Solving Mixed-Integer Nonlinear Programs
(with SCIP)

Ambros M. Gleixner

Zuse Institute Berlin · Matheon · Berlin Mathematical School

5th Porto Meeting on Mathematics for Industry, April 10–11, 2014, Porto

Outline

Solving MINLPs (with SCIP)

Solving convex MINLPs

Solving nonconvex MINLPs

Modeling, Reformulation, Presolving

Primal Solutions: The Undercover Heuristic

Outline

Solving MINLPs (with SCIP)

Solving convex MINLPs

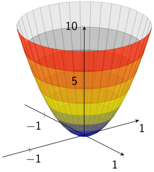
Solving nonconvex MINLPs

Modeling, Reformulation, Presolving

Primal Solutions: The Undercover Heuristic

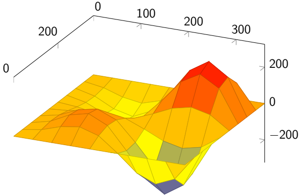
What is Mixed-Integer Nonlinear Programming?

$$
\begin{array}{lll}
\min & c^{\mathsf{T}} x & \text{for } c \in \mathbb{R}^n, \\
\text{s.t.} & \mathbf{g_k(x) \leqslant 0} & \text{for } k = 1, \dots, m,\ g_k : [\ell, u] \to \mathbb{R} \in C^1, \\
 & x \in [\ell, u], & \\
 & x_i \in \mathbb{Z} & \text{for } i \in \mathcal{I} \subseteq \{1, \dots, n\}.
\end{array}
$$



g_k **convex**

local = global optimality



g_k **nonconvex**

suboptimal local optima

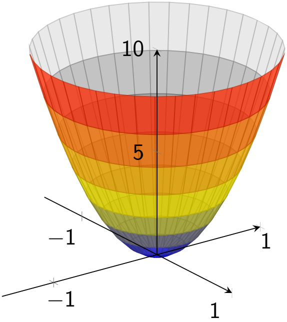
Convex MINLP

Assumption $g_1, \ldots, g_m$ convex

NLP-based

replace LP by NLP solver

branch on integer var.s with fractional NLP value



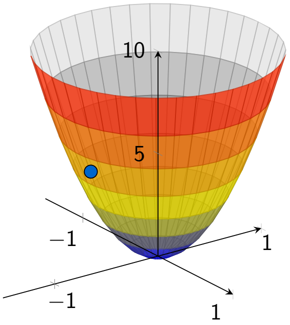
Convex MINLP

Assumption $g_1, \ldots, g_m$ convex

NLP-based

replace LP by NLP solver

branch on integer var.s with fractional NLP value



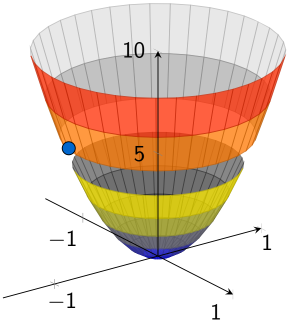
Convex MINLP

Assumption $g_1, \ldots, g_m$ convex

NLP-based

replace LP by NLP solver

branch on integer var.s with fractional NLP value



Convex MINLP

Assumption $g_1, \ldots, g_m$ convex

NLP-based

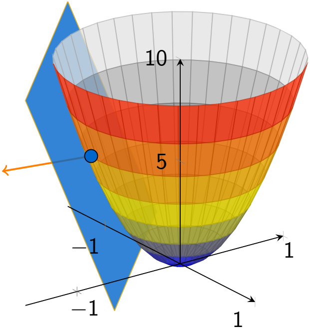
replace LP by NLP solver

branch on integer var.s with fractional NLP value

LP-based

underestimate by gradient cuts

$$g_k(\hat{x}) + \nabla g_k(\hat{x})^\mathsf{T}(x - \hat{x}) \leqslant 0$$



Convex MINLP

Assumption $g_1, \ldots, g_m$ convex

NLP-based

replace LP by NLP solver

branch on integer var.s with fractional NLP value

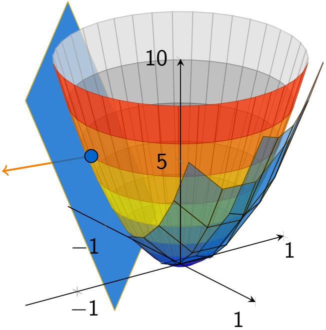
LP-based

underestimate by gradient cuts

$$g_k(\hat{x}) + \nabla g_k(\hat{x})^\mathsf{T}(x - \hat{x}) \leqslant 0$$

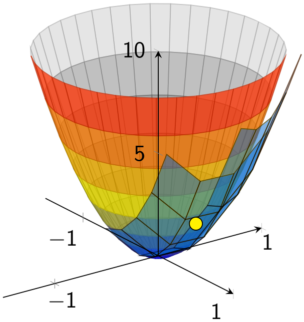
bound by polyhedral relaxation

- at MIP/NLP/sub-NLP solutions
- at node LP solutions



Convex MINLP

Assumption $g_1, \ldots, g_m$ convex



NLP-based

replace LP by NLP solver

branch on integer var.s with fractional NLP value

LP-based

underestimate by gradient cuts

$$g_k(\hat{x}) + \nabla g_k(\hat{x})^{\mathsf{T}}(x - \hat{x}) \leqslant 0$$

bound by polyhedral relaxation

- at MIP/NLP/sub-NLP solutions
- at node LP solutions

Many algorithms, many solvers

α-ECP [Westerlund and Pettersson], BONMIN [Bonami et al.], DICOPT [Duran and Grossmann], sBB [ARKI Software & Consulting], ...

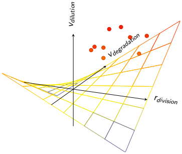
[see, e.g., Bonami, Biegler, Conn, Cornuéjols, Grossmann, Laird, Lee, Lodi, Margot, Sawaya, Wächter 2008]

Applications, applications, applications

- industrial engineering: mining with stockpiling constraints
- manufacturing: sheet metal design
- chemical industry: design of synthesis processes
- networks: operation and design of water and gas networks
- energy production and distribution: plant design, power scheduling
- biological engineering: cell modeling
- ...

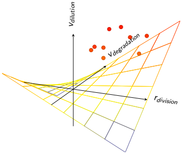






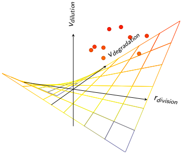
Applications, applications, applications

- industrial engineering: mining with stockpiling constraints
- manufacturing: sheet metal design
- chemical industry: design of synthesis processes
- networks: operation and design of water and gas networks
- energy production and distribution: plant design, power scheduling
- biological engineering: cell modeling
- ...



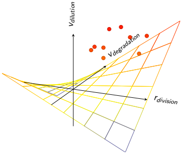
Applications, applications, applications

- industrial engineering: mining with stockpiling constraints
- manufacturing: sheet metal design
- chemical industry: design of synthesis processes
- networks: operation and design of water and gas networks
- energy production and distribution: plant design, power scheduling
- biological engineering: cell modeling
- ...



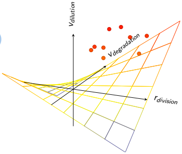
Applications, applications, applications

- industrial engineering: mining with stockpiling constraints
- manufacturing: sheet metal design
- chemical industry: design of synthesis processes
- networks: operation and design of water and gas networks
- energy production and distribution: plant design, power scheduling
- biological engineering: cell modeling
- . . .

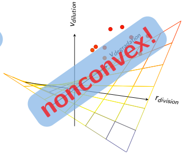
Applications, applications, applications

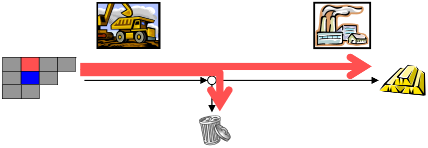
- industrial engineering: mining with stockpiling constraints
- manufacturing: sheet metal design
- chemical industry: design of synthesis processes
- networks: operation and design of water and gas networks
- energy production and distribution: plant design, power scheduling
- biological engineering: cell modeling
- ...



Applications, applications, applications

- industrial engineering: mining with stockpiling constraints
- manufacturing: sheet metal design
- chemical industry: design of synthesis processes
- networks: operation and design of water and gas networks
- energy production and distribution: plant design, power scheduling
- biological engineering: cell modeling
- ...





Variables:

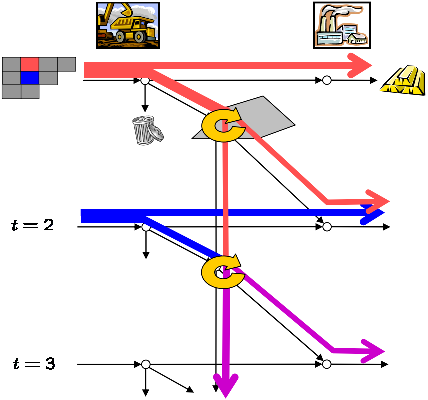
$x_{i,t} \in \{0,1\}$ block i fully mined by t

$f^{m}_{i,t} \in [0,1]$ % of block i mined in t

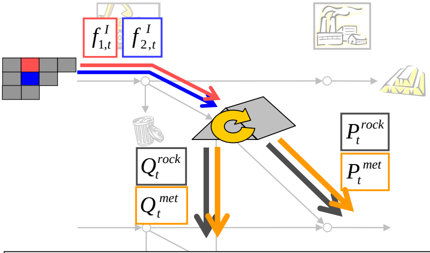
$f^{p}_{i,t} \in [0,1]$ % of block i processed in t

Constraints:

- material flow conservation
- mining & processing capacities
- mining precedences


$t = 2$
$t = 3$

Open Pit Mine Production Scheduling with Stockpiles



Aggregated stockpile model

$f^{I}_{i,t} \in [0,1]$ % of block i into stockpiled

Q_t^{rock}, Q_t^{met} total rock / metal tons held

P_t^{rock}, P_t^{met} total rock / metal tons out

Mixing constraints:

$$\frac{P_t^{met}}{Q_t^{met}} = \frac{P_t^{rock}}{Q_t^{rock}}$$ (metal fraction out = rock fraction out)

Outline

Solving MINLPs (with SCIP)

Solving convex MINLPs

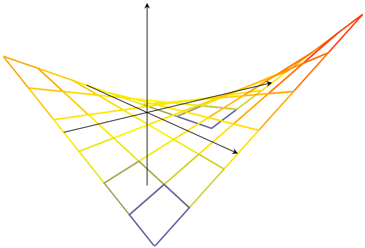
Solving nonconvex MINLPs

Modeling, Reformulation, Presolving

Primal Solutions: The Undercover Heuristic

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



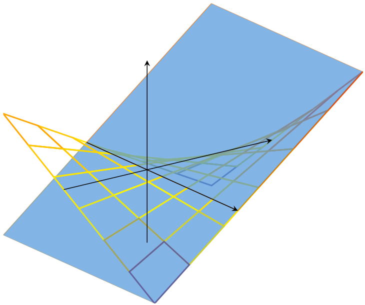
Relaxation

gradient cuts invalid

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \dots, g_m$ nonconvex

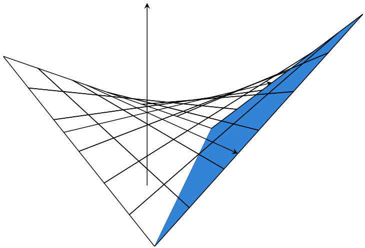


Relaxation
gradient cuts invalid

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

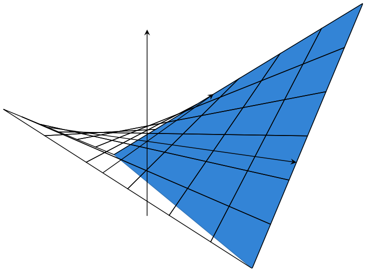
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

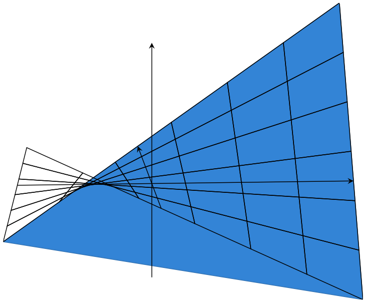
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, . . .]

Nonconvex MINLP

Now some $g_1, \dots, g_m$ nonconvex



Relaxation

gradient cuts invalid

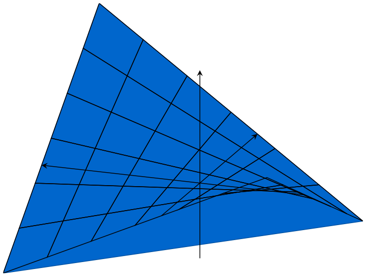
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

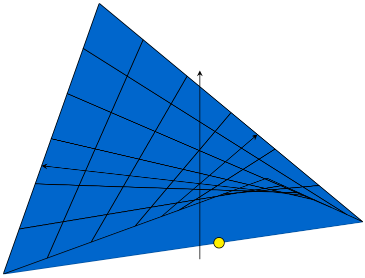
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

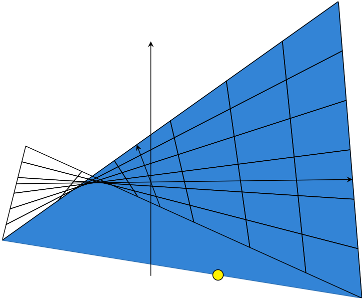
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

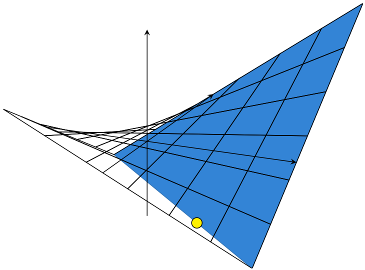
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

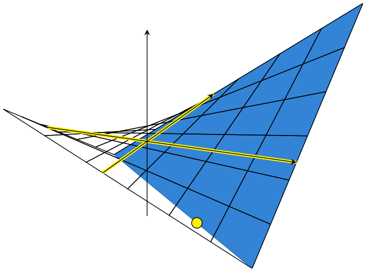
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, . . .]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

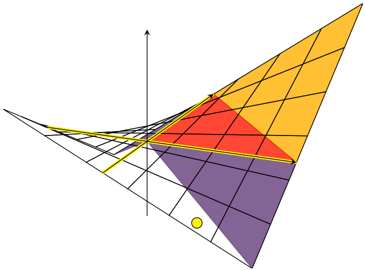
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

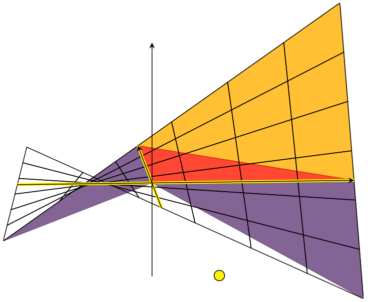
linear relaxation of convex hull

convexification gap

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, . . .]

Nonconvex MINLP

Now some $g_1, \ldots, g_m$ nonconvex



Relaxation

gradient cuts invalid

linear relaxation of convex hull

convexification gap

Spatial branch-and-bound

branch on int. variables with fractional LP value

branch on variables in violated nonlinear constraints

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, ...]

Convex Relaxation

Convex envelopes

- largest convex function that underestimates some $g_j(x)$
- difficult to find in general
- known for many elementary cases: convex, univariate concave, bilinear, ...

Convex Relaxation

Convex envelopes

- largest convex function that underestimates some $g_j(x)$
- difficult to find in general
- known for many elementary cases: convex, univariate concave, bilinear, ...

Example

McCormick underestimators for $x_1 x_2$

$$(x_1 - \ell_1) \cdot (x_2 - \ell_2) \geqslant 0$$

Convex Relaxation

Convex envelopes

- largest convex function that underestimates some $g_j(x)$
- difficult to find in general
- known for many elementary cases: convex, univariate concave, bilinear, ...

Example

McCormick underestimators for $x_1 x_2$

$$(x_1 - \ell_1) \cdot (x_2 - \ell_2) \geqslant 0$$
$$x_1 x_2 - \ell_1 x_2 - \ell_2 x_1 + \ell_1 \ell_2 \geqslant 0$$

Convex Relaxation

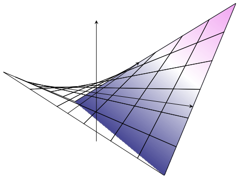
Convex envelopes

- ▷ largest convex function that underestimates some $g_j(x)$
- ▷ difficult to find in general
- ▷ known for many elementary cases: convex, univariate concave, bilinear, ...

Example

McCormick underestimators for $x_1 x_2$

$$(x_1 - \ell_1) \cdot (x_2 - \ell_2) \geqslant 0$$
$$x_1 x_2 - \ell_1 x_2 - \ell_2 x_1 + \ell_1 \ell_2 \geqslant 0$$
$$x_1 x_2 \geqslant \ell_1 x_2 + \ell_2 x_1 - \ell_1 \ell_2$$

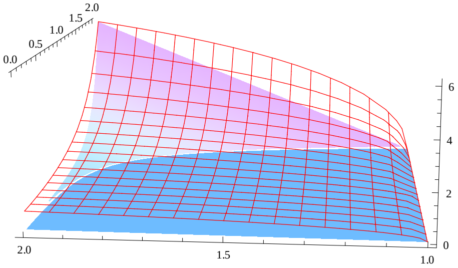


Convex Relaxation

Factorable functions

- recursive sum of products of univariate functions)
- reformulate into simple cases by introducing new variables and equations

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$



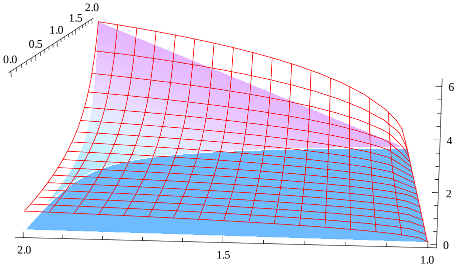
Convex Relaxation

Factorable functions

- recursive sum of products of univariate functions)
- reformulate into simple cases by introducing new variables and equations

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$

$$g = \sqrt{y_1}$$
$$y_1 = y_2 y_3$$
$$y_2 = \exp(y_4)$$
$$y_3 = \ln(x_2)$$
$$y_4 = x_1^2$$



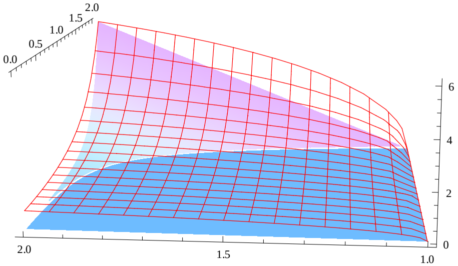
Convex Relaxation

Factorable functions

- recursive sum of products of univariate functions)
- reformulate into simple cases by introducing new variables and equations

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$

$$g = \sqrt{y_1}$$
$$y_1 = y_2 y_3$$
$$y_2 = \exp(y_4)$$
$$y_3 = \ln(x_2)$$
$$y_4 = x_1^2$$

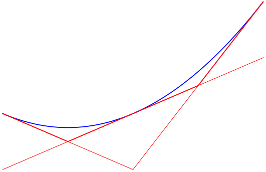


Tighter relaxations

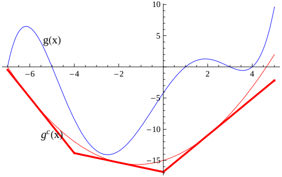
Reformulation-Linearization-Technique, SDP cuts, Disjunctive Programming, ...

General MINLP solving techniques

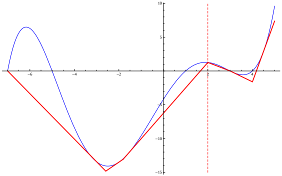
Gradient cuts



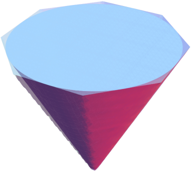
Underestimators



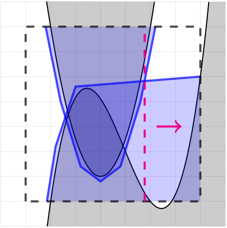
Spatial branching



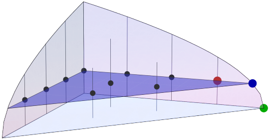
Presolving



Bound tightening

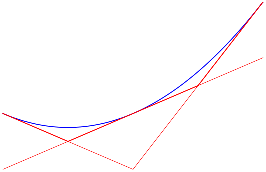


Primal heuristics

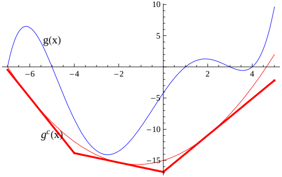


General MINLP solving techniques

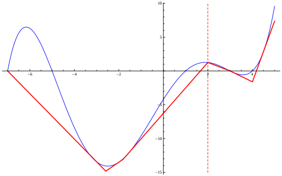
Gradient cuts



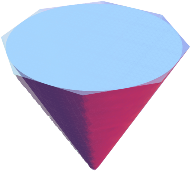
Underestimators



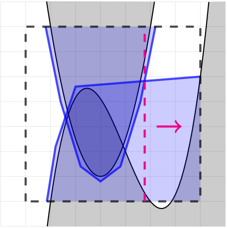
Spatial branching



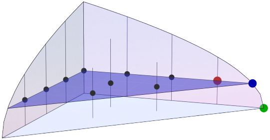
Presolving



Bound tightening



Primal heuristics



Outline

Solving MINLPs (with SCIP)

Modeling Considerations

Provide bounds on variables (as tight as possible)

- tighter relaxations

Modeling Considerations

Provide bounds on variables (as tight as possible)

- tighter relaxations

Scaling

- ideally: nonzeros with absolute values in the range $[0.01, 100]$
- also intermediate expressions are important:

$$\exp\left(-\frac{1}{x}\right) \in [0, 0.4] \quad \text{for} \quad x \in [10^{-6}, 1], \qquad \text{but} \quad \frac{1}{x} \in [1, 10^6]$$

Modeling Considerations

Provide bounds on variables (as tight as possible)

- tighter relaxations

Scaling

- ideally: nonzeros with absolute values in the range $[0.01, 100]$
- also intermediate expressions are important:

$$\exp\left(-\frac{1}{x}\right) \in [0, 0.4] \quad \text{for} \quad x \in [10^{-6}, 1], \qquad \text{but} \quad \frac{1}{x} \in [1, 10^6]$$

Prefer Linearity and Convexity

$$\frac{x}{y} = 1 \qquad \Rightarrow \text{nonlinear and nonconvex}$$

Modeling Considerations

Provide bounds on variables (as tight as possible)

- tighter relaxations

Scaling

- ideally: nonzeros with absolute values in the range $[0.01, 100]$
- also intermediate expressions are important:

$$\exp\left(-\frac{1}{x}\right) \in [0, 0.4] \quad \text{for} \quad x \in [10^{-6}, 1], \qquad \text{but} \quad \frac{1}{x} \in [1, 10^6]$$

Prefer Linearity and Convexity

$$x = y \qquad \Rightarrow \text{linear and thus convex}$$

Modeling Considerations

Provide bounds on variables (as tight as possible)

- tighter relaxations

Scaling

- ideally: nonzeros with absolute values in the range $[0.01, 100]$
- also intermediate expressions are important:

$$\exp\left(-\frac{1}{x}\right) \in [0, 0.4] \quad \text{for} \quad x \in [10^{-6}, 1], \qquad \text{but} \quad \frac{1}{x} \in [1, 10^{6}]$$

Prefer Linearity and Convexity

$$xy \geqslant 1 \qquad \Rightarrow \text{nonconvex}$$

Modeling Considerations

Provide bounds on variables (as tight as possible)

- tighter relaxations

Scaling

- ideally: nonzeros with absolute values in the range $[0.01, 100]$
- also intermediate expressions are important:

$$\exp\left(-\frac{1}{x}\right) \in [0, 0.4] \quad \text{for} \quad x \in [10^{-6}, 1], \qquad \text{but} \quad \frac{1}{x} \in [1, 10^6]$$

Prefer Linearity and Convexity

$$y \geqslant \frac{1}{x} \qquad \Rightarrow \text{convex}$$

Reformulation of products with binary variables

A quadratic term

$$x \cdot \sum_{k=1}^{N} a_k y_k \quad \text{with} \quad x \in \{0,1\}$$

can be linearly reformulated:

- auxiliary continuous variable w
- additional linear constraints

$$M^L x \leqslant w \leqslant M^U x,$$

$$\sum_{k=1}^{N} a_k y_k - M^U(1-x) \leqslant w \leqslant \sum_{k=1}^{N} a_k y_k - M^L(1-x),$$

where M^L and M^U are bounds on $\sum_{k=1}^{N} a_k y_k$.

Convexity check for quadratic constraints

A quadratic constraint $x^\mathsf{T} A x + b^\mathsf{T} x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Convexity check for quadratic constraints

A quadratic constraint $x^\mathsf{T} A x + b^\mathsf{T} x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leqslant 1$ in $[-1, 1] \times [-1, 1]$

Convexity check for quadratic constraints

A quadratic constraint $x^\mathsf{T} A x + b^\mathsf{T} x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

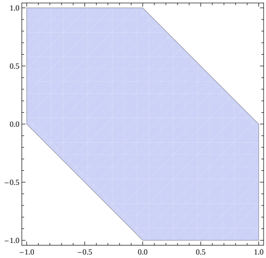
Example $x^2 + 2xy + y^2 \leqslant 1 \;\Leftrightarrow\; (x+y)^2 \leqslant 1$ in $[-1,1] \times [-1,1]$

Convexity check for quadratic constraints

A quadratic constraint $x^\mathsf{T} A x + b^\mathsf{T} x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leqslant 1 \quad \Leftrightarrow \quad |x + y| \leqslant 1$ in $[-1,1] \times [-1,1]$



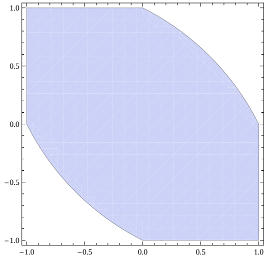
feasible region

Convexity check for quadratic constraints

A quadratic constraint $x^{\mathsf{T}} A x + b^{\mathsf{T}} x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leqslant 1 \;\Leftrightarrow\; (x+y)^2 \leqslant 1$ in $[-1,1] \times [-1,1]$



using McCormick underestimators:

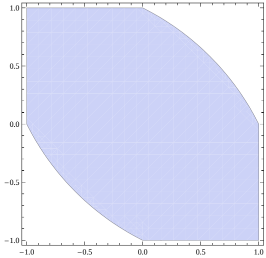
$$\left\{ \begin{array}{c} x^2 + 2w + y^2 \leqslant 1 \\ w \geqslant L^y x + L^x y - L^x L^y \\ w \geqslant U^y x + U^x y - U^x U^y \end{array} \right\}$$

Convexity check for quadratic constraints

A quadratic constraint $x^{\mathsf{T}}Ax + b^{\mathsf{T}}x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leqslant 1 \;\Leftrightarrow\; (x+y)^2 \leqslant 1$ in $[-1,1] \times [-1,1]$



using McCormick underestimators:

$$\left\{ \begin{array}{c} x^2 + 2w + y^2 \leqslant 1 \\ w \geqslant L^y x + L^x y - L^x L^y \\ w \geqslant U^y x + U^x y - U^x U^y \end{array} \right\}$$

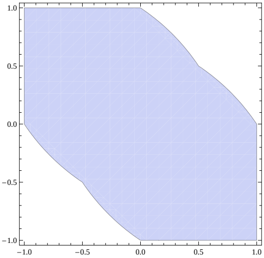
branched into 4 subproblems

Convexity check for quadratic constraints

A quadratic constraint $x^{\mathsf{T}}Ax + b^{\mathsf{T}}x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leqslant 1 \;\Leftrightarrow\; (x+y)^2 \leqslant 1$ in $[-1,1] \times [-1,1]$



using McCormick underestimators:

$$\left\{ \begin{array}{c} x^2 + 2w + y^2 \leqslant 1 \\ w \geqslant L^y x + L^x y - L^x L^y \\ w \geqslant U^y x + U^x y - U^x U^y \end{array} \right\}$$

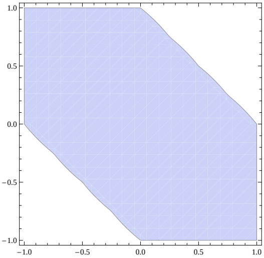
branched into 16 subproblems

Convexity check for quadratic constraints

A quadratic constraint $x^{\mathsf{T}}Ax + b^{\mathsf{T}}x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leqslant 1 \;\Leftrightarrow\; (x+y)^2 \leqslant 1$ in $[-1,1] \times [-1,1]$



using McCormick underestimators:

$$\left\{ \begin{array}{c} x^2 + 2w + y^2 \leqslant 1 \\ w \geqslant L^y x + L^x y - L^x L^y \\ w \geqslant U^y x + U^x y - U^x U^y \end{array} \right\}$$

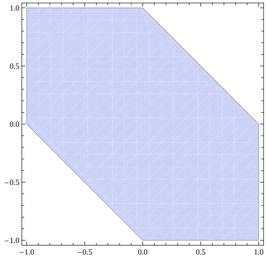
branched into 64 subproblems

Convexity check for quadratic constraints

A quadratic constraint $x^{\mathsf{T}}Ax + b^{\mathsf{T}}x \leqslant c$:

- convex if A is positive-semidefinite
- check by computing its minimal eigenvalue with `LAPACK`
- if yes: gradient cuts are valid
  $\Rightarrow$ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leqslant 1 \;\Leftrightarrow\; (x+y)^2 \leqslant 1$ in $[-1,1] \times [-1,1]$



$$A = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \text{ positive-semidefinite}$$

$\Rightarrow$ gradient cuts at 4 corners yield exact feasible region

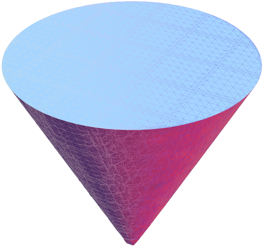
Second-order cone upgrade

Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leqslant 0 \;\Leftrightarrow\; \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leqslant \sqrt{\alpha_{N+1}} x_{N+1}$$

with $\alpha_1, \ldots, \alpha_{N+1} \geqslant 0, L_{N+1} \geqslant 0$ describe a convex feasible region.

Example $x^2 + y^2 - z^2 \leqslant 0$ in $[-1, 1] \times [-1, 1] \times [0, 1]$



feasible region

"ice cream cone"

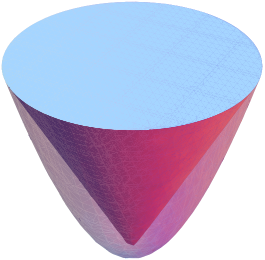
Second-order cone upgrade

Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leqslant 0 \;\Leftrightarrow\; \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leqslant \sqrt{\alpha_{N+1}} x_{N+1}$$

with $\alpha_1, \ldots, \alpha_{N+1} \geqslant 0, L_{N+1} \geqslant 0$ describe a convex feasible region.

Example $x^2 + y^2 - z^2 \leqslant 0$ in $[-1,1] \times [-1,1] \times [0,1]$



using secant underestimator:

$$\left\{ \begin{array}{c} x^2 + y^2 + w \leqslant 1 \\ w \geqslant \frac{(L^z)^2 - (U^z)^2}{U^z - L^z}(z - L^z) - (L^z)^2 \end{array} \right\}$$

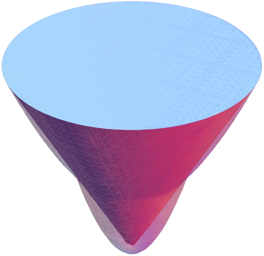
Second-order cone upgrade

Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leqslant 0 \;\Leftrightarrow\; \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leqslant \sqrt{\alpha_{N+1}}\, x_{N+1}$$

with $\alpha_1, \ldots, \alpha_{N+1} \geqslant 0, L_{N+1} \geqslant 0$ describe a convex feasible region.

Example $x^2 + y^2 - z^2 \leqslant 0$ in $[-1,1] \times [-1,1] \times [0,1]$



using secant underestimator:

$$\left\{ \begin{array}{c} x^2 + y^2 + w \leqslant 1 \\ w \geqslant \frac{(L^z)^2 - (U^z)^2}{U^z - L^z}(z - L^z) - (L^z)^2 \end{array} \right\}$$

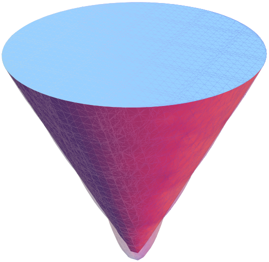
Second-order cone upgrade

Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leqslant 0 \;\Leftrightarrow\; \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leqslant \sqrt{\alpha_{N+1}} x_{N+1}$$

with $\alpha_1, \ldots, \alpha_{N+1} \geqslant 0, L_{N+1} \geqslant 0$ describe a convex feasible region.

Example $x^2 + y^2 - z^2 \leqslant 0$ in $[-1,1] \times [-1,1] \times [0,1]$



using secant underestimator:

$$\left\{ \begin{array}{c} x^2 + y^2 + w \leqslant 1 \\ w \geqslant \frac{(L^z)^2 - (U^z)^2}{U^z - L^z}(z - L^z) - (L^z)^2 \end{array} \right\}$$

after branching on $z = 0.25, 0.5, 0.75$

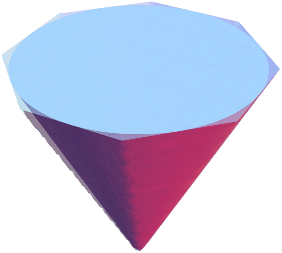
Second-order cone upgrade

Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leqslant 0 \;\Leftrightarrow\; \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leqslant \sqrt{\alpha_{N+1}} x_{N+1}$$

with $\alpha_1, \ldots, \alpha_{N+1} \geqslant 0$, $L_{N+1} \geqslant 0$ describe a convex feasible region.

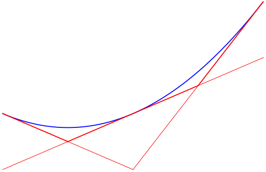
Example $x^2 + y^2 - z^2 \leqslant 0$ in $[-1,1] \times [-1,1] \times [0,1]$



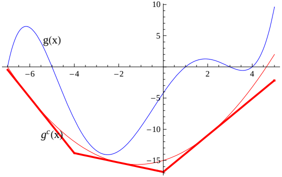
using gradient cuts at 8 corners

General MINLP solving techniques

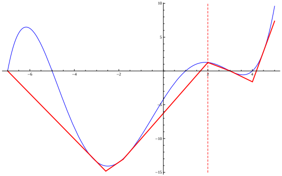
Gradient cuts



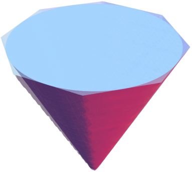
Underestimators



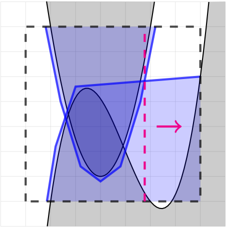
Spatial branching



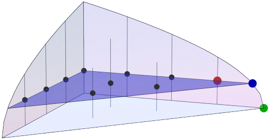
Presolving



Bound tightening

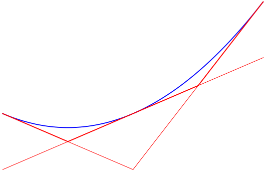


Primal heuristics

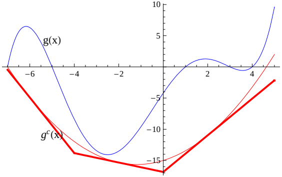


General MINLP solving techniques

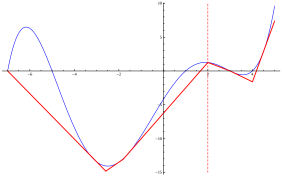
Gradient cuts



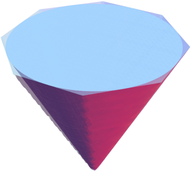
Underestimators



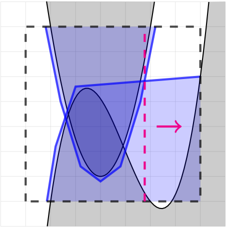
Spatial branching



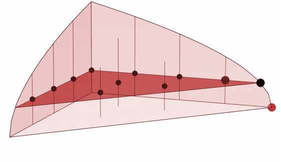
Presolving



Bound tightening



Primal heuristics



Solving MINLPs (with SCIP)

Primal Solutions

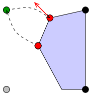
Feasible LP solutions . . .

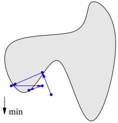
Standard MIP heuristics applied to MIP relaxation

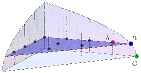
NLP local search

MINLP heuristics

- ▷ nonlinear feasibility pumps [Bonami et al. 2009, D'Ambrosio et al. 2010]
- ▷ RENS [Berthold 2013]
- ▷ Undercover [Berthold and G. 2013]







The Motivation

- Large Neighborhood Search: 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 variables in minimum cover
- solution of LP/NLP relaxation as fixing values

The Structure

Definition Let us be given

- a domain box $[L, U] = \bigtimes_i [L_i, U_i]$,
- a function $g_j : [L, U] \to \mathbb{R}$, $x \mapsto g_j(x)$ on $[L, U]$, and
- a set $\mathcal{C} \subseteq \mathcal{N} := \{1, \ldots, n\}$ of variable indices.

We call $\mathcal{C}$ a cover of g if and only if for all $\bar{x} \in [L, U]$ the set

$$\{(x, g_j(x)) \mid x \in [L, U], x_k = \bar{x}_k \text{ for all } k \in \mathcal{C}\}$$

is an affine set intersected with $[L, U] \times \mathbb{R}$.

We call $\mathcal{C}$ a cover of P if and only if $\mathcal{C}$ is a cover for $g_1, \ldots, g_m$.

Covers of an MINLP

Definition Let P be an MINLP with $g_1, \ldots, g_m$ twice continuously differentiable on the interior of $[L, U]$.

We call $G_P = (V_P, E_P)$ the co-occurrence graph of P with

- node set $V_P = \{1, \ldots, n\}$ and
- edge set $E_P = \left\{ij \mid i, j \in V, \exists k \in \{1, \ldots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \not\equiv 0\right\}$,

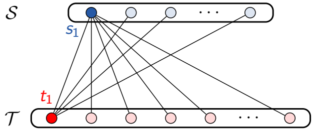
Covers of an MINLP

Definition Let P be an MINLP with $g_1, \dots, g_m$ twice continuously differentiable on the interior of $[L, U]$.

We call $G_P = (V_P, E_P)$ the co-occurrence graph of P with

- node set $V_P = \{1, \dots, n\}$ and
- edge set $E_P = \left\{ij \mid i, j \in V, \exists k \in \{1, \dots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \not\equiv 0\right\}$,

Example



$$\min \quad \dots \quad \text{s.t.} \quad s_1 t_i \leqslant a_i \text{ for all } i = 1, \dots$$
$$s_j t_1 \leqslant b_j \text{ for all } j = 1, \dots$$

Covers of an MINLP

Definition Let P be an MINLP with $g_1, \dots, g_m$ twice continuously differentiable on the interior of $[L, U]$.

We call $G_P = (V_P, E_P)$ the co-occurrence graph of P with

- node set $V_P = \{1, \dots, n\}$ and
- edge set $E_P = \left\{ ij \mid i, j \in V, \exists k \in \{1, \dots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \not\equiv 0 \right\}$,

Theorem [Berthold and G. 2010, 2013]
$\mathcal{C} \subseteq \{1, \dots, n\}$ is a cover of P if and only if it is a vertex cover of the co-occurrence graph G_P.

Covers of an MINLP

Definition Let P be an MINLP with $g_1, \dots, g_m$ twice continuously differentiable on the interior of $[L, U]$.

We call $G_P = (V_P, E_P)$ the co-occurrence graph of P with

- node set $V_P = \{1, \dots, n\}$ and
- edge set $E_P = \left\{ ij \mid i, j \in V, \exists k \in \{1, \dots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \not\equiv 0 \right\}$,

Theorem [Berthold and G. 2010, 2013]
$\mathcal{C} \subseteq \{1, \dots, n\}$ is a cover of P if and only if it is a vertex cover of the co-occurrence graph G_P.

Corollary Computing a minimum cover of an MINLP is $\mathcal{NP}$-hard.

Computing a minimum cover

Auxiliary binary variables

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

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

$$\alpha_k = 1 \qquad \text{for all loops } kk \in E_P, \tag{1}$$

$$\alpha_k + \alpha_j \geqslant 1 \qquad \text{for all edges } kj \in E_p, k > j. \tag{2}$$

⇝ Covering problem

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

Computing a minimum cover

Auxiliary binary variables

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

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

$$\alpha_k = 1 \qquad \text{for all loops } kk \in E_P, \tag{1}$$

$$\alpha_k + \alpha_j \geqslant 1 \qquad \text{for all edges } kj \in E_p, k > j. \tag{2}$$

$\rightsquigarrow$ Covering problem

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

Computing a minimum cover

Auxiliary binary variables

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

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

$$\alpha_k = 1 \qquad \text{for all loops } kk \in E_P, \tag{1}$$

$$\alpha_k + \alpha_j \geqslant 1 \qquad \text{for all edges } kj \in E_p, k > j. \tag{2}$$

$\rightsquigarrow$ Covering problem

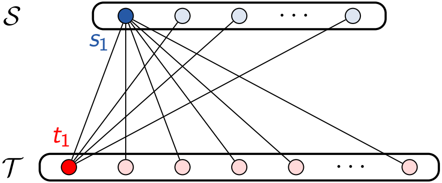
$$\min\left\{\sum_{k=1}^{n} \alpha_k : (1), (2), \alpha \in \{0,1\}^n\right\}. \tag{3}$$

Optimization matters

The co-occurence graph of the bilinear program

$$\min \quad \dots \quad \text{s.t.} \quad s_1 t_i \leqslant a_i \text{ for all } i = 1, \dots,$$
$$s_j t_1 \leqslant b_j \text{ for all } j = 1, \dots,$$

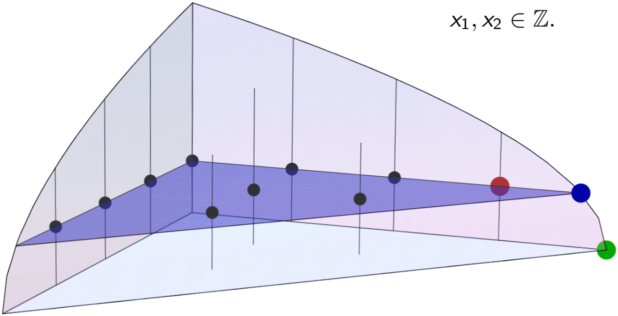
is



The cover $\mathcal{S}$ of complicating variables may be arbitrarily large compared to the minimum cover $\{s_1, t_1\}$.

A simple example

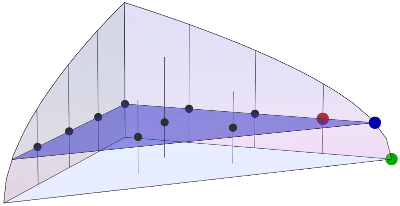
$$\begin{aligned} \max \quad & x_2 + x_3 \\ \text{s.t.} \quad & x_1 + x_2 + x_3^2 \leqslant 4, \\ & x_1, x_2, x_3 \geqslant 0, \\ & x_1, x_2 \in \mathbb{Z}. \end{aligned}$$



Fixing x_3 to any value within its bounds yields a linear subproblem.

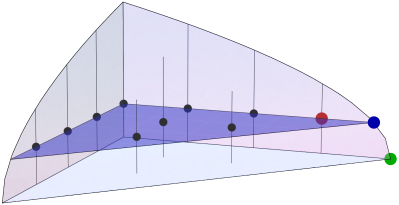
The Undercover Heuristic

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



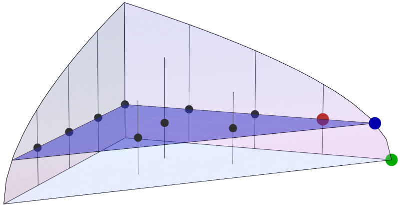
The Undercover Heuristic

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



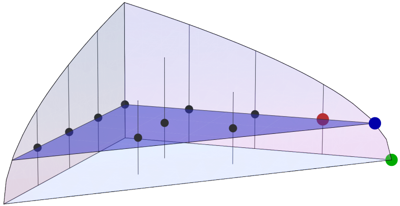
The Undercover Heuristic

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



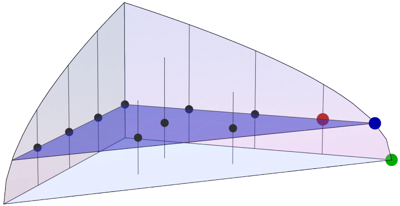
The Undercover Heuristic

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



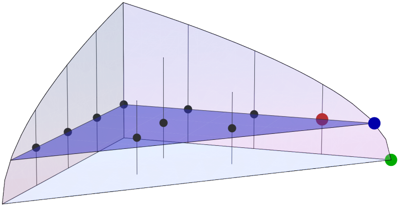
The Undercover Heuristic

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



The Undercover Heuristic

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



Remark:

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

  here: eliminate nonlinearities by fixing as few as possible variables

  $\rightarrow$ minimum cover!

NLP postprocessing

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

Fix-and-propagate & Backtracking

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

Avoiding/exploiting Infeasibility

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.

Computational experiments

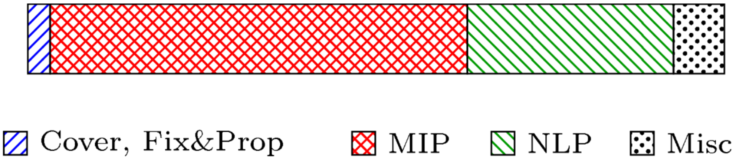
Test set

- 149 MIQCPs from GloMIQO test set

Comparison to other heuristics

- Undercover: solution for 76 instances (typically less than 0.1 sec)
- root heuristics: Baron 65, Couenne 55, SCIP 98
- lower success rate on general MINLPs

Undercover components



Take-away messages

- SCIP can solve nonconvex MINLPs to global optimality
- like other solvers: Antigone/GloMIQO, BARON, Couenne, ...

Take-away messages

- SCIP can solve nonconvex MINLPs to global optimality
- like other solvers: Antigone/GloMIQO, BARON, Couenne, ...

- sometimes problem-specific algorithms can be efficiently generalized to structure-specific algorithms (Undercover)

Take-away messages

- SCIP can solve nonconvex MINLPs to global optimality
- like other solvers: Antigone/GloMIQO, BARON, Couenne, ...

- sometimes problem-specific algorithms can be efficiently generalized to structure-specific algorithms (Undercover)

- convex MINLPs can be solved much more efficiently
- convex modelling/reformulation/detection crucial
- convex solvers can be used heuristically for nonconvex MINLPs

Take-away messages

- SCIP can solve nonconvex MINLPs to global optimality
- like other solvers: Antigone/GloMIQO, BARON, Couenne, ...

- sometimes problem-specific algorithms can be efficiently generalized to structure-specific algorithms (Undercover)

- convex MINLPs can be solved much more efficiently
- convex modelling/reformulation/detection crucial
- convex solvers can be used heuristically for nonconvex MINLPs

Thank you very much for your attention!

Muito obrigado!

Solving Mixed-Integer Nonlinear Programs
(with SCIP)

Ambros M. Gleixner

Zuse Institute Berlin · Matheon · Berlin Mathematical School

5th Porto Meeting on Mathematics for Industry, April 10–11, 2014, Porto