in parallel not possible for root node.

$\Rightarrow$ Use racing ramp-up in root node to decide between different settings, in particular LP vs. SDP.

- ▷ Start a number of SCIP-SDP instances in parallel with half of them using LP-based and the other half SDP-based settings.
- ▷ After enough nodes have been generated, decide on "best" solver and distribute this solver's tree.

Numerical Results for Parallelization

TECHNISCHE
UNIVERSITÄT
DARMSTADT

solver / # threads	TTD opt	TTD time	CLS opt	CLS time	Mk-P opt	Mk-P time	Total opt	Total time
`SCIP-SDP`	55	84.01	62	142.19	67	54.44	184	86.59
`UG-MISDP 1`	54	107.49	62	156.70	58	107.81	174	122.23
`UG-MISDP 2`	56	64.93	64	23.31	56	92.25	176	53.79
`UG-MISDP 4`	58	39.76	65	18.48	60	85.61	183	42.07
`UG-MISDP 8`	58	32.07	65	14.51	60	72.35	183	34.57
`UG-MISDP 16`	59	21.03	65	16.37	59	78.46	183	32.65
`UG-MISDP 32`	59	21.27	65	18.38	56	92.14	180	36.11

run on Intel Xeon E5-4650 CPUs running at 2.70 GHz with 512 GB of shared RAM; time limit of 3600 seconds, times as shifted geometric means; using developer versions of SCIP 6.0.0, SCIP-SDP 3.1.1, UG 0.8.6, SDPs solved using MOSEK 8.1.0.54, LPs using CPLEX 12.6.3; instances from CBLIB

Overview

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.
  Computational Mixed-Integer Semidefinite Programming.
  PhD thesis, TU Darmstadt, 2019.
- 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.
- S. Mars.
  Mixed-Integer Semidefinite Programming with an Application to Truss Topology Design.
  PhD thesis, FAU Erlangen-Nürnberg, 2013.
- Y. Shinano, D. Rehfeldt, and T. Gally.
  An easy way to build parallel state-of-the-art combinatorial optimization problem solvers: A computational study on solving steiner tree problems and mixed integer semidefinite programs by using ug[scip-*,*]-libraries.
  In *Proceedings of the 9th IEEE Workshop Parallel / Distributed Combinatorics and Optimization*, pages 530 – 541, 2019.