Undercover

A primal heuristic for MINLP based on sub-MIPs generated by set covering

Timo Berthold

Zuse Institute Berlin • MATHEON • BMS

joint work with Ambros M. Gleixner

DFG Research Center MATHEON
Mathematics for key technologies

1. Introduction: MINLP & primal heuristics
2. A generic algorithm for Undercover
3. Finding minimum covers
4. Computational environment and experiments
5. Extensions: fix-and-propagate etc.
6. Conclusion

Outline

Mixed Integer Linear Program

Objective function:

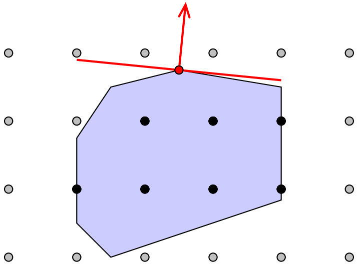
▷ linear function

Feasible set:

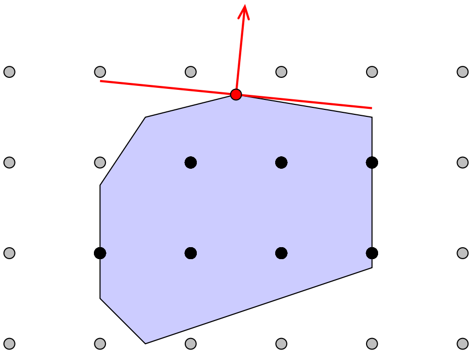
▷ described by linear constraints

Variable domains:

▷ real or integer values

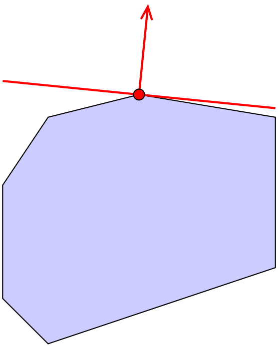


$$\begin{aligned} \min\ & c^T x && c \in \mathbb{R}^n \\ \text{s. t. } & Ax \leqslant b && A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^m \\ & x_i \in \mathbb{Z} && i \in \mathcal{I} \subseteq \{1, \ldots, n\} \end{aligned}$$



Important terms

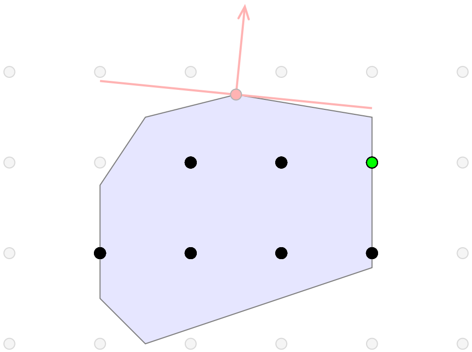
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions



Important terms

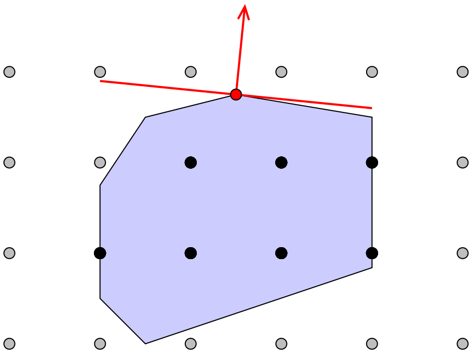
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions

What is Mixed-Integer Linear Programming?



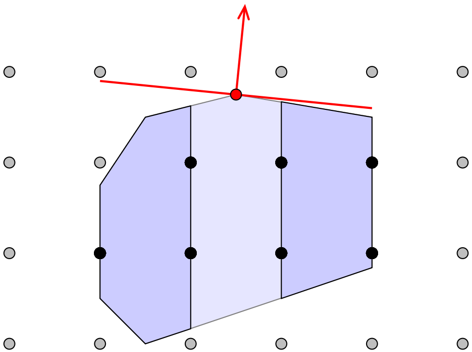
Important terms

- ▷ LP relaxation
- ▷ LP-based branch-and-bound
- ▷ feasible solutions
- ▷ optimal solutions



Important terms

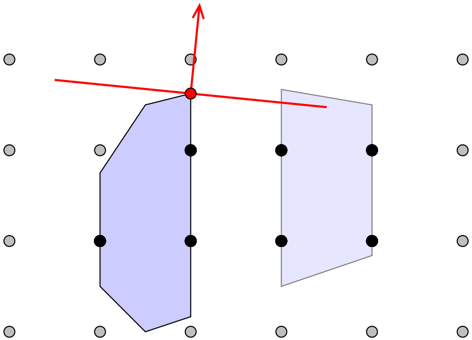
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions



Important terms

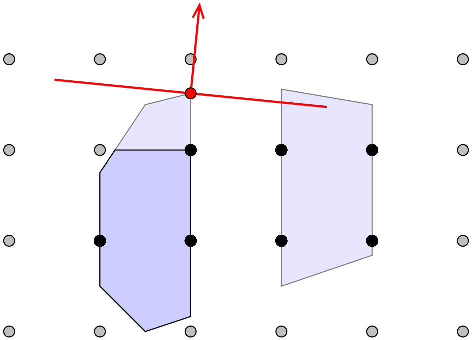
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions

What is Mixed-Integer Linear Programming?



Important terms

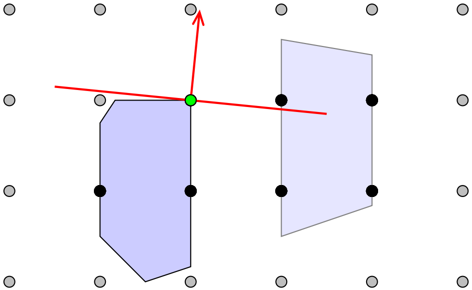
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions



Important terms

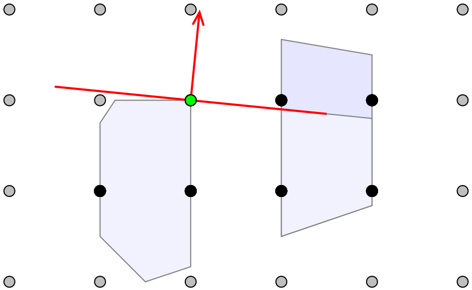
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions

What is Mixed-Integer Linear Programming?



Important terms

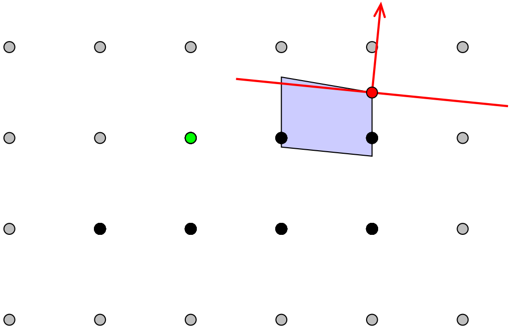
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions



Important terms

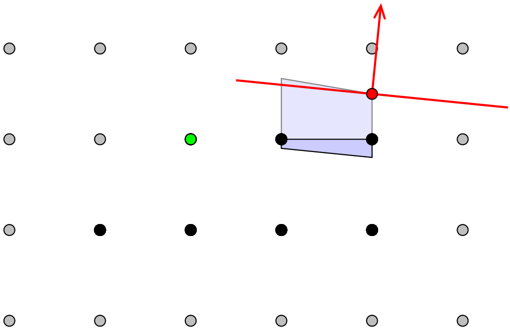
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions

What is Mixed-Integer Linear Programming?



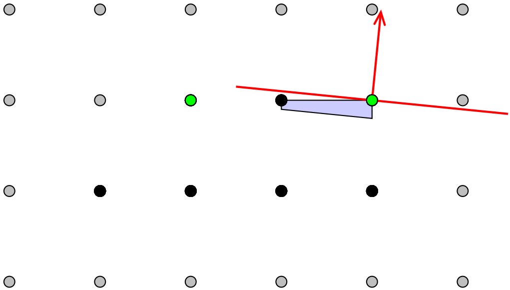
Important terms

- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions



Important terms

- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions



Important terms

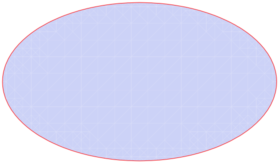
- LP relaxation
- LP-based branch-and-bound
- feasible solutions
- optimal solutions

What is Mixed-Integer Nonlinear Programming?

$$\begin{aligned} \min\ & c^T x & & c \in \mathbb{R}^n \\ \text{such that } & g(x) \leqslant 0, & & g \in C^1(\mathbb{R}^n, \mathbb{R}^m) \\ & x_i \in \mathbb{Z}, \quad i \in \mathcal{I}. & & \mathcal{I} \subseteq \{1, \dots, n\} \end{aligned}$$

MINLP is difficult due to combination of

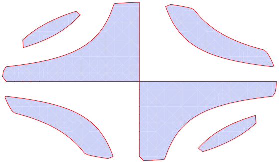
- nonlinearity $\quad x_1^2 + 3x_2^2 \leqslant 3$



$$\begin{aligned} \min\ & c^T x && c \in \mathbb{R}^n \\ \text{such that } & g(x) \leqslant 0, && g \in C^1(\mathbb{R}^n, \mathbb{R}^m) \\ & x_i \in \mathbb{Z}, \quad i \in \mathcal{I}. && \mathcal{I} \subseteq \{1, \dots, n\} \end{aligned}$$

MINLP is difficult due to combination of

- nonlinearity $x_1^2 + 3x_2^2 \leqslant 3$
- nonconvexity $\sin(10x_1x_2) \leqslant 0$

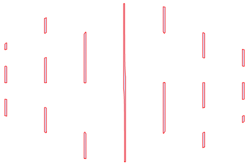


What is Mixed-Integer Nonlinear Programming?

$$
\begin{aligned}
\min\ & c^T x & & c \in \mathbb{R}^n \\
\text{such that } & g(x) \leqslant 0, & & g \in C^1(\mathbb{R}^n, \mathbb{R}^m) \\
& x_i \in \mathbb{Z}, \qquad i \in \mathcal{I}. & & \mathcal{I} \subseteq \{1, \dots, n\}
\end{aligned}
$$

MINLP is difficult due to combination of

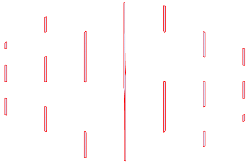
- ▷ nonlinearity $x_1^2 + 3x_2^2 \leqslant 3$
- ▷ nonconvexity $\sin(10x_1x_2) \leqslant 0$
- ▷ integrality $x_1 \in \mathbb{Z}$



$$\begin{aligned} \min\ & c^T x & & c \in \mathbb{R}^n \\ \text{such that } & g(x) \leqslant 0, & & g \in C^1(\mathbb{R}^n, \mathbb{R}^m) \\ & x_i \in \mathbb{Z}, \qquad i \in \mathcal{I}. & & \mathcal{I} \subseteq \{1, \dots, n\} \end{aligned}$$

MINLP is difficult due to combination of

- nonlinearity $x_1^2 + 3x_2^2 \leqslant 3$
- nonconvexity $\sin(10 x_1 x_2) \leqslant 0$
- integrality $x_1 \in \mathbb{Z}$



Important subclass: convex MINLP
MINLP is convex $\Leftrightarrow$ each function $g_j : \mathbb{R}^n \to \mathbb{R}$ is convex

Applications in many areas, e.g.,

- engineering design: e.g., mining with stockpiling constraints
- manufacturing: e.g., sheet metal design
- chemical industry: e.g., design of synthesis processes
- networks: operation and design of water and gas networks
- energy production and distribution: e.g., plant design, power scheduling
- logistics: e.g., public transport
- ...





$$\begin{aligned} \min \quad & c^T x \\ \text{such that} \quad & g(x) \leqslant 0 \\ & x_i \in \mathbb{Z}, \quad i \in \mathcal{I} \end{aligned}$$

Assumption: $g : \mathbb{R}^n \to \mathbb{R}^m$ is convex, each g_j continuously differentiable

NLP based branch–and–bound

▷ bounding: solve convex nonlinear relaxation (NLP)

$$\begin{aligned} \min \quad & c^T x \\ \text{such that} \quad & g(x) \leqslant 0 \end{aligned}$$

▷ branching: on integer variables with fractional LP value

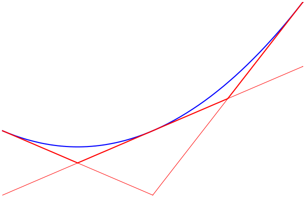
LP based branch–and–cut

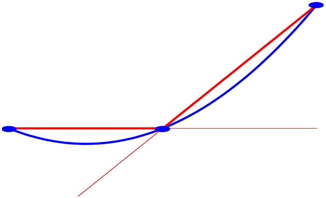
▷ bounding: solve polyhedral outer-approximation (LP)

$$\tilde{x} = \operatorname{argmin} \quad c^T x$$
$$\text{such that} \quad g_j(\hat{x}) + \nabla g_j(\hat{x})(x - \hat{x}) \leqslant 0, \quad j = 1, \dots, m, \quad \hat{x} \in S$$

If $g(\tilde{x}) > 0$, add supporting hyperplane to LP, i.e., add $\tilde{x}$ to S.

▷ branching: on integer variables with fractional LP value





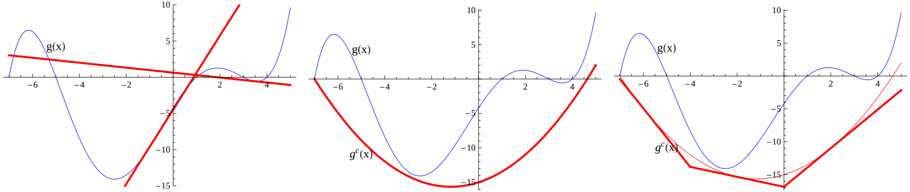
$$\begin{aligned} \min \quad & c^T x \\ \text{such that} \quad & g(x) \leqslant 0 \\ & x_i \in \mathbb{Z}, \quad i \in \mathcal{I} \end{aligned}$$

Now: some components of $g : \mathbb{R}^n \to \mathbb{R}^m$ may be nonconvex

$\Rightarrow$ inequalities $g_j(\hat{x}) + \nabla g_j(\hat{x})(x - \hat{x}) \leqslant 0$ may not be valid!

$\Rightarrow$ use convex underestimator: convex and below $g(x)$ for all $x \in [L, U]$

$\Rightarrow$ introduces convexification gap



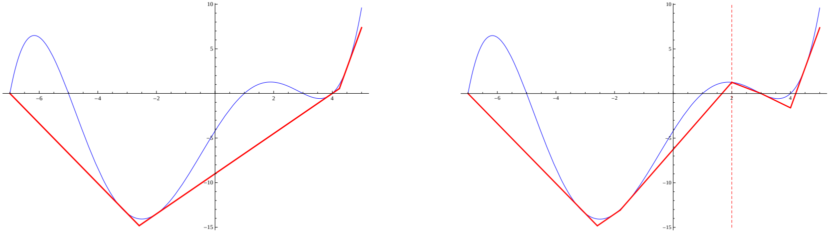
Spatial branch–and–bound

- **bounding:** solve polyhedral outer-approximation

$$
\begin{aligned}
\min \quad & c^T x \\
\text{such that} \quad & g_j^c(\hat{x}) + \nabla g_j^c(\hat{x})(x - \hat{x}) \leqslant 0, \quad j = 1, \ldots, m, \quad \hat{x} \in S, \\
& x \in [L, U]
\end{aligned}
$$

- **branching**: close gap between relaxation and problem
  - on integer variables with fractional value in LP relax
  - on continuous variables in nonconvex terms

    ⇒ tighter bounds ⇒ better underestimators



Finding feasible solutions. . .

- wait for the LP relaxation to become feasible
- MIP heuristics applied to LP
  - rounding, diving, feasibility pump,. . .
- extend MIP heuristics to MINLP
- MINLP specific heuristics → this talk

Why use primal heuristics inside an exact solver?

- Able to prove feasibility of the model
- Often nearly optimal solutions suffice in practice
- Feasible solutions guide remaining search process

Source		convex	nonconvex
MIP heuristics for linear outer approximations		✓	✓
NLP local search with fixed integralities		✓	✓
Simple NLP Rounding		✓	✓
Fractional Diving & Vectorlength Diving			
	BonamiGonçalves08	✓	(✓)
Iterative Rounding	NanniciniBelotti	✓	✓
FeasPump	BonamiCornuéjolsLodiMargot08	✓	nonconvex obj. convex feas. region
	D'AmbrosioFrangioniLibertiLodi09	✓	✓
	LinderothAbhishekLeyfferSartenaer08	✓	
Local Branching	NanniciniBelottiLiberti08	✓	✓
RECIPE	LibertiNanniciniMladenović08	✓	✓
RENS	BertholdHeinzVigerske09 (for MIQCPs)	✓	✓

⋮

- LNS: common paradigm in MIP heuristics

  fix a subset of variables $\rightsquigarrow$ easy subproblem $\rightsquigarrow$ solve

  MIP: “easy” $=$ few integralities MINLP: “easy” $=$ few nonlinearities

- Observation: Any MINLP can be reduced to a MIP by fixing (sufficiently many) variables.

  Experience: Often, few fixings are sufficient!

- Idea: Fix a small subset of variables to obtain a linear subproblem (MIP).
- Use solution of a LP or NLP relaxation to determine fixing values

Definition (cover of a function)

Let

- a function $g : D \to \mathbb{R}$, $x \mapsto g(x)$ on a domain $D \subseteq \mathbb{R}^n$,
- a point $x^\star \in D$, and
- a set $\mathcal{C} \subseteq \{1, \ldots, n\}$ of variable indices be given.

We call $\mathcal{C}$ an *$x^\star$-cover of g* if and only if the set

$$\{(x, g(x)) \mid x \in D, x_k = x_k^\star \text{ for all } k \in \mathcal{C}\} \tag{1}$$

is affine.

We call $\mathcal{C}$ a *(global) cover of g* if and only if $\mathcal{C}$ is an $x^\star$-cover of g for all $x^\star \in D$.

Definition (cover of an MINLP)

Let

- P be an MINLP
- $x^\star \in [L, U]$, and
- $\mathcal{C} \subseteq \{1, \ldots, n\}$ be a set of variable indices of P.

We call $\mathcal{C}$ an *$x^\star$-cover of P* if and only if $\mathcal{C}$ is an $x^\star$-cover for $g_1, \ldots, g_m$.

We call $\mathcal{C}$ a *(global) cover of P* if and only if $\mathcal{C}$ is an $x^\star$-cover of P for all $x^\star \in [L, U]$.

A generic algorithm

Input: MINLP P

begin

compute a solution $x^\star$ of an approximation of P

round $x^\star_k$ for all $k \in \mathcal{I}$

determine an $x^\star$-cover $\mathcal{C}$ of P

solve the sub-MIP of P given by fixing $x_k = x^\star_k$ for all $k \in \mathcal{C}$

end

A generic algorithm

```
Input: MINLP P
begin
    compute a solution x* of an approximation of P
    round x*_k for all k ∈ I
    determine an x*-cover C of P
    solve the sub-MIP of P given by fixing x_k = x*_k for all k ∈ C
end
```

Input: MINLP P
begin
  compute a solution $x^\star$ of an approximation of P
  round $x_k^\star$ for all $k \in \mathcal{I}$
  determine an $x^\star$-cover $\mathcal{C}$ of P
  solve the sub-MIP of P given by fixing $x_k = x_k^\star$ for all $k \in \mathcal{C}$
end

A generic algorithm

Input: MINLP P
begin
  compute a solution $x^\star$ of an approximation of P
  round $x_k^\star$ for all $k \in \mathcal{I}$
  determine an $x^\star$-cover $\mathcal{C}$ of P
  solve the sub-MIP of P given by fixing $x_k = x_k^\star$ for all $k \in \mathcal{C}$
end

A generic algorithm

```
Input: MINLP P
begin
    compute a solution x*
    of an approximation of P
    round x*_k for all k ∈ I
    determine an
    x*-cover C of P
    solve the sub-MIP of P
    given by fixing x_k = x*_k
    for all k ∈ C
end
```

Input: MINLP P

begin

- compute a solution $x^\star$ of an approximation of P
- round $x_k^\star$ for all $k \in \mathcal{I}$
- determine an $x^\star$-cover $\mathcal{C}$ of P
- solve the sub-MIP of P given by fixing $x_k = x_k^\star$ for all $k \in \mathcal{C}$

end

A generic algorithm

Input: MINLP P

begin

compute a solution $x^\star$ of an approximation of P

round $x_k^\star$ for all $k \in \mathcal{I}$

determine an $x^\star$-cover $\mathcal{C}$ of P

solve the sub-MIP of P given by fixing $x_k = x_k^\star$ for all $k \in \mathcal{C}$

end

Remark:

- MIP heuristics: trade-off fixing many vs. few variables

  Here: Eliminate nonlinearities by fixing as few as possible variables $\rightarrow$ minimum $x^\star$-cover!

A generic algorithm

```
Input: MINLP P
begin
    compute a solution x* of an approximation of P
    round x*_k for all k ∈ I
    determine an x*-cover C of P
    solve the sub-MIP of P given by fixing x_k = x*_k for all k ∈ C
end
```

Remark:

- MIP heuristics: trade-off fixing many vs. few variables

  Here: Eliminate nonlinearities by fixing as few as possible variables $\rightarrow$ minimum $x^\star$-cover!
- How to find minimum cover?

Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Qx + qx + c,\ Q \in \mathbb{R}^{n\times n}$ symmetric, $x^\star \in \mathbb{R}^n$.

Fixing variables with indices in $\mathcal{C} \subseteq \{1, ..., n\}$ transforms

$$x^T Qx \quad \overset{x_k = x_k^\star\ \forall k \in \mathcal{C}}{\rightsquigarrow} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

with $y = (x_k)_{k \notin \mathcal{C}} \in \mathbb{R}^{n-|\mathcal{C}|}$, and $\tilde{Q} = (Q_{uv})_{u,v \notin \mathcal{C}} \in \mathbb{R}^{(n-|\mathcal{C}|)\times(n-|\mathcal{C}|)}$, ...

Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Qx + qx + c,\ Q \in \mathbb{R}^{n \times n}$ symmetric, $x^\star \in \mathbb{R}^n$.

Fixing variables with indices in $\mathcal{C} \subseteq \{1, ..., n\}$ transforms

$$x^T Qx \quad \overset{x_k = x_k^\star \,\forall k \in \mathcal{C}}{\rightsquigarrow} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

with $y = (x_k)_{k \notin \mathcal{C}} \in \mathbb{R}^{n-|\mathcal{C}|}$, and $\tilde{Q} = (Q_{uv})_{u,v \notin \mathcal{C}} \in \mathbb{R}^{(n-|\mathcal{C}|)\times(n-|\mathcal{C}|)}, \ldots$

Thus: $\mathcal{C}$ is a cover of g iff

$q_{uv} = 0$ for all $u, v \notin \mathcal{C}$

independent of fix. values.

Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Qx + qx + c,\ Q \in \mathbb{R}^{n\times n}$ symmetric, $x^\star \in \mathbb{R}^n$.

Fixing variables with indices in $\mathcal{C} \subseteq \{1, ..., n\}$ transforms

$$x^T Qx \quad \underset{}{\overset{x_k = x_k^\star\ \forall k \in \mathcal{C}}{\rightsquigarrow}} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

with $y = (x_k)_{k \notin \mathcal{C}} \in \mathbb{R}^{n-|\mathcal{C}|}$, and $\tilde{Q} = (Q_{uv})_{u,v \notin \mathcal{C}} \in \mathbb{R}^{(n-|\mathcal{C}|)\times(n-|\mathcal{C}|)}$, ...

Thus: $\mathcal{C}$ is a cover of g iff

$q_{uv} = 0$ for all $u, v \notin \mathcal{C}$

independent of fix. values.

$\leftrightsquigarrow$

set covering:

cover nonzeros in Q by incident rows/columns

$$\begin{pmatrix} & * & & & \\ * & * & * & & \\ & * & & * & * \\ & & * & & \\ & & * & & \end{pmatrix}$$

Covering quadratic functions

Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Qx + qx + c,\ Q \in \mathbb{R}^{n\times n}$ symmetric, $x^\star \in \mathbb{R}^n$.

Fixing variables with indices in $\mathcal{C} \subseteq \{1, ..., n\}$ transforms

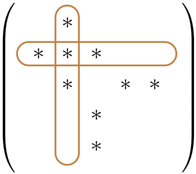
$$x^T Qx \quad \overset{x_k = x_k^\star\ \forall k \in \mathcal{C}}{\rightsquigarrow} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

with $y = (x_k)_{k \notin \mathcal{C}} \in \mathbb{R}^{n-|\mathcal{C}|}$, and $\tilde{Q} = (Q_{uv})_{u,v \notin \mathcal{C}} \in \mathbb{R}^{(n-|\mathcal{C}|)\times(n-|\mathcal{C}|)}, \ldots$

Thus: $\mathcal{C}$ is a cover of g iff

$q_{uv} = 0$ for all $u, v \notin \mathcal{C}$ $\quad\leftrightsquigarrow\quad$ **set covering:** cover nonzeros in Q by incident rows/columns

independent of fix. values.



Covering quadratic functions

Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Qx + qx + c$, $Q \in \mathbb{R}^{n \times n}$ symmetric, $x^\star \in \mathbb{R}^n$.

Fixing variables with indices in $\mathcal{C} \subseteq \{1, ..., n\}$ transforms

$$x^T Qx \quad \overset{x_k = x_k^\star \; \forall k \in \mathcal{C}}{\rightsquigarrow} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

with $y = (x_k)_{k \notin \mathcal{C}} \in \mathbb{R}^{n-|\mathcal{C}|}$, and $\tilde{Q} = (Q_{uv})_{u,v \notin \mathcal{C}} \in \mathbb{R}^{(n-|\mathcal{C}|)\times(n-|\mathcal{C}|)}$, ...

Thus: $\mathcal{C}$ is a cover of g iff

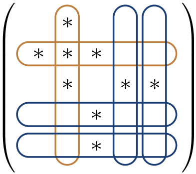
$q_{uv} = 0$ for all $u, v \notin \mathcal{C}$

independent of fix. values.

$\leftrightsquigarrow$

set covering:

cover nonzeros in Q by incident rows/columns



Covering quadratic functions

Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Qx + qx + c$, $Q \in \mathbb{R}^{n\times n}$ symmetric, $x^\star \in \mathbb{R}^n$.

Fixing variables with indices in $\mathcal{C} \subseteq \{1, \ldots, n\}$ transforms

$$x^T Qx \quad \overset{x_k = x_k^\star \,\forall k \in \mathcal{C}}{\rightsquigarrow} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

with $y = (x_k)_{k \notin \mathcal{C}} \in \mathbb{R}^{n-|\mathcal{C}|}$, and $\tilde{Q} = (Q_{uv})_{u,v \notin \mathcal{C}} \in \mathbb{R}^{(n-|\mathcal{C}|)\times(n-|\mathcal{C}|)}$, ...

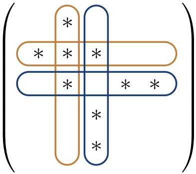
Thus: $\mathcal{C}$ is a cover of g iff

$q_{uv} = 0$ for all $u, v \notin \mathcal{C}$ $\quad\leftrightsquigarrow$

independent of fix. values.

set covering:

cover nonzeros
in Q by incident
rows/columns



Auxiliary binary variables:

$$\alpha_k = 1 :\Leftrightarrow x_k \text{ is fixed in } P$$

Set Covering constraints:

$\mathcal{C}(\alpha) := \{k \mid \alpha_k = 1\}$ is a cover of P if and only if

$$\alpha_k = 1 \qquad \text{for all square nonzeros: } Q^i_{kk} \neq 0, \tag{2}$$

$$\alpha_k + \alpha_j \geqslant 1 \qquad \text{for all bilinear nonzeros: } Q^i_{kj} \neq 0, k \neq j. \tag{3}$$

To find a minimum cover, we solve the covering problem

$$\min \Big\{ \sum_{k=1}^{n} \alpha_k : (2), (3), \alpha \in \{0,1\}^n \Big\}. \tag{4}$$

Auxiliary binary variables:

$$\alpha_k = 1 :\Leftrightarrow x_k \text{ is fixed in } P$$

Set Covering constraints:

$\mathcal{C}(\alpha) := \{k \mid \alpha_k = 1\}$ is a cover of P if and only if

$$\alpha_k = 1 \qquad \text{for all square nonzeros: } Q^i_{kk} \neq 0, \tag{2}$$

$$\alpha_k + \alpha_j \geqslant 1 \qquad \text{for all bilinear nonzeros: } Q^i_{kj} \neq 0, k \neq j. \tag{3}$$

To find a minimum cover, we solve the covering problem

$$\min\Big\{\sum_{k=1}^{n} \alpha_k : (2), (3), \alpha \in \{0,1\}^n\Big\}. \tag{4}$$

Auxiliary binary variables:

$$\alpha_k = 1 :\Leftrightarrow x_k \text{ is fixed in } P$$

Set Covering constraints:

$\mathcal{C}(\alpha) := \{k \mid \alpha_k = 1\}$ is a cover of P if and only if

$$\alpha_k = 1 \qquad \text{for all square nonzeros: } Q^i_{kk} \neq 0, \tag{2}$$

$$\alpha_k + \alpha_j \geqslant 1 \qquad \text{for all bilinear nonzeros: } Q^i_{kj} \neq 0, k \neq j. \tag{3}$$

To find a minimum cover, we solve the covering problem

$$\min \Big\{ \sum_{k=1}^{n} \alpha_k : (2), (3), \alpha \in \{0,1\}^n \Big\}. \tag{4}$$

- (4) is an $\mathcal{NP}$-hard problem, but standard branch-and-cut is (empirically) very fast.

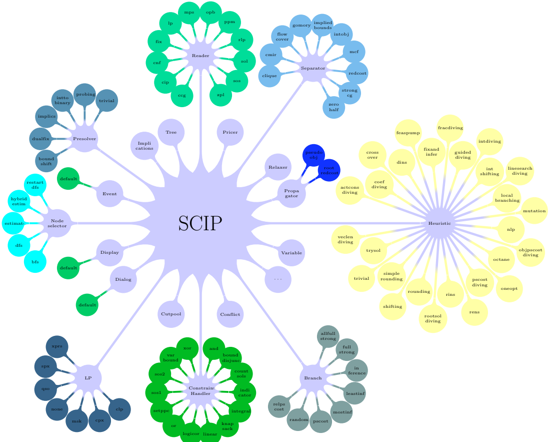
- For general MINLPs, the covering problem becomes more difficult, e.g. for a global cover of a monomial $x_1^{p_1} \cdots x_n^{p_n}$, $p_1, \ldots, p_n \in \mathbb{N}_0$:

$$\alpha_k = 1 \qquad \text{for all } p_k \geqslant 2$$

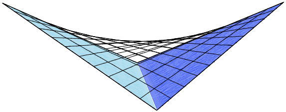
$$\sum_{k:p_k=1} (1 - \alpha_k) \leqslant 1.$$

- For general MINLPs, global covers become larger and larger. *However:* $x^\star$-covers are a weaker notion, may be significantly smaller

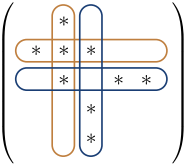
- a branch-cut-and-price framework
- incorporates CP, MIP, and SAT-solving features
- provides full-scale MIP solver
- modular structure via plugins
- free for academic purposes, `http://scip.zib.de`



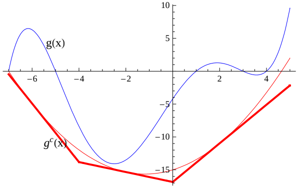
SCIP has recently been extended to handle nonconvex MIQCPs
$\Rightarrow$ all nonlinear constraints are of quadratic form $g_i(x) = x^T A_i x + b_i^T x + c_i$



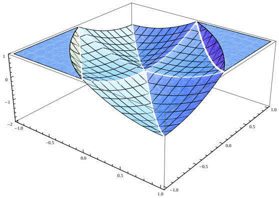
McCormick



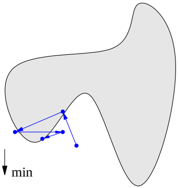
Undercover



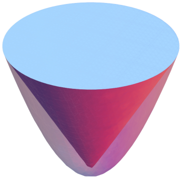
Convexification



Geometric branching

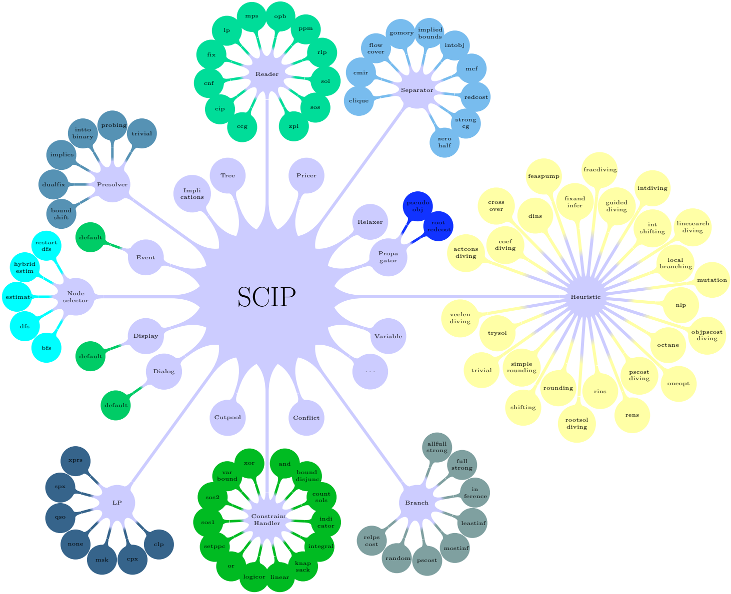


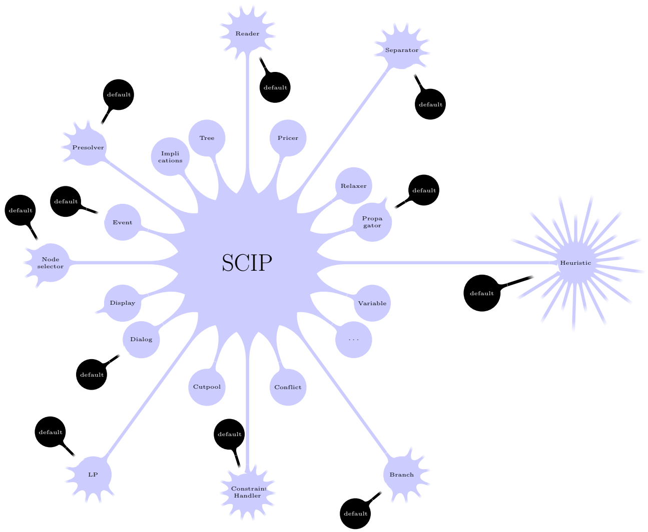
NLP local search

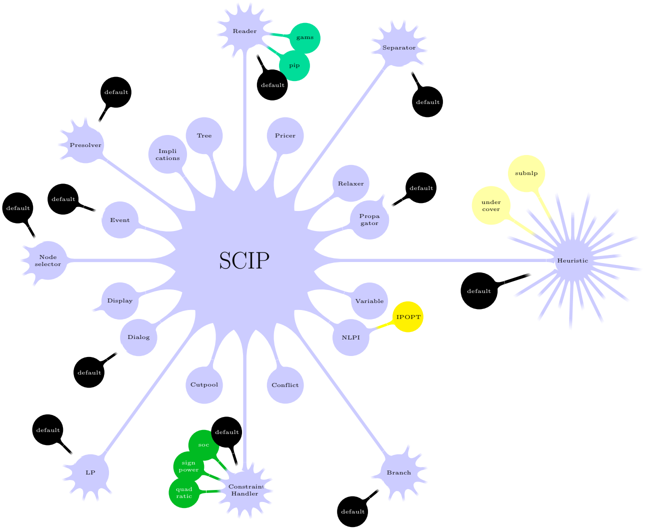


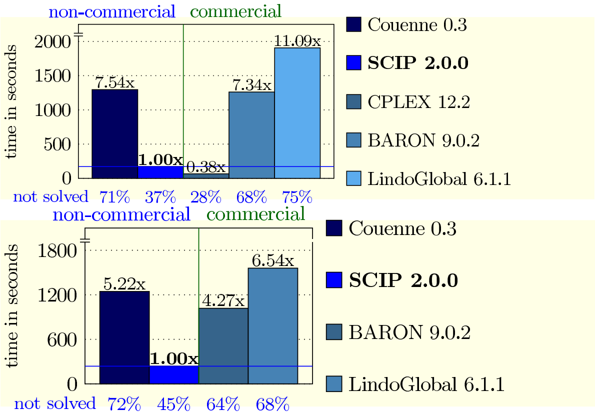
Second Order Cone

SCIP: MIQCP Plugins



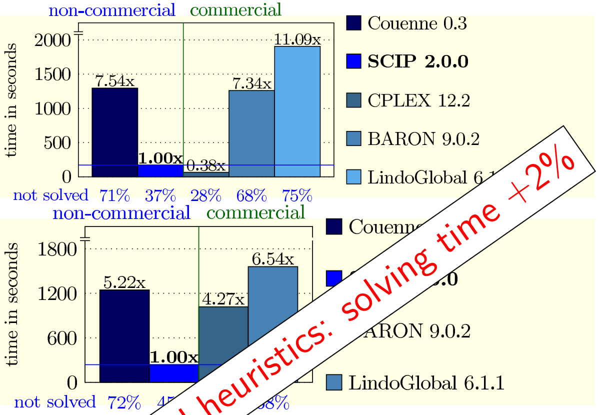

SCIP
Reader
default
Separator
default
default
Presolver
Tree
Pricer
Impli cations
Relaxer
default
Propa gator
default
Event
default
Node selector
Heuristic
default
Display
Variable
Dialog
...
default
Cutpool
Conflict
default
default
LP
Constraint Handler
Branch
default


SCIP
Reader
gams
pip
default
Separator
default
Presolver
default
Tree
Impli cations
Pricer
Relaxer
Propa gator
default
subnlp
under cover
Heuristic
default
Event
default
Node selector
default
Display
Dialog
default
Variable
NLPI
IPOPT
Cutpool
Conflict
LP
default
soc
default
sign power
quad ratic
Constraint Handler
Branch
default



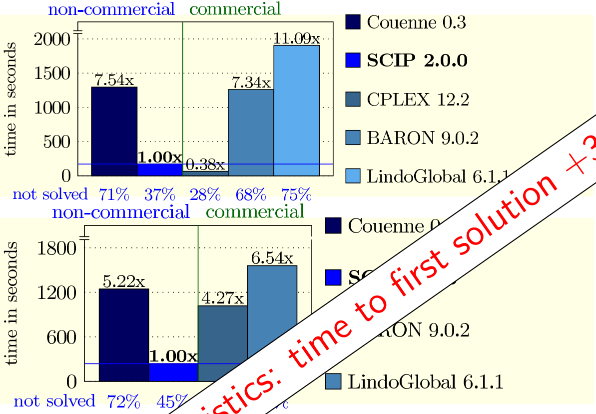
- 94 publicly available instances from 7 sources
- SCIP uses Cplex as LP solver and Ipopt as NLP solver
- 1 hour time limit

SCIP: Computational Results



- 94 publicly availabl… …nces from 7 sources
- SCIP uses Cpl… …P solver and Ipopt as NLP solver
- 1 hour tim… …t

Impact primal heuristics: solving time +2%



Impact primal heuristics: time to first solution +347%

- 94 publicly available … ces from 7 sources
- SCIP uses Cple… solver and Ipopt as NLP solver
- 1 hour tim…

▷ Goal
- evaluate potential as start heuristic at the root node

▷ Test set
- 33 MIQCP instances from MINLPLib

▷ Undercover parameters
- running as only root node heuristic in SCIP
- for sub-MIP: emphasis feasibility and fast presolving settings

▷ Reference solvers
- SCIP 1.2.1.1
- BARON 9.02
- Couenne 0.2
- default, node limit 1

▷ Reported
- nonlinear nonzeros/variable
- % variables fixed by Undercover
- solution values of each solver
- best known solution

12 instances with ⩽ 5% variables fixed

instance	nnz/var	% cov	UC	SCIP	BARON	Couenne	known
netmod_dol1	0.00	0.30	0	**-0.26321**	0	–	-0.56
netmod_dol2	0.00	0.38	-0.07802	**-0.50562**	0	–	-0.56
netmod_kar1	0.01	0.88	**0**	**0**	**0**	–	-0.4198
netmod_kar2	0.01	0.88	**0**	**0**	**0**	–	-0.4198
space25	0.12	1.04	–	–	–	–	484.33
ex1266	0.40	3.03	**16.3**	–	–	–	16.3
util	0.07	3.13	**999.58**	1000.5	1006.5	–	999.58
feedtray2	10.70	3.26	–	–	**0**	–	0
ex1265	0.38	3.52	**15.1**	–	–	**15.1**	10.3
ex1263	0.34	3.88	**30.1**	–	–	–	19.6
tln12	1.70	3.99	–	–	–	–	90.5
ex1264	0.36	4.26	**11.1**	–	–	–	8.6

- ▷ 9 instances feasible, 7 times best solution value
- ▷ `ex1266` and `util` optimal

10 instances with 5–15% variables fixed

instance	nnz/var	% cov	UC	SCIP	BARON	Couenne	known
waste	1.10	5.65	**608.76**	–	712.301	–	598.92
space25a	0.29	5.84	–	–	–	–	484.33
nuclear14a	4.98	6.43	–	–	–	–	-1.1280
nuclear14b	2.42	6.43	–	–	–	**-1.1105**	-1.1135
tln7	1.53	6.67	**30.3**	–	–	–	15
tln6	1.47	7.69	**20.3**	–	–	–	15.3
tloss	1.47	7.89	**16.3**	–	–	–	16.3
tln5	1.39	9.09	15.1	–	–	**14.5**	10.3
sep1	0.40	10.53	**-510.08**	–	**-510.08**	**-510.08**	-510.08
tltr	1.10	12.50	**74.2**	–	–	–	48.067

- 7 instances feasible, 6 times best solution value
- `tloss` and `sep1` optimal

11 instances with 15–96% variables fixed

instance	nnz/var	% cov	UC	SCIP	BARON	Couenne	known
nous1	2.39	19.44	–	–	–	**1.5671**	1.5671
nous2	2.39	19.44	–	1.3843	**0.62597**	1.3843	0.626
meanvarx	0.19	23.33	15.925	**14.369**	**14.369**	18.702	14.369
product2	0.37	26.15	–	–	–	–	-2102.4
product	0.17	30.87	–	–	–	–	-2142.9
spectra2	3.43	35.71	31.981	**13.978**	119.87	–	13.978
fac3	0.81	78.26	13065	**7213**	38329	–	3198
nvs19	8.00	88.89	–	0	**-1098**	–	-1098.4
nvs23	9.00	90.00	–	0	**-1124.8**	–	-1125.2
du-opt5	0.95	94.74	3407.1	**14.168**	–	1226.0	8.0737
du-opt	0.95	95.24	4233.9	4233.9	108.33	**41.304**	3.5563

▷ 5 instances feasible, no best solution value

- Feasible solutions
  - Undercover: 21 instances
  - SCIP: 13 instances
  - BARON: 15 instances
  - Couenne: 9 instances
  - All: 27 instances

- Solution quality
  if both found a solution
  - Undercover : SCIP = 1:6 (2 equal)
  - Undercover : BARON = 5:2 (3 equal)
  - Undercover : Couenne = 1:3 (2 equal)

- Undercover time always < 0.2 seconds (except for `waste` with 1.1 sec)
  - covering problem always solved to optimality at root
  - most time spent in sub-MIP
  - 20 of 21 feasible sub-MIPs solved to optimality
  - infeasibility of sub-MIP usually detected in advance (10 of 12)

Outline

Fix-and-propagate

- Do not fix variables in $\mathcal{C}$ simultaneously, but sequentially and propagate after each fixing.
- If $x_k^\star$ falls out of bounds then
  - fix to the closest bound (similar to FischettiSalvagnin09)
  - recompute the approximation

Backtracking

- If fix-and-propagate deduces infeasibility, apply a one-level backtracking: undo last fixing and try another value

Fix-and-propagate

- Do not fix variables in $\mathcal{C}$ simultaneously, but sequentially and pro ng.
- If $x_k^\star$ falls out of bounds
  - fix to the closest bound vagnin09)
  - recompute the approxim

Backtracking

- If fix-and-propagate deduces infeasibility, apply a one-level backtracking: undo last fixing and try another value

Using different covers

Covers minimising different impact measures

- minimum cardinality covers: minimise impact on MINLP

- Alternative impact measures as objective function of covering problem:
  - appearance in nonlinear terms
  - appearance in violated nonlinear constraints
  - domain size
  - variable type
  - rounding locks on integer variables
  - hybrid measures

- In particular: if a minimum cardinality cover yields infeasible sub-MIP

Covers minimising different impact measures

- minimum cardinality cov : on MINLP
- Alternative impact meas ction of covering problem:
  - appearance in nonlinear
  - appearance in violated n
  - domain size
  - variable type
  - rounding locks on intege
  - hybrid measures

- In particular: if a minimum cardinality cover yields infeasible sub-MIP

Recovering

- fix-and-propagate may fix variables outside the cover $\mathcal{C}$
- $\leadsto$ variables in $\mathcal{C}$ might not need to be fixed

  $\leadsto$ "re-cover": solve the covering problem again with propagated bounds

Recovering

- fix-and-propagate may fix va … le the cover $\mathcal{C}$
- $\rightsquigarrow$ variables in $\mathcal{C}$ might not … ‹ed

  $\rightsquigarrow$ "re-cover": solve the cov … ı again with propagated bounds

NLP postprocessing

- All sub-MIP solutions are fully feasible for the original MINLP.
- Still, sub-MIP solution $\tilde{x}$ could be improved by NLP local search:
  - fix all integer variables of the original MINLP to their values in $\tilde{x}$
  - solve the resulting NLP to local optimality

NLP postprocessing

▷ All sub-MIP solutions ar … e original MINLP.

▷ Still, sub-MIP solution $\tilde{x}$ … by NLP local search:

- fix all integer variables c … o their values in $\tilde{x}$
- solve the resulting NLP

If the sub-MIP is infeasible, this is typically detected

- during fix-and-propagate, or
- via infeasible root LP.

$\rightsquigarrow$ Generate conflict clauses for the original MINLP .

- Add them to the original MINLP.
- Use them to revise fixing values and/or fixing order.
- Start another fix-and-propagate run.

If the sub-MIP remains infeasible, at least this gives us valid conflicts to prune the search tree in the original problem.

Avoiding/exploiting infeasibility

If the sub-MIP is infeasible, this is typically detected

- during fix-and-propagate, or
- via infeasible root

⤳ Generate conflict

- Add them to the o
- Use them to revise r.
- Start another fix-a

If the sub-MIP remains infeasible, at least this gives us valid conflicts to prune the search tree in the original problem.

Variations: convexification & domain reduction

- Idea of Undercover: fix few variables to obtain an “easy” subproblem.
  - switch to easier problem class
  - switch to easier problem of the same class
- Switch to easier problem class:
  - MIQCP $\rightsquigarrow$ MIP
  - MINLP $\rightsquigarrow$ MIQCP
  - nonconvex MINLP $\rightsquigarrow$ convex MINLP
  - ...
- Switch to easier problem of the same class: restrict variable domains
  - significantly better outer approximations
  - leaves more freedom to the problem

Variations: convexification & domain reduction

- Idea of Undercover: fix few variables to obtain an “easy” subproblem.
  - switch to easier problem class
  - switch to easier p
- Switch to easier pr
  - MIQCP $\rightsquigarrow$ MIP
  - MINLP $\rightsquigarrow$ MIQC
  - nonconvex MINL
  - ...

- Switch to easier problem of the same class: restrict variable domains
  - significantly better outer approximations
  - leaves more freedom to the problem

- Scheme of a general-purpose start heuristic for MINLP
  - solve a set covering problem
  - to identify few variable fixings
  - yielding a mixed-integer linear subproblem
- Preliminary experiments
  - MIQCPs from MINLPLib – often few fixings sufficient: $\leqslant 5\%$ on $1/3$ of the test set, $\leqslant 15\%$ on $2/3$ of the test set
  - successfully applied as root node heuristic
- Future research
  - extensions and variations
  - experiments on general MINLPs
  - tuning for efficient use within branch-and-bound tree
  - use NLP relaxation instead of LP outer approximation

Undercover

A primal heuristic for MINLP based on sub-MIPs generated by set covering

Timo Berthold

Zuse Institute Berlin • MATHEON • BMS

joint work with Ambros M. Gleixner

DFG Research Center MATHEON
Mathematics for key technologies