Undercover

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

Ambros M. Gleixner
joint work with Timo Berthold

Zuse Institute Berlin
MATHEON
Berlin Mathematical School
1 Introduction: primal solutions for MINLP
2 A generic algorithm for Undercover
3 Finding minimum covers
   ■ Covering MIQCPs
   ■ General covering problems
4 First experiments with MIQCPs
5 Extensions: fix-and-propagate etc.
6 Variations: convexification & domain reduction
7 Conclusion
An **MINLP** is an optimisation problem of the form

\[
\begin{align*}
\text{minimise} \quad & d^T x \\
\text{subject to} \quad & g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, m, \\
& L_k \leq x_k \leq U_k \quad \text{for } k = 1, \ldots, n, \\
& x_k \in \mathbb{Z} \quad \text{for } k \in \mathcal{I},
\end{align*}
\]

with \( \mathcal{I} \subseteq \{1, \ldots, n\} \), \( d \in \mathbb{R}^n \), \( g_i : \mathbb{R}^n \to \mathbb{R} \), \( L_k \in \mathbb{R} \cup \{-\infty\} \), \( U_k \in \mathbb{R} \cup \{\infty\} \).
Mixed-integer nonlinear programming

An MINLP is an optimisation problem of the form

\[
\begin{align*}
\text{minimise} & \quad d^T x \\
\text{subject to} & \quad g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, m, \\
& \quad L_k \leq x_k \leq U_k \quad \text{for } k = 1, \ldots, n, \\
& \quad x_k \in \mathbb{Z} \quad \text{for } k \in \mathcal{I},
\end{align*}
\]

with \( \mathcal{I} \subseteq \{1, \ldots, n\} \), \( d \in \mathbb{R}^n \), \( g_i : \mathbb{R}^n \rightarrow \mathbb{R} \), \( L_k \in \mathbb{R} \cup \{-\infty\} \), \( U_k \in \mathbb{R} \cup \{\infty\} \).

▷ Special case MIQCP:

\[
\begin{align*}
g_i(x) &= x^T A_i x + b_i^T x + c_i \\
\end{align*}
\]

with \( A_i \in \mathbb{R}^{n \times n} \) symmetric, \( b_i \in \mathbb{R}^n \), \( c_i \in \mathbb{R} \).
An MINLP is an optimisation problem of the form

\[
\begin{align*}
\text{minimise} & \quad d^T x \\
\text{subject to} & \quad g_i(x) \leq 0 \quad \text{for } i = 1, \ldots, m, \\
& \quad L_k \leq x_k \leq U_k \quad \text{for } k = 1, \ldots, n, \\
& \quad x_k \in \mathbb{Z} \quad \text{for } k \in I,
\end{align*}
\]

with \( I \subseteq \{1, \ldots, n\} \), \( d \in \mathbb{R}^n \), \( g_i : \mathbb{R}^n \to \mathbb{R} \), \( L_k \in \mathbb{R} \cup \{-\infty\} \), \( U_k \in \mathbb{R} \cup \{\infty\} \).

\[ \Rightarrow \] Special case MIQCP:

\[
\begin{align*}
g_i(x) &= x^T A_i x + b_i^T x + c_i \\
\end{align*}
\]

with \( A_i \in \mathbb{R}^{n \times n} \) symmetric, \( b_i \in \mathbb{R}^n \), \( c_i \in \mathbb{R} \).

\[ \Rightarrow \] Main classification:

\[
\text{convex} \iff g_i \text{ convex for all } i = 1, \ldots, m
\]

vs. nonconvex MINLPs.
### Primal solutions for generic MINLP

<table>
<thead>
<tr>
<th>Source</th>
<th>convex</th>
<th>nonconvex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasible <strong>relaxation</strong> solution</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>MIP heuristics</strong> for linear outer approximations</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>NLP local search</strong> with fixed integralities</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Simple NLP Rounding</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fractional Diving &amp; Vectorlength Diving</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Iterative Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Feasibility Pump</strong></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td><strong>RECIPE</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>RENS</strong></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

...
## Primal solutions for generic MINLP

<table>
<thead>
<tr>
<th>Source</th>
<th>convex</th>
<th>nonconvex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasible relaxation solution</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MIP heuristics for linear outer approximations</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NLP local search with fixed integralities</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Simple NLP Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fractional Diving &amp; Vectorlength Diving</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiGonçalves08</td>
<td>✓</td>
<td>(✓)</td>
</tr>
<tr>
<td>Iterative Rounding</td>
<td>NanniciniBelotti</td>
<td>✓</td>
</tr>
<tr>
<td>Source</td>
<td>convex</td>
<td>nonconvex</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>Feasible relaxation solution</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>MIP heuristics for linear outer approximations</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>NLP local search with fixed integralities</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Simple NLP Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Fractional Diving &amp; Vectorlength Diving</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiGonçalves08</td>
<td>✓</td>
<td>(✓)</td>
</tr>
<tr>
<td><strong>Iterative Rounding</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Feasibility Pump</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiCornuéjolsLodiMargot08</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>D’AmbrosioFrangioniLibertiLodi09</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LinderothAbhishekLeyfferSartenaer08</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

**D’AmbrosioFrangioniLibertiLodi09** nonconvex obj. convex feas. region
<table>
<thead>
<tr>
<th>Source</th>
<th>convex</th>
<th>nonconvex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasible relaxation solution</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MIP heuristics for linear outer approximations</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NLP local search with fixed integralities</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Simple NLP Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fractional Diving &amp; Vectorlength Diving</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiGonçalves08</td>
<td>✓</td>
<td>(✓)</td>
</tr>
<tr>
<td>Iterative Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Feasibility Pump</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiCornuéjolsLodiMargot08</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>D’AmbrosioFrangioniLibertiLodi09</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LinderothAbhishekLeyfferSartenaer08</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Local Branching</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NanniciniBelottiLiberti08</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Primal solutions for generic MINLP

<table>
<thead>
<tr>
<th>Source</th>
<th>convex</th>
<th>nonconvex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasible relaxation solution</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MIP heuristics for linear outer approximations</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NLP local search with fixed integralities</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Simple NLP Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fractional Diving &amp; Vector length Diving</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiGonçalves08</td>
<td>✓</td>
<td>(✓)</td>
</tr>
<tr>
<td>Iterative Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Feasibility Pump</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiCornuéjolsLodiMargot08</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>D’AmbrosioFrangioniLibertiLodi09</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LinderothAbhishekLeyfferSartenaer08</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Local Branching</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NanniciniBelottiLiberti08</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RECIPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LibertiNanniciniMladenović08</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
### Primal solutions for generic MINLP

<table>
<thead>
<tr>
<th>Source</th>
<th>convex</th>
<th>nonconvex</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feasible relaxation solution</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>MIP heuristics for linear outer approximations</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>NLP local search with fixed integralities</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Simple NLP Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fractional Diving &amp; Vectorlength Diving</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BonamiGonçalves08</td>
<td>✓</td>
<td>(✓)</td>
</tr>
<tr>
<td>Iterative Rounding</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Feasibility Pump</td>
<td>BonamiCornuéjolsLodiMargot08 ✓</td>
<td>✓ nonconvex obj. convex feas. region</td>
</tr>
<tr>
<td>D’AmbrosioFrangioniLibertiLodi09</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>LinderothAbhishekLeyfferSartenaer08</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Local Branching</td>
<td>NanniciniBelottiLiberti08 ✓</td>
<td>✓</td>
</tr>
<tr>
<td>RECIPE</td>
<td>LibertiNanniciniMladenović08 ✓</td>
<td>✓</td>
</tr>
<tr>
<td>RENS</td>
<td>BertholdHeinzVigerske09 (for MIQCPs) ✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
1 Introduction: primal solutions for MINLP
2 A generic algorithm for Undercover
3 Finding minimum covers
   ■ Covering MIQCPs
   ■ General covering problems
4 First experiments with MIQCPs
5 Extensions: fix-and-propagate etc.
6 Variations: convexification & domain reduction
7 Conclusion
Common paradigm in MIP heuristics (e.g. RINS, DINS, RENS):

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

“easy” in MIP context: few integralities

“easy” in MINLP context rather: few nonlinearities
Common paradigm in MIP heuristics (e.g. RINS, DINS, RENS):

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

“easy” in MIP context: few integralities
“easy” in MINLP context rather: few nonlinearities

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

Experience: For several practically relevant MIQCPs comparatively few fixings are sufficient!
Common paradigm in MIP heuristics (e.g. RINS, DINS, RENS):

- fix a subset of variables \( \rightarrow \) easy subproblem \( \rightarrow \) solve

“easy” in MIP context: few integralities

“easy” in MINLP context rather: few nonlinearities

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

Experience: For several practically relevant MIQCPs comparatively few fixings are sufficient!

Idea: try to identify a small subset of variables to fix in order to obtain a mixed-integer linear subproblem.
Definition (cover of a function)

Let

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

We call \( C \) an \( x^* \)-cover of \( g \) if and only if the set

\[
\{(x, g(x)) \mid x \in D, x_k = x_k^* \text{ for all } k \in C\}
\] (4)

is affine.

We call \( C \) a (global) cover of \( g \) if and only if \( C \) is an \( x^* \)-cover of \( g \) for all \( x^* \in D \).
Definition (cover of an MINLP)

Let

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

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

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

1 **Input**: MINLP $P$ as in (1)
2 **begin**
3 compute a solution $x^*$
of an approximation of $P$
4 round $x_k^*$ for all $k \in \mathcal{I}$
5 determine an $x^*$-cover $C$ of $P$
6 solve the sub-MIP of $P$
given by fixing $x_k = x_k^*$
   for all $k \in C$
7 **end**

Remarks:
- As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.
- MIP heuristics need to trade-off between fixing many vs. few (integer) variables: often minimum fixing rate.
- We have to fix nonlinear variables, thus as few as possible to reduce the impact on the MINLP.
- Minimum cover $\not\leftrightarrow$ (dimension-wise) largest sub-MIP
A generic algorithm

1 Input: MINLP $P$ as in (1)
2 begin
3 compute a solution $x^*$ of an approximation of $P$
4 round $x_k^*$ for all $k \in I$
5 determine an $x^*$-cover $C$ of $P$
6 solve the sub-MIP of $P$
given by fixing $x_k = x_k^*$ for all $k \in C$
7 end

Remarks:
1. As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.
2. MIP heuristics need to trade-off between fixing many vs. few (integer) variables: often minimum fixing rate.
3. We have to fix nonlinear variables, thus as few as possible to reduce the impact on the MINLP.
1 **Input**: MINLP \( P \) as in (1)

2 **begin**

3 compute a solution \( x^* \)
of an approximation of \( P \)

4 round \( x_k^* \) for all \( k \in \mathcal{I} \)

5 determine an\n\( x^* \)-cover \( C \) of \( P \)

6 solve the sub-MIP of \( P \)given by fixing \( x_k = x_k^* \)for all \( k \in \mathcal{C} \)

7 **end**

Remarks:
- As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.
- MIP heuristics need to trade-off between fixing many vs. few (integer) variables: often minimum fixing rate.
- We have to fix nonlinear variables, thus as few as possible to reduce the impact on the MINLP.
- Minimum cover \( \not\leftrightarrow \) (dimension-wise) largest sub-MIP.
1 **Input:** MINLP $P$ as in (1)
2 **begin**
3 compute a solution $x^*$ of an approximation of $P$
4 round $x_k^*$ for all $k \in I$
5 **determine an** $x^*$-cover $C$ of $P$
6 solve the sub-MIP of $P$ given by fixing $x_k = x_k^*$ for all $k \in C$
7 **end**
A generic algorithm

1 \textbf{Input}: MINLP $P$ as in (1)

2 \textbf{begin}

3 \quad \text{compute a solution } x^* \\
\quad \text{of an approximation of } P

4 \quad \text{round } x_k^* \text{ for all } k \in \mathcal{I}

5 \quad \text{determine an } \\
\quad x^*\text{-cover } C \text{ of } P

6 \quad \text{solve the sub-MIP of } P \\
\quad \text{given by fixing } x_k = x_k^* \\
\quad \text{for all } k \in C

7 \textbf{end}

\textbf{Remarks:}

1. As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.

2. MIP heuristics need to trade-off between fixing many vs. few (integer) variables: often minimum fixing rate.

3. We have to fix nonlinear variables, thus as few as possible to reduce the impact on the MINLP.

4. Minimum cover $\not\leftrightarrow$ (dimension-wise) largest sub-MIP.
A generic algorithm

1 **Input:** MINLP $P$ as in (1)
2 **begin**
3 compute a solution $x^*$ of an approximation of $P$
4 round $x_k^*$ for all $k \in I$
5 determine an $x^*$-cover $C$ of $P$
6 solve the sub-MIP of $P$
given by fixing $x_k = x_k^*$
7 for all $k \in C$
8 **end**

Remarks:

- As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.
A generic algorithm

1 **Input:** MINLP $P$ as in (1)
2 **begin**
3 compute a solution $x^*$ of an approximation of $P$
4 round $x^*_k$ for all $k \in I$
5 **determine an** $x^*$-cover $C$ of $P$
6 solve the sub-MIP of $P$
   given by fixing $x_k = x^*_k$
   for all $k \in C$
7 **end**

Remarks:

- As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.
- MIP heuristics need to trade-off between fixing many vs. few (integer) variables: often minimum fixing rate.

We have to fix nonlinear variables, thus **as few as possible** to reduce the impact on the MINLP.
A generic algorithm

1 **Input**: MINLP $P$ as in (1)
2 **begin**
3 \hspace{1em} compute a solution $x^\star$ of an approximation of $P$
4 \hspace{1em} round $x_k^\star$ for all $k \in \mathcal{I}$
5 \hspace{1em} determine an $x^\star$-cover $C$ of $P$
6 \hspace{1em} solve the sub-MIP of $P$
7 \hspace{1em} given by fixing $x_k = x_k^\star$
8 \hspace{1em} for all $k \in C$
9 **end**

Remarks:

▷ As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.

▷ MIP heuristics need to trade-off between fixing many vs. few (integer) variables: often minimum fixing rate.

We have to fix nonlinear variables, thus as few as possible to reduce the impact on the MINLP.

▷ Minimum cover $\not\leftrightarrow$ (dimension-wise) largest sub-MIP
1 Introduction: primal solutions for MINLP
2 A generic algorithm for Undercover
3 Finding minimum covers
   - Covering MIQCPs
   - General covering problems
4 First experiments with MIQCPs
5 Extensions: fix-and-propagate etc.
6 Variations: convexification & domain reduction
7 Conclusion
Let \( g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Q x \), \( Q \in \mathbb{R}^{n \times n} \) symmetric, \( x^* \in \mathbb{R}^n \), \( C \subseteq \{1, \ldots, n\} \).

Fixing variables with indices in \( C \) transforms

\[
x^T Q x \quad \overset{x_k = x^*_k \ \forall k \in C}{\implies} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}
\]

with \( y = (x_k)_{k \notin C} \in \mathbb{R}^{n-|C|} \), and \( \tilde{Q} = (Q_{uv})_{u,v \notin C} \in \mathbb{R}^{(n-|C|) \times (n-|C|)} \), \ldots
Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Q x$, $Q \in \mathbb{R}^{n \times n}$ symmetric, $x^* \in \mathbb{R}^n$, $C \subseteq \{1, \ldots, n\}$.

Fixing variables with indices in $C$ transforms

$$x^T Q x \quad \overset{x_k = x^*_k \ \forall k \in C}{\longrightarrow} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

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

Thus: $C$ is a cover of $g$ iff

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

independent of fix. values.
Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^TQx, \ Q \in \mathbb{R}^{n \times n}$ symmetric, $x^* \in \mathbb{R}^n, C \subseteq \{1, \ldots, n\}$.

Fixing variables with indices in $C$ transforms

$$x^TQx \xrightarrow{\sim} x_k^* \quad \forall k \in C$$

$$y^T\tilde{Q}y + \tilde{q}^Ty + \tilde{c}$$

with $y = (x_k)_{k \not\in C} \in \mathbb{R}^{n-|C|}$, and $\tilde{Q} = (Q_{uv})_{u,v \not\in C} \in \mathbb{R}^{(n-|C|) \times (n-|C|)}$, ... 

Thus: $C$ is a cover of $g$ iff

$$q_{uv} = 0 \text{ for all } u, v \not\in C$$

independent of fix. values.

**set covering:**

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

cover nonzeros in $Q$ by incident rows/columns
Let \( g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Q x \), \( Q \in \mathbb{R}^{n \times n} \) symmetric, \( x^* \in \mathbb{R}^n \), \( C \subseteq \{1, \ldots, n\} \).

Fixing variables with indices in \( C \) transforms

\[
x^T Q x \quad \xrightarrow{\text{fix.}} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}
\]

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

Thus: \( C \) is a cover of \( g \) iff

\[
q_{uv} = 0 \quad \text{for all} \quad u, v \notin C
\]

independent of fix. values.

set covering: \( \begin{pmatrix} * \\ * \end{pmatrix} \)

cover nonzeros in \( Q \) by incident rows/columns
Let $g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Q x$, $Q \in \mathbb{R}^{n \times n}$ symmetric, $x^* \in \mathbb{R}^n$, $C \subseteq \{1, \ldots, n\}$.

Fixing variables with indices in $C$ transforms

$$x^T Q x \underset{x_k = x^*_k \ \forall k \in C}{\sim} y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

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

Thus: $C$ is a cover of $g$ iff

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

independent of fix. values.
Let \( g : \mathbb{R}^n \rightarrow \mathbb{R}, x \mapsto x^T Q x, \ Q \in \mathbb{R}^{n \times n} \text{ symmetric}, \ x^* \in \mathbb{R}^n, \ C \subseteq \{1, \ldots, n\}. \)

Fixing variables with indices in \( C \) transforms

\[
x^T Q x \quad \mapsto \quad x_k = x^*_k \ \forall k \in C \quad \mapsto \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}
\]

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

Thus: \( C \) is a cover of \( g \) iff \( q_{uv} = 0 \) for all \( u, v \notin C \) independent of fix. values.

\( \iff \) \hspace{1cm} \underline{set covering:}

\[
\begin{pmatrix}
* \\
* \\
* \\
* \\
* \\
* \\
* \\
* \\
* \\
* \\
\end{pmatrix}
\]

cover nonzeros in \( Q \) by incident rows/columns
For an MIQCP $P$, introduce one auxiliary binary variables

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

for each original variable $x_k$, $k = 1, \ldots, n$.

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

$$\alpha_k = 1 \text{ f.a. } i \in \{1, \ldots, m\}, k \in \{1, \ldots, n\}, A^i_{kk} \neq 0, L_k \neq U_k, \quad (5)$$

$$\alpha_k + \alpha_j \geq 1 \text{ f.a. } i \in \{1, \ldots, m\}, k \neq j \in \{1, \ldots, n\}, A^i_{kj} \neq 0,$$

$$L_k \neq U_k, L_j \neq U_j. \quad (6)$$

To find a minimum cover, we solve the covering problem

$$\min \left\{ \sum_{k=1}^{n} \alpha_k : (5), (6), \alpha \in \{0, 1\}^n \right\}. \quad (7)$$
For an MIQCP $P$, introduce one auxiliary binary variables

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

for each original variable $x_k$, $k = 1, \ldots, n$.

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

$$\alpha_k = 1 \quad \text{f.a. } i \in \{1, \ldots, m\}, k \in \{1, \ldots, n\}, A_{ik} \neq 0, L_k \neq U_k, \quad (5)$$

$$\alpha_k + \alpha_j \geq 1 \quad \text{f.a. } i \in \{1, \ldots, m\}, k \neq j \in \{1, \ldots, n\}, A_{kj} \neq 0,$$

$$L_k \neq U_k, L_j \neq U_j. \quad (6)$$

To find a minimum cover, we solve the covering problem

$$\min \left\{ \sum_{k=1}^{n} \alpha_k : (5), (6), \alpha \in \{0, 1\}^n \right\}. \quad (7)$$
Covering MIQCPs

For an MIQCP $P$, introduce one auxiliary binary variables

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

for each original variable $x_k$, $k = 1, \ldots, n$.

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

$$\alpha_k = 1 \text{ f.a. } i \in \{1, \ldots, m\}, k \in \{1, \ldots, n\}, A^i_{kk} \neq 0, L_k \neq U_k, \quad (5)$$

$$\alpha_k + \alpha_j \geq 1 \text{ f.a. } i \in \{1, \ldots, m\}, k \neq j \in \{1, \ldots, n\}, A^i_{kj} \neq 0,$$
$$L_k \neq U_k, L_j \neq U_j. \quad (6)$$

To find a minimum cover, we solve the covering problem

$$\min \left\{ \sum_{k=1}^{n} \alpha_k : (5), (6), \alpha \in \{0, 1\}^n \right\}. \quad (7)$$
(7) is an optimisation version of 2-SAT, hence \textit{polynomial-time solvable.} Though the feasible region of (7) is not integral, also standard branch-and-cut is (empirically) fast.

For general MINLPs, the covering problem becomes more difficult, e.g. the conditions for a global cover of a monomial \( x^{p_1} \cdots x^{p_n} \), \( p_1, \ldots, p_n \in \mathbb{N}_0 \), are

\[ \alpha_k = 1 \text{ f.a. } k \in \{ 1, \ldots, n \}, \\
L_k \neq U_k, \\
\sum_{k: p_k = 1, L_k \neq U_k} (1 - \alpha_k) \leq 1. \] (8) (9)

For general MINLPs, global covers become larger and larger. However: \( x^* \)-covers are now a weaker notion and may be significantly smaller, e.g. due to “0-fixings.”
(7) is an optimisation version of 2-SAT, hence polynomial-time solvable. Though the feasible region of (7) is not integral, also standard branch-and-cut is (empirically) fast.

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

\[
\alpha_k = 1 \quad \text{f.a. } k \in \{1, \ldots, n\}, \ p_k \geq 2, \ L_k \neq U_k, \quad (8)
\]

\[
\sum_{k: p_k=1, L_k \neq U_k} (1 - \alpha_k) \leq 1. \quad (9)
\]
(7) is an optimisation version of 2-SAT, hence polynomial-time solvable. Though the feasible region of (7) is not integral, also standard branch-and-cut is (empirically) fast.

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

$$\alpha_k = 1 \quad \text{f.a. } k \in \{1, \ldots, n\}, p_k \geq 2, L_k \neq U_k, \quad (8)$$

$$\sum_{k : p_k = 1, L_k \neq U_k} (1 - \alpha_k) \leq 1. \quad (9)$$

For general MINLPs, global covers become larger and larger. However: $x^*$-covers are now a weaker notion and may be significantly smaller, e.g. due to “0-fixings”.
1. Introduction: primal solutions for MINLP
2. A generic algorithm for Undercover
3. Finding minimum covers
   - Covering MIQCPs
   - General covering problems
4. First experiments with MIQCPs
5. Extensions: fix-and-propagate etc.
6. Variations: convexification & domain reduction
7. Conclusion
The constraint integer programming solver SCIP has recently been extended to handle nonconvex MIQCPs [BertholdHeinzVigerske09]:

- LP-based (safe) outer approximation
  - gradient cuts for convex terms
  - McCormick for bilinear terms
  - secant underestimators for concave univariate terms

- primal solutions via
  - feasible relaxation and MIP heuristics
  - NLP local search with fixed integralities
  - extended RENS heuristic

- Undercover as MIQCP start heuristic
  - set covering problem and sub-MIP solved by secondary SCIP instance
  - fixing values from outer approximation

- implemented features: fix-and-propagate, backtracking, NLP postprocessing (later)
The constraint integer programming solver SCIP has recently been extended to handle nonconvex MIQCPs [BertholdHeinzVigerske09]:

- **LP-based (safe) outer approximation**
  - gradient cuts for convex terms
  - McCormick for bilinear terms
  - secant underestimators for concave univariate terms

- **Primal solutions via**
  - feasible relaxation and MIP heuristics
  - NLP local search with fixed integralities
  - extended RENS heuristic
The constraint integer programming solver **SCIP** has recently been extended to handle nonconvex MIQCPs [BertholdHeinzVigerske09]:

- **LP-based (safe) outer approximation**
  - gradient cuts for convex terms
  - McCormick for bilinear terms
  - secant underestimators for concave univariate terms

- **Primal solutions via**
  - feasible relaxation and MIP heuristics
  - NLP local search with fixed integralities
  - extended RENS heuristic

- **Undercover as MIQCP start heuristic**
  - set covering problem and sub-MIP solved by secondary SCIP instance
  - fixing values from outer approximation
  - implemented features: fix-and-propagate, backtracking, NLP postprocessing (later)
First experiments with MIQCPs from MINLPLib

- **Goal**: evaluate potential as start heuristic at the root node

- **Test set**: 33 MIQCP instances from MINLPLib
  - excluded instances which are linear after presolve
  - selected only two **nuclear** instances (often unbounded root LP in SCIP)

- **Undercover parameters**
  - running as only heuristic at the root node in SCIP 1.2.0.4 with CPLEX 12.1 and Ipopt 3.7 (ma27)
  - for sub-MIP: emphasis feasibility and fast presolving settings, node limit 500

- **Reference solvers**
  - SCIP 1.2.0.4 with CPLEX 12.1 and Ipopt 3.7 (incl. nonlinear RENS)
  - BARON 9.02 with CPLEX 12.1 and MINOS 5.51
  - Couenne 0.2 with Clp 1.10 and Ipopt 3.7 (ma27)
    - settings: default, node limit 1, no time limit

- Reported: nonlinear nonzeros/variable, % variables fixed by Undercover, solution values of Undercover (**∗**: sub-MIP optimal) and each solver
First experiments with MIQCPs from MINLPLib

- **Goal:** evaluate potential as start heuristic at the root node

- **Test set:** 33 MIQCP instances from MINLPLib
  - excluded instances which are linear after presolve
  - selected only two nuclear instances (often unbounded root LP in SCIP)

- **Undercover parameters**
  - running as only heuristic at the root node in SCIP 1.2.0.4 with CPLEX 12.1 and Ipopt 3.7 (ma27)
  - for sub-MIP: emphasis feasibility and fast presolving settings, node limit 500
First experiments with MIQCPs from MINLPLib

- **Goal:** evaluate potential as start heuristic at the root node

- **Test set:** 33 MIQCP instances from MINLPLib
  - excluded instances which are linear after presolve
  - selected only two nuclear instances (often unbounded root LP in SCIP)

- **Undercover parameters**
  - running as only heuristic at the root node in SCIP 1.2.0.4 with CPLEX 12.1 and Ipopt 3.7 (ma27)
  - for sub-MIP: emphasis feasibility and fast presolving settings, node limit 500

- **Reference solvers**
  - SCIP 1.2.0.4 with CPLEX 12.1 and Ipopt 3.7 (ma27) (incl. nonlinear RENS)
  - BARON 9.02 with CPLEX 12.1 and MINOS 5.51
  - Couenne 0.2 with Clp 1.10 and Ipopt 3.7 (ma27)
  - settings: default, node limit 1, no time limit

- **Reported:** nonlinear nonzeros/variable, % variables fixed by Undercover, solution values of Undercover (*: sub-MIP optimal) and each solver
### Computational results for MIQCPs

12 instances with \( \leq 5\% \) variables fixed

<table>
<thead>
<tr>
<th>instance</th>
<th>nnz/var</th>
<th>% cov</th>
<th>UC</th>
<th>SCIP</th>
<th>BARON</th>
<th>Couenne</th>
</tr>
</thead>
<tbody>
<tr>
<td>netmod_dol1</td>
<td>0.00</td>
<td>0.30</td>
<td>0*</td>
<td>-0.31730</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>netmod_dol2</td>
<td>0.00</td>
<td>0.38</td>
<td>-0.078023*</td>
<td>-0.50468</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>netmod_kar1</td>
<td>0.01</td>
<td>0.88</td>
<td>0*</td>
<td>-0.13281</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>netmod_kar2</td>
<td>0.01</td>
<td>0.88</td>
<td>0*</td>
<td>-0.13281</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>space25</td>
<td>0.12</td>
<td>1.04</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ex1266</td>
<td>0.40</td>
<td>3.03</td>
<td>16.3*</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>util</td>
<td>0.07</td>
<td>3.13</td>
<td>999.58*</td>
<td>1000.5</td>
<td>1006.5</td>
<td>–</td>
</tr>
<tr>
<td>feedtray2</td>
<td>10.70</td>
<td>3.26</td>
<td>–</td>
<td>0</td>
<td>0</td>
<td>–</td>
</tr>
<tr>
<td>ex1265</td>
<td>0.38</td>
<td>3.52</td>
<td>15.1*</td>
<td>–</td>
<td>–</td>
<td>15.1</td>
</tr>
<tr>
<td>ex1263</td>
<td>0.34</td>
<td>3.88</td>
<td>30.1*</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>tln12</td>
<td>1.70</td>
<td>3.99</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>ex1264</td>
<td>0.36</td>
<td>4.26</td>
<td>15.1*</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

- 9 instances feasible, 5 times best solution value
- ex1266 optimal, util 0.1% gap
## Computational results for MIQCPs

### 10 instances with 5–15% variables fixed

<table>
<thead>
<tr>
<th>instance</th>
<th>nnz/var</th>
<th>% cov</th>
<th>UC</th>
<th>SCIP</th>
<th>BARON</th>
<th>Couenne</th>
</tr>
</thead>
<tbody>
<tr>
<td>waste</td>
<td>1.10</td>
<td>5.65</td>
<td>693.39</td>
<td>693.29</td>
<td>712.301</td>
<td>–</td>
</tr>
<tr>
<td>space25a</td>
<td>0.29</td>
<td>5.84</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>nuclear14a</td>
<td>4.98</td>
<td>6.43</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>nuclear14b</td>
<td>2.42</td>
<td>6.43</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>-1.1105</td>
</tr>
<tr>
<td>tln7</td>
<td>1.53</td>
<td>6.67</td>
<td>30.3*</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>tln6</td>
<td>1.47</td>
<td>7.69</td>
<td>32.3*</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>tloss</td>
<td>1.47</td>
<td>7.89</td>
<td>27.3*</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>tln5</td>
<td>1.39</td>
<td>9.09</td>
<td>15.1*</td>
<td>–</td>
<td>–</td>
<td>14.5</td>
</tr>
<tr>
<td>sep1</td>
<td>0.40</td>
<td>10.53</td>
<td>-510.08*</td>
<td>–</td>
<td>-510.08</td>
<td>-510.08</td>
</tr>
<tr>
<td>tltr</td>
<td>1.10</td>
<td>12.50</td>
<td>61.133*</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

> 7 instances feasible, 5 times best solution value
### Computational results for MIQCPs

#### 11 instances with 15–96% variables fixed

<table>
<thead>
<tr>
<th>instance</th>
<th>nnz/var</th>
<th>% cov</th>
<th>UC</th>
<th>SCIP</th>
<th>BARON</th>
<th>Couenne</th>
</tr>
</thead>
<tbody>
<tr>
<td>nous1</td>
<td>2.39</td>
<td>19.44</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>1.5671</td>
</tr>
<tr>
<td>nous2</td>
<td>2.39</td>
<td>19.44</td>
<td>–</td>
<td>1.3843</td>
<td>0.62597</td>
<td>1.3843</td>
</tr>
<tr>
<td>meanvarx</td>
<td>0.19</td>
<td>23.33</td>
<td>16.997*</td>
<td>14.369</td>
<td>14.369</td>
<td>18.702</td>
</tr>
<tr>
<td>product2</td>
<td>0.37</td>
<td>26.15</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>product</td>
<td>0.17</td>
<td>30.87</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>spectra2</td>
<td>3.43</td>
<td>35.71</td>
<td>26.608*</td>
<td>23.284</td>
<td>119.87</td>
<td>–</td>
</tr>
<tr>
<td>fac3</td>
<td>0.81</td>
<td>78.26</td>
<td>13065e4*</td>
<td>–</td>
<td>38329e3</td>
<td>–</td>
</tr>
<tr>
<td>nvs19</td>
<td>8.00</td>
<td>88.89</td>
<td>–</td>
<td>0</td>
<td>-1098</td>
<td>–</td>
</tr>
<tr>
<td>nvs23</td>
<td>9.00</td>
<td>90.00</td>
<td>–</td>
<td>0</td>
<td>-1124.8</td>
<td>–</td>
</tr>
<tr>
<td>du-opt5</td>
<td>0.95</td>
<td>94.74</td>
<td>3407.1*</td>
<td>14.168</td>
<td>–</td>
<td>1226.0</td>
</tr>
<tr>
<td>du-opt</td>
<td>0.95</td>
<td>95.24</td>
<td>4233.9*</td>
<td>4233.9</td>
<td>108.33</td>
<td>41.304</td>
</tr>
</tbody>
</table>

▶ 5 instances feasible, no best solution value
Feasible solutions
- Undercover: 21 instances
- SCIP: 14 instances
- BARON: 15 instances
- Couenne: 9 instances
- All: 27 instances

Solution quality: comparison on instances where Undercover and a solver both found a solution
- Undercover : SCIP = 1:8 (1 equal)
- Undercover : BARON = 4:3 (4 equal)
- Undercover : Couenne = 1:3 (2 equal)
Computational results for MIQCPs

▶ **Feasible solutions**
  - Undercover: 21 instances
  - SCIP: 14 instances
  - BARON: 15 instances
  - Couenne: 9 instances
  - All: 27 instances

▶ **Solution quality**: comparison on instances where Undercover and a solver both found a solution
  - Undercover : SCIP = 1:8 (1 equal)
  - Undercover : BARON = 4:3 (4 equal)
  - Undercover : Couenne = 1:3 (2 equal)

▶ **SCIP time** (presolve, outer approx., LP, Undercover) always < 2 seconds

▶ **Undercover time** always < 0.2 seconds (except for waste with 1.1 sec)
  - set covering always solved to optimality at root
  - most time spent in sub-MIP
  - infeasibility of sub-MIP usually detected fast during fix-and-propagate (in 10 out of 12 infeasible cases)
  - 20 of 21 feasible sub-MIPs solved to optimality
1. Introduction: primal solutions for MINLP
2. A generic algorithm for Undercover
3. Finding minimum covers
   - Covering MIQCPs
   - General covering problems
4. First experiments with MIQCPs
5. Extensions: fix-and-propagate etc.
6. Variations: convexification & domain reduction
7. Conclusion
Fix-and-propagate

- Do not fix the variables in the cover $C$ simultaneously to $x^*$-values, but **sequentially** and propagate the bound changes after each fixing.

- If by that, some fixing value $x_k^*$ falls out of its propagated bounds then
  - fix to the closest bound (similar to FischettiSalvagnin09)
  - alternatively recompute the approximation
Fix-and-propagate

- Do not fix the variables in the cover $C$ simultaneously to $x^*$-values, but sequentially and propagate the bound changes after each fixing.

- If by that, some fixing value $x^*_k$ falls out of its propagated bounds then
  - fix to the closest bound (similar to FischettiSalvagnin09)
  - alternatively recompute the approximation

Backtracking

If fix-and-propagate deduces infeasibility, apply a one-level backtracking: undo the last fixing and try other values instead (bounds, zero, etc.).
Fix-and-propagate & Backtracking

Fix-and-propagate

▷ Do not fix the variables in the cover $C$ simultaneously to $x^*$-values, but sequentially and propagate the bound changes after each fixing.

▷ If by that, some fixing value $x^*_k$ falls out of its propagated bounds then
  ▷ fix to the closest bound (similar to FischettiSalvagnin09)
  ▷ alternatively recompute the approximation

Backtracking

▷ If fix-and-propagate deduces infeasibility, apply a one-level backtracking: undo the last fixing and try other values instead (bounds, zero, etc.).
Recovering

- During fix-and-propagate, variables outside of the precomputed cover $C$ may also be fixed.

- In this case, yet unfixed variables in $C$ might not have to be fixed anymore.

$\hookrightarrow$ “re-cover”: solve the covering problem again considering all bound changes from fix-and-propagate.
Using different covers

Covers minimising different impact measures

- Motivation for minimum cardinality covers: minimise impact on MINLP

- Alternative impact measures can be used in the objective function of the 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
All sub-MIP solutions are fully feasible for the original MINLP.

Still, the best found sub-MIP solution $\tilde{x}$ can possibly be improved by NLP local search:

- fix all integer variables of the original MINLP to their values in $\tilde{x}$
- solve the resulting (possibly nonconvex) NLP to local optimality
1 Introduction: primal solutions for MINLP
2 A generic algorithm for Undercover
3 Finding minimum covers
   ■ Covering MIQCPs
   ■ General covering problems
4 First experiments with MIQCPs
5 Extensions: fix-and-propagate etc.
6 Variations: convexification & domain reduction
7 Conclusion
Idea of Undercover: identify few variables to fix in order to obtain an “easy” subproblem. Possible by
- switching to an easier problem class
- switching to an easier problem of the same class (MINLP)
Variations: convexification & domain reduction

- **Idea of Undercover:** identify few variables to fix in order to obtain an “easy” subproblem. Possible by
  - switching to an easier problem class
  - switching to an easier problem of the same class (MINLP)

- **Switching to an easier problem class:**
  - MINLP $\leadsto$ MIP (so far, in general very restrictive)
  - MINLP $\leadsto$ MIQCP
  - nonconvex MINLP $\leadsto$ convex MINLP
  - ...
Variations: convexification & domain reduction

- Idea of Undercover: identify few variables to fix in order to obtain an "easy" subproblem. Possible by
  - switching to an easier problem class
  - switching to an easier problem of the same class (MINLP)

- Switching to an easier problem class:
  - MINLP $\leadsto$ MIP (so far, in general very restrictive)
  - MINLP $\leadsto$ MIQCP
  - nonconvex MINLP $\leadsto$ convex MINLP
  - ...

- Switching to an easier problem of the same class: restrict domains of variables in the cover
  - can yield significantly better outer approximations
  - while leaving more freedom to the problem
An example: soft rectangle packing

Given

- a fixed number \( n \) of rectangles
- with fixed areas \( A_1, \ldots, A_n \)
- and bounded widths and heights,

arrange them without gap and overlap to form a large rectangle:

Application: sheet metal design [FügenschuhHessScheweMartinUlbrich08]
An example: soft rectangle packing

Given

▷ a fixed number $n$ of rectangles
▷ with fixed areas $A_1, \ldots, A_n$
▷ and bounded widths and heights,

arrange them without gap and overlap to form a large rectangle:

\[
\text{minimise} \quad W + H + \sum_i w_i + \sum_i h_i
\]
\[
\text{subject to} \quad \text{linear/combinatorial constraints,}
\]
\[
w_i \cdot h_i = A_i \quad \text{for } i = 1, \ldots, n,
\]
\[
W \cdot H = \sum_i A_i,
\]

bounded widths and heights.

Application: sheet metal design [FügenschuhHessScheweMartinUlbrich08]
An example: soft rectangle packing

Given

- a fixed number $n$ of rectangles
- with fixed areas $A_1, \ldots, A_n$
- and bounded widths and heights,

arrange them without gap and overlap to form a large rectangle:

minimise $W + H + \sum_i w_i + \sum_i h_i$

subject to linear/combinatorial constraints,

bilnear $\quad w_i h_i = A_i$ for $i = 1, \ldots, n,$

bilnear $\quad WH = \sum_i A_i,$

bounded widths and heights.

Application: sheet metal design [FügenschuhHessScheweMartinUlbrich08]
An example: soft rectangle packing

Given

- a fixed number $n$ of rectangles
- with fixed areas $A_1, \ldots, A_n$
- and bounded widths and heights,

arrange them without gap and overlap to form a large rectangle:

minimise $W + H + \sum_i w_i + \sum_i h_i$

subject to linear/combinatorial constraints,

univariate nonlinear $w_i = A_i/h_i$ for $i = 1, \ldots, n$,

univariate nonlinear $W = (\sum_i A_i)/H$,

bounded widths and heights.

Application: sheet metal design [FügenschuhHessScheweMartinUlbrich08]
An example: soft rectangle packing

Given

- a fixed number \( n \) of rectangles
- with fixed areas \( A_1, \ldots, A_n \)
- and bounded widths and heights,

arrange them without gap and overlap to form a large rectangle:

\[
\begin{align*}
\text{minimise} & \quad W + H + \sum_i w_i + \sum_i h_i \\
\text{subject to} & \quad \text{linear/combinatorial constraints,} \\
& \quad \text{convex nonlinear} \quad w_i \geq A_i/h_i \quad \text{for } i = 1, \ldots, n, \\
& \quad \text{nonconvex nonlinear} \quad W \leq (\sum_i A_i)/H,
\end{align*}
\]

bounded widths and heights.

Application: sheet metal design [FügenschuhHessScheweMartinUlbrich08]
Computational results for soft rectangle packing

- 25 test instances from [FügenschuhHessScheweMartinUlbrich08]
- SCIP with and without “convex” Undercover at root and in the tree:
  Undercover fixes $W$ or $H$ at the current node $\rightsquigarrow$ convex sub-MINLP

<table>
<thead>
<tr>
<th>$A_1, \ldots, A_n$</th>
<th>time to opt. [s]</th>
<th>$A_1, \ldots, A_n$</th>
<th>time to opt. [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SCIP</td>
<td>SCIP&amp;UC</td>
<td>SCIP</td>
</tr>
<tr>
<td>1,3,6</td>
<td>0.98</td>
<td>0.26</td>
<td>1,5,6,8</td>
</tr>
<tr>
<td>2,3,4</td>
<td>0.58</td>
<td>0.15</td>
<td>2,3,4,5</td>
</tr>
<tr>
<td>2,5,8</td>
<td>0.58</td>
<td>0.27</td>
<td>2,3,5,6</td>
</tr>
<tr>
<td>2,7,8</td>
<td>1.72</td>
<td>0.37</td>
<td>2,8,9,10</td>
</tr>
<tr>
<td>3,5,6</td>
<td>0.46</td>
<td>0.16</td>
<td>4,4,5,7</td>
</tr>
<tr>
<td>3,8,13</td>
<td>0.60</td>
<td>0.33</td>
<td>4,5,5,6</td>
</tr>
<tr>
<td>4,5,7</td>
<td>1.17</td>
<td>0.15</td>
<td>1,2,3,4,5</td>
</tr>
<tr>
<td>5,7,11</td>
<td>1.33</td>
<td>0.17</td>
<td>1,2,3,8,9</td>
</tr>
<tr>
<td>7,8,9</td>
<td>0.92</td>
<td>0.22</td>
<td>1,3,4,5,5</td>
</tr>
<tr>
<td>7,9,12</td>
<td>0.36</td>
<td>0.20</td>
<td>3,4,5,6,7</td>
</tr>
<tr>
<td>1,2,3,4</td>
<td>3.61</td>
<td>2.81</td>
<td>4,4,4,5,6</td>
</tr>
<tr>
<td>1,2,9,10</td>
<td>5.45</td>
<td>2.20</td>
<td>1,2,3,4,5,6,7</td>
</tr>
<tr>
<td>1,3,5,12</td>
<td>2.70</td>
<td>2.58</td>
<td>1,2,3,4,5,6,7</td>
</tr>
</tbody>
</table>
1. Introduction: primal solutions for MINLP
2. A generic algorithm for Undercover
3. Finding minimum covers
   - Covering MIQCPs
   - General covering problems
4. First experiments with MIQCPs
5. Extensions: fix-and-propagate etc.
6. Variations: convexification & domain reduction
7. Conclusion
Scheme of a general-purpose start heuristic for MINLP

- solve a set covering/satisfiability problem
- to identify few variable fixings
- yielding a mixed-integer linear subproblem

Preliminary experiments

- MIQCPs from MINLPLib – often few fixings sufficient:
  - \( \leq 5\% \) on 1/3 of the test set, \( \leq 15\% \) on 2/3 of the test set
- soft rectangle packing

Future research

- extensions and variations
- experiments on general MINLPs
- tuning for efficient use within branch-and-bound tree
Pietro Belotti.
*Couenne: a user's manual.*

Timo Berthold and Ambros M. Gleixner.
Undercover – a primal heuristic for MINLP based on sub-MIPs generated by set covering.

Timo Berthold, Stefan Heinz, and Stefan Vigerske.
Extending a CIP framework to solve MIQCPs.

Pierre Bonami, Gérard Cornuéjols, Andrea Lodi, and François Margot.
A feasibility pump for mixed integer nonlinear programs.

Pierre Bonami and João P.M. Gonçalves.
Primal heuristics for mixed integer nonlinear programs.

Michael R. Bussieck, Arne S. Drud, and Alexander Meeraus.
Claudia D’Ambrosio, Antonio Frangioni, Leo Liberti, and Andrea Lodi.  
A feasibility pump heuristic for non-convex MINLPs. 
CIMINLP, Bordeaux, March 2009.

Matteo Fischetti and Domenico Salvagnin.  
Feasibility pump 2.0.  

Verfeinerte Modelle zur Topologie- und Geometrie-Optimierung von Blechprofilen mit Kammern.  

Leo Liberti, Giacomo Nannicini, and Nenad Mladenović.  
A good recipe for solving MINLPs.  

Jeff T. Linderoth, Kumar Abhishek, Sven Leyffer, and Annick Sartenaer.  
Feasibility pump heuristics for MINLP.  
Giacomo Nannicini and Pietro Belotti.
Rounding-based heuristics for nonconvex MINLPs.
Submitted.

Giacomo Nannicini, Pietro Belotti, and Leo Liberti.
A local branching heuristic for MINLPs.

Nikolaos V. Sahinidis and Mohit Tawarmalani.
Undercover

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

Ambros M. Gleixner
joint work with Timo Berthold

Zuse Institute Berlin
MATHEON
Berlin Mathematical School

Aussois International Workshop on Combinatorial Optimization, 6 January 2010