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
Solving MINLPs (with SCIP)

Solving convex MINLPs

Solving nonconvex MINLPs

Modeling, Reformulation, Presolving

Primal Solutions: The Undercover Heuristic
What is Mixed-Integer Nonlinear Programming?

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s. t.} & \quad g_k(x) \leq 0 \\
& \quad x \in [\ell, u], \\
& \quad x_i \in \mathbb{Z} \\
\end{align*}
\]

for \( c \in \mathbb{R}^n \),

for \( k = 1, \ldots, m \), \( g_k : [\ell, u] \to \mathbb{R} \in C^1 \),

for \( i \in I \subseteq \{1, \ldots, n\} \).

\( g_k \) \textbf{ convex} \\
local = \text{global optimality}

\( g_k \) \textbf{ 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

\[
g_k(\hat{x}) + \nabla g_k(\hat{x})^T (x - \hat{x}) \leq 0
\]
bound by polyhedral relaxation
\( \Delta \) at MIP/NLP/sub-NLP solutions
\( \Delta \) at node LP solutions

Many algorithms, many solvers
\( \alpha \)-ECP [Westerlund and Pettersson], BONMIN [Bonami et al.], DICOPT [Duran and Grossmann], sBB [ARKI Software & Consulting], . . .
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

\[
g_k(\hat{x}) + \nabla g_k(\hat{x})^T(x - \hat{x}) \leq 0
\]
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

\[ g_k(\hat{x}) + \nabla g_k(\hat{x})^T(x - \hat{x}) \leq 0 \]

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

\[ g_k(\hat{x}) + \nabla g_k(\hat{x})^T(x - \hat{x}) \leq 0 \]

bound by polyhedral relaxation

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

**Many algorithms, many solvers**
\( \alpha \)-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
- ...

![Various images related to industrial engineering and network design](image-url)
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
- ...

![nonconvex!](image)
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
- ...

nonconvex!
Open Pit Mine Production Scheduling with Stockpiles

Variables:
- $x_{i,t} \in \{0,1\}$ block $i$ fully mined by $t$
- $f_{i,m}^m \in [0,1]$ % of block $i$ mined in $t$
- $f_{i,t}^p \in [0,1]$ % of block $i$ processed in $t$

Constraints:
- material flow conservation
- mining & processing capacities
- mining precedences
Open Pit Mine Production Scheduling with Stockpiles

Stockpile for interim storage
better use of capacities
Difficulty:
stockpile mixes material

$t = 2$

$t = 3$
Aggregated stockpile model

\[ f_{i,t} \in [0,1] \quad \% \text{ of block } i \text{ into stockpiled} \]
\[ Q_{t}^{\text{rock}}, Q_{t}^{\text{met}} \quad \text{total rock / metal tons held} \]
\[ P_{t}^{\text{rock}}, P_{t}^{\text{met}} \quad \text{total rock / metal tons out} \]

Mixing constraints:

\[ \frac{P_{t}^{\text{met}}}{Q_{t}^{\text{met}}} = \frac{P_{t}^{\text{rock}}}{Q_{t}^{\text{rock}}} \quad (\text{metal fraction out} \quad = \text{rock fraction out}) \]
Solving MINLPs (with SCIP)

Solving convex MINLPs

Solving nonconvex MINLPs

Modeling, Reformulation, Presolving

Primal Solutions: The Undercover Heuristic
Nonconvex MINLP

Now some $g_1, \ldots, g_m$ non-convex

Relaxation
gradient cuts invalid

[McCormick 76, Smith and Pantelides 99, Tawarmalani and Sahinidis 02, Belotti et al. 09, Vigerske 13, \ldots ]
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 $g_1, \ldots, g_m$ 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, \ldots]
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, ...]
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, \ldots]
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, ...]
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 $x_1 x_2$

$$(x_1 - l_1) \cdot (x_2 - l_2) \geq 0$$
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 $x_1 x_2$

\[
(x_1 - l_1) \cdot (x_2 - l_2) \geq 0 \\
x_1 x_2 - l_1 x_2 - l_2 x_1 + l_1 l_2 \geq 0
\]
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 $x_1x_2$

$$(x_1 - l_1) \cdot (x_2 - l_2) \geq 0$$

$$x_1x_2 - l_1x_2 - l_2x_1 + l_1l_2 \geq 0$$

$$x_1x_2 \geq l_1x_2 + l_2x_1 - l_1l_2$$
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
Solving MINLPs (with SCIP)

Solving convex MINLPs

Solving nonconvex MINLPs

Modeling, Reformulation, Presolving

Primal Solutions: The Undercover Heuristic
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:

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

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

Prefer Linearity and Convexity

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

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

Prefer Linearity and Convexity

\[
x = y \quad \Rightarrow \quad \text{linear and thus convex}
\]
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:

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

**Prefer Linearity and Convexity**

\[
x y \geq 1 \quad \Rightarrow \quad \text{nonconvex}
\]
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:

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

**Prefer Linearity and Convexity**

\[
y \geq \frac{1}{x} \quad \Rightarrow \text{convex}
\]
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

\[ M^L x \leq w \leq M^U x, \]

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

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 $x^T A x + b^T x \leq c$:

- convex if $A$ is positive-semidefinite
- check by computing its minimal eigenvalue with LAPACK
- if yes: gradient cuts are valid
  \[\Rightarrow\text{ enforcement by separation instead of branching}\]
Convexity check for quadratic constraints

A quadratic constraint $x^T Ax + b^T x \leq c$:

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

**Example** $x^2 + 2xy + y^2 \leq 1$ in $[-1, 1] \times [-1, 1]$
Convexity check for quadratic constraints

A quadratic constraint $x^T Ax + b^T x \leq c$:
- convex if $A$ is positive-semidefinite
- check by computing its minimal eigenvalue with LAPACK
- if yes: gradient cuts are valid
  ⇒ enforcement by separation instead of branching

Example $x^2 + 2xy + y^2 \leq 1 \Leftrightarrow (x + y)^2 \leq 1$ in $[-1, 1] \times [-1, 1]$
Convexity check for quadratic constraints

A quadratic constraint \( x^T A x + b^T x \leq c \):

- 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** \( x^2 + 2xy + y^2 \leq 1 \iff |x + y| \leq 1 \) in \([-1, 1] \times [-1, 1]\)

![feasible region]
Convexity check for quadratic constraints

A quadratic constraint $x^T Ax + b^T x \leq c$:

- 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** $x^2 + 2xy + y^2 \leq 1 \iff (x + y)^2 \leq 1$ in $[-1, 1] \times [-1, 1]$

using McCormick underestimators:

\[
\begin{aligned}
  x^2 + 2w + y^2 &\leq 1 \\
  w &\geq L^y x + L^x y - L^x L^y \\
  w &\geq U^y x + U^x y - U^x U^y
\end{aligned}
\]
Convexity check for quadratic constraints

A quadratic constraint $x^T Ax + b^T x \leq c$:

- 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** $x^2 + 2xy + y^2 \leq 1 \iff (x + y)^2 \leq 1$ in $[-1, 1] \times [-1, 1]$ 

using McCormick underestimators:

\[
\begin{cases}
  x^2 + 2w + y^2 \leq 1 \\
  w \geq L^y x + L^x y - L^x L^y \\
  w \geq U^y x + U^x y - U^x U^y
\end{cases}
\]

branched into 4 subproblems
Convexity check for quadratic constraints

A quadratic constraint $x^T A x + b^T x \leq c$:

- 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** $x^2 + 2xy + y^2 \leq 1 \iff (x + y)^2 \leq 1$ in $[-1, 1] \times [-1, 1]$ using McCormick underestimators:

\[
\begin{cases}
  x^2 + 2w + y^2 \leq 1 \\
  w \geq L^y x + L^x y - L^x L^y \\
  w \geq U^y x + U^x y - U^x U^y
\end{cases}
\]

branched into 16 subproblems
Convexity check for quadratic constraints

A quadratic constraint $x^T A x + b^T x \leq c$:

- 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** $x^2 + 2xy + y^2 \leq 1 \Leftrightarrow (x+y)^2 \leq 1$

in $[-1, 1] \times [-1, 1]$

Using McCormick underestimators:

\[
\begin{cases}
  x^2 + 2w + y^2 \leq 1 \\
  w \geq L^y x + L^x y - L^x L^y \\
  w \geq U^y x + U^x y - U^x U^y
\end{cases}
\]

branched into 64 subproblems
Convexity check for quadratic constraints

A quadratic constraint $x^T A x + b^T x \leq c$:

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

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

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

⇒ gradient cuts at 4 corners yield exact feasible region
Second-order cone upgrade

Quadratic constraints of the form

\[ \sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0 \iff \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1} x_{N+1}} \]

with \( \alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0 \) describe a convex feasible region.

**Example** \( x^2 + y^2 - z^2 \leq 0 \) in \([-1, 1] \times [-1, 1] \times [0, 1]\)

feasible region
“ice cream cone”
Second-order cone upgrade

Quadratic constraints of the form

\[ \sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0 \iff \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1} x_{N+1}} \]

with \( \alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0 \) describe a convex feasible region.

**Example** \( x^2 + y^2 - z^2 \leq 0 \) in \([-1, 1] \times [-1, 1] \times [0, 1]\)

using secant underestimator:

\[
\begin{cases}
  x^2 + y^2 + w \leq 1 \\
  w \geq \frac{(Lz)^2 - (Uz)^2}{Uz - Lz} (z - Lz) - (Lz)^2
\end{cases}
\]
Second-order cone upgrade

Quadratic constraints of the form

\[ \sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0 \iff \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1}} x_{N+1} \]

with \( \alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0 \) describe a convex feasible region.

**Example** \( x^2 + y^2 - z^2 \leq 0 \) in \([-1, 1] \times [-1, 1] \times [0, 1]\)

using secant underestimator:

\[
\begin{aligned}
\left\{ x^2 + y^2 + w \leq 1 \\
w \geq \frac{(L^z)^2 - (U^z)^2}{U^z - L^z} (z - L^z) - (L^z)^2 \right\}
\end{aligned}
\]
Second-order cone upgrade

Quadratic constraints of the form

\[ \sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0 \iff \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1} x_{N+1}} \]

with \( \alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0 \) describe a convex feasible region.

**Example** \( x^2 + y^2 - z^2 \leq 0 \) in \([-1, 1] \times [-1, 1] \times [0, 1]\)

using secant underestimator:

\[ \left\{ \begin{array}{c} x^2 + y^2 + w \leq 1 \\ w \geq \frac{(L^z)^2 - (U^z)^2}{U^z - L^z} (z - L^z) - (L^z)^2 \end{array} \right\} \]

after branching on \( z = 0.25, 0.5, 0.75 \)
Second-order cone upgrade

Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0 \iff \sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1} x_{N+1}}$$

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

**Example** $x^2 + y^2 - z^2 \leq 0$ 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
Outline

Solving MINLPs (with SCIP)

Solving convex MINLPs

Solving nonconvex MINLPs

Modeling, Reformulation, Presolving

Primal Solutions: The Undercover Heuristic
Feasible LP solutions . . .

Standard MIP heuristics applied to MIP relaxation

NLP local search

MINLP heuristics

▷ nonlinear feasibility pumps

▷ RENS [Berthold 2013]

▷ Undercover [Berthold and G. 2013]
The Motivation

- **Large Neighborhood Search**: common paradigm in MIP heuristics
  
  fix a subset of variables $\leadsto$ easy subproblem $\leadsto$ 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 $[L, U] = \times_i [L_i, U_i]$,
- a function $g_j : [L, U] \rightarrow \mathbb{R}$, $x \mapsto g_j(x)$ 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 $E_P = \{ij \mid i, j \in V, \exists k \in \{1, \ldots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \neq 0\}$. 
**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 $E_P = \{ij \mid i, j \in V, \exists k \in \{1, \ldots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \neq 0\}$.

**Example**

\[
\begin{align*}
\min \quad & \ldots \\
\text{s.t.} \quad & s_1 t_i \leq a_i \text{ for all } i = 1, \ldots \\
& s_j t_1 \leq b_j \text{ for all } j = 1, \ldots
\end{align*}
\]
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 $E_P = \{ij \mid i, j \in V, \exists k \in \{1, \ldots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \neq 0\}$.

**Theorem** [Berthold and G. 2010, 2013]

$\mathcal{C} \subseteq \{1, \ldots, n\}$ 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 $g_1, \ldots, g_m$ 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 $E_P = \{ij \mid i, j \in V, \exists k \in \{1, \ldots, m\} : \frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \neq 0\}$.

**Theorem** [Berthold and G. 2010, 2013] $\mathcal{C} \subseteq \{1, \ldots, n\}$ 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

\[ \alpha_k = 1 \iff x_k \text{ is fixed in } P \]

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

\[ \begin{align*}
\alpha_k &= 1 & \text{for all loops } kk \in E_P, \\
\alpha_k + \alpha_j &\geq 1 & \text{for all edges } kj \in E_p, k > j.
\end{align*} \] (1) (2)

\( \rightsquigarrow \) Covering problem

\[ \min \left\{ \sum_{k=1}^{n} \alpha_k : (1), (2), \alpha \in \{0, 1\}^n \right\}. \] (3)
Computing a minimum cover

Auxiliary binary variables

\[ \alpha_k = 1 \iff x_k \text{ is fixed in } P \]

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

\[ \begin{align*}
\alpha_k & = 1 & \text{for all loops } kk \in E_P, \\
\alpha_k + \alpha_j & \geq 1 & \text{for all edges } kj \in E_p, k > j.
\end{align*} \]  

(1) (2)

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

(3)
Computing a minimum cover

Auxiliary binary variables

\[ \alpha_k = 1 : \iff x_k \text{ is fixed in } P \]

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

\[ \alpha_k = 1 \quad \text{for all loops } kk \in E_P, \quad (1) \]

\[ \alpha_k + \alpha_j \geq 1 \quad \text{for all edges } kj \in E_p, \ k > j. \quad (2) \]

\[ \Rightarrow \text{ Covering problem} \]

\[ \min \left\{ \sum_{k=1}^{n} \alpha_k : (1), (2), \alpha \in \{0, 1\}^n \right\}. \quad (3) \]
Optimization matters

The co-occurrence graph of the bilinear program

\[
\begin{align*}
\text{min} & \quad \ldots \\
\text{s.t.} & \quad s_{1} t_{i} \leq a_{i} \text{ for all } i = 1, \ldots, \\
& \quad s_{j} t_{1} \leq b_{j} \text{ for all } j = 1, \ldots,
\end{align*}
\]

is

The cover \( S \) of complicating variables may be \textit{arbitrarily large} compared to the minimum cover \( \{s_{1}, t_{1}\} \).
A simple example

\[
\begin{align*}
\text{max} & \quad x_2 + x_3 \\
\text{s.t.} & \quad x_1 + x_2 + x_3^2 \leq 4, \\
& \quad x_1, x_2, x_3 \geq 0, \\
& \quad x_1, x_2 \in \mathbb{Z}.
\end{align*}
\]

Fixing \(x_3\) to any value within its bounds yields a linear subproblem.
1 **Input**: MINLP $P$
2 **begin**
3 compute a solution $\bar{x}$ of an approximation of $P$;
4 round $\bar{x}_k$ for all $k \in \mathcal{I}$;
5 determine a cover $\mathcal{C}$ of $P$;
6 solve the sub-MIP of $P$ given by fixing $x_k = \bar{x}_k$ for all $k \in \mathcal{C}$;
The Undercover Heuristic

1. **Input**: MINLP $P$
2. begin
3. compute a solution $\bar{x}$ of an approximation of $P$;
4. round $\bar{x}_k$ for all $k \in \mathcal{I}$;
5. determine a cover $\mathcal{C}$ of $P$;
6. 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!
The Undercover Heuristic

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

**Remark:**
- MIP heuristics: trade-off fixing many vs. few variables
  - eliminate nonlinearities by fixing as few as possible variables
  $\rightarrow$ minimum cover!
The Undercover Heuristic

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

**Remark:**
▶ MIP heuristics: trade-off fixing many vs. few variables here: eliminate nonlinearities by fixing as few as possible variables $\rightarrow$ minimum cover!
The Undercover Heuristic

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

**Remark**: MIP heuristics: trade-off fixing many vs. few variables here: eliminate nonlinearities by fixing as few as possible variables $\rightarrow$ minimum cover!
The Undercover Heuristic

1 **Input**: MINLP $P$
2 **begin**
3 \hspace{1em} compute a solution $\bar{x}$ of an approximation of $P$;
4 \hspace{1em} round $\bar{x}_k$ for all $k \in \mathcal{I}$;
5 \hspace{1em} determine a cover $\mathcal{C}$ of $P$;
6 \hspace{1em} 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

- 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 $C$ simultaneously, but sequentially and propagate after each fixing.
- If $x_k^*$ falls out of bounds then
  - fix to the closest bound (similar to [FischettiSalvagnin09])
  - recomputes 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.

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

Cover, Fix&Prop  MIP  NLP  Misc
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)

Thank you very much for your attention!
Muito obrigado!
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!
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