Parallelizing SCIP-SDP via the UG framework

Tristan Gally

joint work with Marc E. Pfetsch, Chuen-Teck See, and Yuji Shinano

SFB 805

Control of Uncertainty in Load-Carrying Structures in Mechanical Engineering

Mixed-Integer Semidefinite Programming

TECHNISCHE UNIVERSITÄT DARMSTADT

- Mixed-integer semidefinite program

MISDP

$$
\begin{aligned}
\inf \quad & b^T y \\
\text{s.t.} \quad & \sum_{i=1}^{m} A_i y_i - A_0 \succeq 0, \\
& y_i \in \mathbb{Z} \quad \forall\, i \in \mathcal{I}
\end{aligned}
$$

for symmetric matrices $(A_i)_{i \leq m}$.

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

Discrete Optimization

Contents

Contents

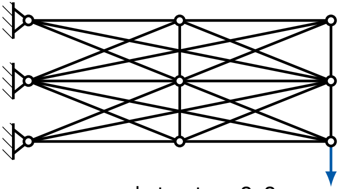
TECHNISCHE UNIVERSITÄT DARMSTADT

Discrete Optimization

Truss Topology Design

TECHNISCHE UNIVERSITÄT DARMSTADT

- 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 $d_f = d \cdot n_f$

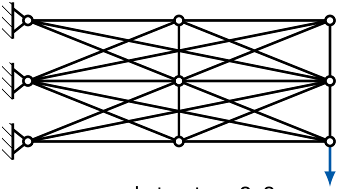


ground structure 3x3

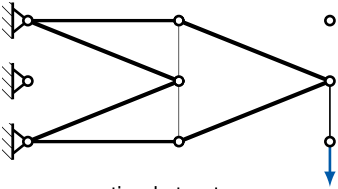
Truss Topology Design

- n nodes $V = \left\{ v_i \in \mathbb{R}^d : i = 1, \dots, n \right\}$
- n_f free nodes $V_f \subset V$
- m possible bars
  $E \subseteq \left\{ \{v_i, v_j\} : i \neq j \right\}, |E| = m$
- 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

TECHNISCHE UNIVERSITÄT DARMSTADT

TTD-SDP [Ben-Tal, Nemirovski 1997]

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

- E : set of possible bars
- ℓ_e : length of bar e
- x : cross-sectional areas
- f : external force
- $c_{\max}$: upper bound on compliance
- A_e: bar stiffness matrices

with stiffness matrix $A(x) = \sum_{e \in E} A_e \ell_e x_e$.

Truss Topology Design

TECHNISCHE UNIVERSITÄT DARMSTADT

- In practice, we won't be able to produce/buy bars of any desired size.

$\Rightarrow$ Only allow cross-sectional areas from a discrete set $\mathcal{A}$.

Truss Topology Design

TECHNISCHE UNIVERSITÄT DARMSTADT

- In practice, we won't be able to produce/buy bars of any desired size.

$\Rightarrow$ Only allow cross-sectional areas from a discrete set $\mathcal{A}$.

TTD-MISDP [Kočvara 2010, Mars 2013]

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

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

Compressed Sensing

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

ℓ_0-Minimization

$$\begin{aligned} \min \quad & \|x\|_0 \\ \text{s.t.} \quad & Ax = b \\ & x \in \mathbb{R}^n \end{aligned}$$

where $\|x\|_0 := |supp(x)|$.

Compressed Sensing

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

ℓ_0-Minimization

$$\begin{aligned} \min \quad & \|x\|_0 \\ \text{s.t.} \quad & Ax = b \\ & x \in \mathbb{R}^n \end{aligned}$$

where $\|x\|_0 := |supp(x)|$.

- Under certain conditions on A, this is equivalent to

ℓ_1-Minimization

$$\begin{aligned} \min \quad & \|x\|_1 \\ \text{s.t.} \quad & Ax = b \\ & x \in \mathbb{R}^n \end{aligned}$$

Compressed Sensing

TECHNISCHE UNIVERSITÄT DARMSTADT

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

$$\alpha_k^2 \|x\|_2^2 \leq \|Ax\|_2^2 \leq \beta_k^2 \|x\|_2^2 \qquad \forall x : \|x\|_0 \leq k$$

Discrete Optimization

Compressed Sensing

TECHNISCHE UNIVERSITÄT DARMSTADT

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

$$\alpha_k^2 \|x\|_2^2 \leq \|Ax\|_2^2 \leq \beta_k^2 \|x\|_2^2 \qquad \forall x : \|x\|_0 \leq k$$

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 ℓ_0- and the ℓ_1-optimization problem.

Compressed Sensing

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

$$\alpha_k^2 \|x\|_2^2 \leq \|Ax\|_2^2 \leq \beta_k^2 \|x\|_2^2 \qquad \forall x : \|x\|_0 \leq k$$

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 ℓ_0- and the ℓ_1-optimization problem.

But $\mathcal{NP}$-hard in general to compute optimal constants …

Compressed Sensing

TECHNISCHE UNIVERSITÄT DARMSTADT

The optimal constant α_k^2 (and similarly β_k^2) for

$$\alpha_k^2 \|x\|_2^2 \leq \|Ax\|_2^2 \leq \beta_k^2 \|x\|_2^2 \qquad \forall x : \|x\|_0 \leq k$$

can be computed via the following non-convex cardinality-constrained QP:

RIP-QP

$$\begin{aligned} \inf \quad & \|Ax\|_2^2 \\ \text{s.t.} \quad & \|x\|_2^2 = 1, \\ & \|x\|_0 \leq k. \end{aligned}$$

Compressed Sensing

Idea: Substitute $X := xx^\top$ to obtain the non-convex rank-constrained MISDP

RIP-Rk1-MISDP

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

Compressed Sensing

TECHNISCHE UNIVERSITÄT DARMSTADT

RIP-MISDP

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

Theorem [G., Pfetsch 2016]

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

Further Applications

TECHNISCHE UNIVERSITÄT DARMSTADT

- Combinatorial optimization problems strengthened by semidefinite relaxations
  - Max-cut / minimum *k*-partitioning
  - Stable set
  - Quadratic assignment problems (including TSP as special case)
  - ...
- Nonlinear / semidefinite problems with binary decisions
  - Cardinality-constrained least squares
  - Transmission switching problems for AC power flow
  - Design and control of linear time-invariant systems
  - ...

Contents

TECHNISCHE UNIVERSITÄT DARMSTADT

LP-Based Cutting Plane Approach

TECHNISCHE UNIVERSITÄT DARMSTADT

- Solve LP in each node and enforce SDP constraint via linear cuts.
- Cannot use gradient cuts since SDP constraint is non-smooth in general.
- Use characterization $X \succeq 0 \quad \Leftrightarrow \quad u^\top X\, u \geq 0$ for all $u \in \mathbb{R}^n$.
- For eigenvector v to smallest eigenvalue of $Z^\star := \sum_{i=1}^{m} A_i\, y_i^\star - A_0 \not\succeq 0$, the linear cut

$$v^\top \left(\sum_{i=1}^{m} A_i\, y_i - A_0 \right) v \geq 0$$

is valid and cuts off y^*.

SDP-Based Branch and Bound

- Solve an SDP in each node of the branch-and-bound tree.
- Needs additional techniques to ensure constraint qualifications for validity of interior-point SDP solvers.
- Less efficient warmstarts available for interior-point solvers.
- No simplex tableau available for cutting plane generation.

Comparison of the two Approaches

TECHNISCHE UNIVERSITÄT DARMSTADT

- Cutting-plane-based approaches used by most commercial solvers for mixed-integer second-order cone (MISOCP).
- 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 relaxations can still be solved up to given precision in polynomial time.

Contents

TECHNISCHE UNIVERSITÄT DARMSTADT

Discrete Optimization

SCIP-SDP

TECHNISCHE UNIVERSITÄT DARMSTADT

- Plugin for SCIP to extend it to mixed-integer semidefinite programming.
- Supports both SDP-based branch and bound and LP-based cutting plane approach, can switch between both via parameter settings.

SCIP-SDP

- Plugin for SCIP to extend it to mixed-integer semidefinite programming.
- Supports both SDP-based branch and bound and LP-based cutting plane approach, can switch between both via parameter settings.
- Extends SCIP by an SDP constraint handler, SDP relaxation handler and several heuristics, propagators, file readers, ...
- Uses penalty approach in case constraint qualification fails.
- Includes interfaces to three different SDP solvers: MOSEK, SDPA and DSDP.
- Currently one of the fastest publicly available MISDP solvers.

Numerical Comparison of Solution Approaches

settings	TTD		CLS		M*k*-P		Total	
	solved	time	solved	time	solved	time	solved	time
NL-BB	55	84.01	62	142.19	67	54.44	184	86.59
LP-based	44	169.23	65	11.53	30	712.28	139	134.65
best	55	60.59	65	11.53	67	49.73	187	34.69

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, SDPs solved using MOSEK 8.1.0.54, LPs using CPLEX 12.6.3; instances from CBLIB

Contents

Parallelizing an MISDP solver

TECHNISCHE UNIVERSITÄT DARMSTADT

- Parallelization can happen either on the SDP or the MIP side.
- On the SDP side: Parallel computation of Schur complement matrix and parallel Cholesky factorization.
  - Parallel Cholesky may depend on sparsity of the constraint matrices.
  - Due to overhead only efficient for large SDPs, not necessarily the smaller ones appearing in MISDP.

Parallelizing an MISDP solver

TECHNISCHE UNIVERSITÄT DARMSTADT

- Parallelization can happen either on the SDP or the MIP side.
- On the SDP side: Parallel computation of Schur complement matrix and parallel Cholesky factorization.
  - Parallel Cholesky may depend on sparsity of the constraint matrices.
  - Due to overhead only efficient for large SDPs, not necessarily the smaller ones appearing in MISDP.
- On the MIP side: Parallel solution of branch-and-bound subtrees.
  - Independent of solution approach for SDP relaxations.
  - Cannot solve root node in parallel.

Racing Ramp-Up

TECHNISCHE
UNIVERSITÄT
DARMSTADT

- Use racing ramp-up to decide between LP and SDP approach.
- Start a number of SCIP-SDP instances in parallel with half of them using LP-based and the other half SDP-based settings.
- Additionally change further parameters that are relevant for LP/SDP approach.
- Total of 32 different parameter settings for up to 32 threads.
- After enough nodes have been generated, decide on "best" solver and distribute this solver's tree.

Parallelization via UG Framework

TECHNISCHE UNIVERSITÄT DARMSTADT

- Racing ramp-up uses resources in root node and decides on best settings.
- In the remainder of the solution process have different threads solve their subtrees in parallel.
- Implementation allows for both shared and distributed memory.
- Less than 100 lines of additional code.
  - Mostly including plugins and changing default parameters

UG-MISDP Output

TECHNISCHE UNIVERSITÄT DARMSTADT

```
                              Nodes   Active
    Time          Nodes        Left  Solvers     Best Integer        Best Node       Gap     Best Node(S)     Gap(S)
*      0              0           1        8          83.1357                -         -
*      0              0           1        8          79.9734                -         -
*      0              0           1        8          69.7193                -         -
*      1              2           1        8          61.9806          30.3961   103.91%
*      1              2           1        8          60.5663          30.3961    99.26%
*      2             28          27        8          34.8569          30.6254    13.82%
*      2             40          39        8          33.2132          30.6518     8.36%
Racing ramp-up finished after 3.45 seconds. Selected strategy 1.
*      5            223         102        8          33.2132          30.7719     7.93%          30.7719      7.93%
*      5            298         165        8          32.6208          30.7888     5.95%          30.7888      5.95%
       5            298         165        8          32.6208          30.7888     5.95%          30.7888      5.95%
      10           2485         304        8          32.6208          30.8145     5.86%          30.8145      5.86%
*     14           3582         534        8          32.2066          30.8971     4.24%          31.2763      2.97%
      15           3961         406        8          32.2066          30.9182     4.17%          30.9771      3.97%
      20           5851         493        8          32.2066          30.9840     3.95%          30.9840      3.95%
      25           7728         464        8          32.2066          31.1334     3.45%          31.1916      3.25%
      30           9798         190        8          32.2066          31.2641     3.01%          31.2641      3.01%
      36          11080          26        8          32.2066          31.3607     2.70%          31.3607      2.70%
      39          11852           0        0          32.2066                -         -                -
SCIP Status        : problem is solved
Total Time         : 38.6700
  solving          : 38.6700
  presolving       : 0.0011 (included in solving)
B&B Tree           :
  nodes (total)    : 11852
Solution           :
  Solutions found  : 10
  Primal Bound     : +3.22066307259600e+01
  Dual Bound       : +3.22066307259600e+01
Gap                : 0.00000 %
```


Discrete Optimization

UG-MISDP Solver Statistics

TECHNISCHE UNIVERSITÄT DARMSTADT

```
######### Solver Rank = 1 is terminated. #########
#=== Elapsed time to terminate this Solver = 38.67
#=== Total computing time = 37.16
#=== Total idle time = 1.51
#=== ( Idle time to start first ParaNode = 0.12, Idle time between ParaNods = 0.04, Idle Time after last ParaNode = 1.34 )
#=== ( Idle time to wait notification Id messages = 0.01 )
#=== ( Idle time to wait acknowledgment of completion = 0 )
#=== ( Idle time to wait token = 0 )
#=== Total root node process time = 5.18 ( Mean = 0.345333, Min = 0.13, Max = 0.64 )
#=== The number of ParaNodes received in this solver = 15
#=== The number of ParaNodes sent from this solver = 85
#=== The number of nodes solved in this solver = 1589 ( / Subtree : Mean = 105, Min = 20, Max = 250 )
#=== Total number of restarts in this solver = 29( / Subtree : Mean = 1, Min = 1, Max = 2 )
#=== Total number of cuts sent from this solver = 0( / Subtree : Mean = 0, Min = 0, Max = 0 )
#=== The number of ParaNodes solved in this solver = 15
#=== ( Solved at root node  =  0, Solved at pre-checking of root node solvability = 0 )
#=== The number of improved solutions found in this solver = 2
#=== The number of tightened variable bounds in this solver = 14 ( Int: 14 )
```

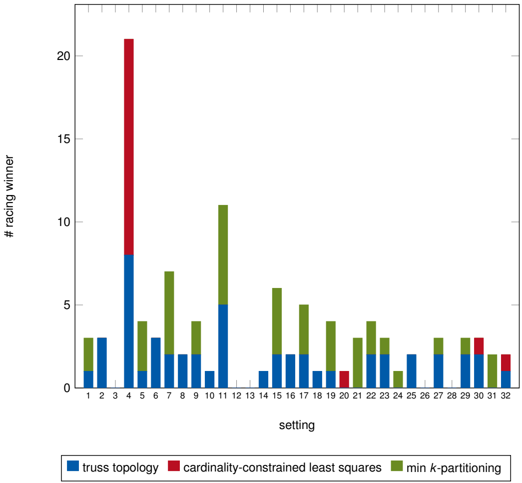

Discrete Optimization

Contents

TECHNISCHE UNIVERSITÄT DARMSTADT

Discrete Optimization

Racing Statistics



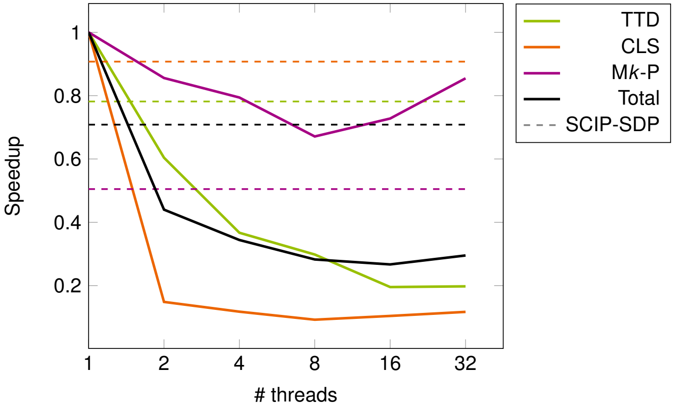
- 4: LP + easycip emphasis
- 5: SDP + objcoef branching
- 7: SDP + most infeasible branching
- 9: SDP + disabled subdeterminant cuts
- 11: SDP + more aggressive rounding
- 15: SDP + objective limit
- 17: SDP + OBBT
- 19: SDP + never increase penaltyparam
- 22: LP + more flow cover cuts

Runtimes

solver	TTD		CLS		M*k*-P		Total	
	solved	time	solved	time	solved	time	solved	time
SCIP-SDP	55	84.01	62	142.19	67	54.44	184	86.59
UG-MISDP 1 thr.	54	107.49	62	156.70	58	107.81	174	122.23
UG-MISDP 2 thr.	56	64.93	64	23.31	56	92.25	176	53.79
UG-MISDP 4 thr.	58	39.76	65	18.48	60	85.61	183	42.07
UG-MISDP 8 thr.	58	32.07	65	14.51	60	72.35	183	34.57
UG-MISDP 16 thr.	59	21.03	65	16.37	59	78.46	183	32.65
UG-MISDP 32 thr.	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

Speedups



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

Contents

Conclusion

TECHNISCHE UNIVERSITÄT DARMSTADT

- Both SDP-based branch and bound and LP-based cutting plane approaches can be fast on different applications.
- Use racing ramp-up to dynamically choose between both approaches.
- Implemented using the UG framework in just over 100 lines of code.
- Large speedup for two threads due to switching to LPs, further speedups due to parallel branch and bound for problems with large enough trees.

Obtaining UG-MISDP

SCIP-SDP is available from
`http://www.opt.tu-darmstadt.de/scipsdp/`

UG-MISDP is available as part of the UG distribution
and the SCIP optimization suite from
`https://scip.zib.de/`

Thank you for your attention!

References

A. Ben-Tal and A. Nemirovski.
Robust truss topology design via semidefinite programming.
SIAM Journal on Optimization, 7(4):991–1016, 1997.

A. Ben-Tal and A. Nemirovski.
On polyhedral approximations of the second-order cone.
Mathematics of Operations Research, 26:193–205, 2001.

G. Braun, S. Fiorini, S. Pokutta, and D. Steurer.
Approximation limits of linear programs (beyond hierarchies).
Mathematics of Operations Research, 40(3):756–772, 2015.

A. Eisenblätter.
The semidefinite relaxation of the k-partition polytope is strong.
In W. J. Cook and A. S. Schulz, editors, *Proceedings of the 9th International IPCO Conference on Integer Programming and Combinatorial Optimization*, volume 2337 of *Lecture Notes in Computer Science*, pages 273–290. Springer, Berlin Heidelberg, 2002.

H. A. Friberg.
CBLIB 2014: a benchmark library for conic mixed-integer and continuous optimization.
Mathematical Programming Computation, 8(2):191–214, 2016.

T. Gally and M. E. Pfetsch.
Computing restricted isometry constants via mixed-integer semidefinite programming.
Technical report, Optimization Online, 2016.

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.

M. Kočvara.
Truss topology design with integer variables made easy.
Technical report, School of Mathematics, University of Birmingham, 2010.

S. Mars.
Mixed-Integer Semidefinite Programming with an Application to Truss Topology Design.
PhD thesis, FAU Erlangen-Nürnberg, 2013.

M. Pilanci, M. J. Wainwright, and L. El Ghaoui.
Sparse learning via Boolean relaxations.
Mathematical Programming Series B, 151(1):62–87, 2015.

A. M. Tillmann and M. E. Pfetsch.
The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing.
IEEE Transactions on Information Theory, 60(2):1248–1259, 2014.