Undercover

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

Ambros M. Gleixner
Zuse Institute Berlin • MATHEON • Berlin Mathematical School

joint work with Timo Berthold

European Workshop on Mixed-Integer Nonlinear Programming, 16 April 2010
1 Introduction: MINLP, MIQCP, MIBCP
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

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

with \( \mathcal{I} \subseteq \{1, \ldots, n\} \), \( d \in \mathbb{R}^n \), \( g_i : \mathbb{R}^n \to \mathbb{R} \), \( L_k \in \mathbb{R} \cup \{-\infty\} \), \( U_k \in \mathbb{R} \cup \{\infty\} \).
An **MINLP** is an optimisation problem of the form

\[
\begin{align*}
\text{minimise} & \quad d^T x \\
\text{subject to} & \quad g_i(x) \leq 0 \quad \text{for} \ i = 1, \ldots, m,
\end{align*}
\]

\[
L_k \leq x_k \leq U_k \quad \text{for} \ k = 1, \ldots, n,
\]

\[
x_k \in \mathbb{Z} \quad \text{for} \ k \in \mathcal{I},
\]

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

**Special case MIQCP:**

\[
g_i(x) = x^T A_i x + b_i^T x + c_i
\]

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

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

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:**

\[
g_i(x) = x^T A_i x + b_i^T x + c_i
\]

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

**Mixed-integer bilinearly constrained programmes:**

\[
x = (x_1, x_2) \quad \text{and} \quad g_i(x) = x_1^T A_i x_2 + b_i^T x + c_i
\]

with \( A_i \in \mathbb{R}^{n_1 \times n_2} \), \( b_i \in \mathbb{R}^n \), \( c_i \in \mathbb{R} \).
1. Introduction: MINLP, MIQCP, MIBCP
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 $\leadsto$ 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!

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 $g : D \to \mathbb{R}$, $x \mapsto g(x)$ 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

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

is affine.

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

<table>
<thead>
<tr>
<th>Definitions</th>
<th>6 / 34</th>
</tr>
</thead>
</table>

---

**Definitions**

| Definition (cover of a function) |  
|----------------------------------|---
| Let                              |  
| - a function $g : D \to \mathbb{R}$, $x \mapsto g(x)$ on a domain $D \subseteq \mathbb{R}^n$, |  
| - a point $x^* \in D$, 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 |  
| $$\{(x, g(x)) \mid x \in D, x_k = x_k^* \text{ for all } k \in C\} \quad (4)$$ |  
| 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

1. $P$ be an MINLP of form (1),
2. $x^* \in [L, U]$, and
3. $C \subseteq \{1, \ldots, n\}$ 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 an approximate solution $x^* \in [L, U]$ 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.
A generic algorithm

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

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 an approximate solution \( x^* \in [L, U] \) 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

\begin{algorithm}
\begin{algorithmic}
\STATE \textbf{Input}: MINLP $P$ as in (1)
\STATE \textbf{begin}
\STATE \hspace{0.5em} compute an approximate solution $x^* \in [L, U]$ of $P$
\STATE \hspace{0.5em} round $x_k^*$ for all $k \in \mathcal{I}$
\STATE \hspace{0.5em} determine an $x^*$-cover $C$ of $P$
\STATE \hspace{0.5em} solve the sub-MIP of $P$
\hspace{0.5em} given by fixing $x_k = x_k^*$
\hspace{0.5em} for all $k \in \mathcal{C}$
\STATE \textbf{end}
\end{algorithmic}
\end{algorithm}

Remarks:
\begin{itemize}
\item As an approximation e.g. use an LP or NLP relaxation within a branch-and-bound solver.
\item MIP heuristics need to trade-off between fixing many vs. few (integer) variables: often minimum fixing rate.
\item We have to fix nonlinear variables, thus as few as possible to reduce the impact on the MINLP.
\end{itemize}
A generic algorithm

1 **Input**: MINLP $P$ as in (1)

2 **begin**

3 compute an approximate solution $x^* \in [L, U]$ 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. compute an approximate solution $x^* \in [L, U]$ 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.
A generic algorithm

1 Input: MINLP $P$ as in (1)
2 begin
3     compute an approximate solution $x^* \in [L, U]$ 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.
1. Introduction: MINLP, MIQCP, MIBCP
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

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

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

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

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 \( g : \mathbb{R}^n \to \mathbb{R}, x \mapsto x^T Q x, \) \( Q \in \mathbb{R}^{n \times n} \) symmetric, \( x^* \in \mathbb{R}^n, C \subseteq \{1, \ldots, n\} \).

Fixing variables with indices in \( C \) transforms

\[
x^T Q x \quad \sim \quad x_k = x^*_k \quad \forall k \in C
\]

\[
y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}
\]

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

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

\[
q_{uv} = 0 \text{ for all } u, v \not\in C \quad \iff \quad \text{set covering:}
\]

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

cover nonzeros in \( Q \) by incident rows/columns

independent of fix. values.
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

$$x^T Q x \quad \xrightarrow{k \in C} \quad y^T \tilde{Q} y + \tilde{q}^T y + \tilde{c}$$

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

$$q_{uv} = 0 \text{ for all } u, v \notin C \quad \iff \quad \text{set covering:}$$

cover nonzeros in $Q$ by incident rows/columns

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

Fixing variables with indices in \( C \) transforms

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

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

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

independent of fix. values.
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

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

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

$q_{uv} = 0$ for all $u, v \notin C$ \iff set covering:

cover nonzeros in $Q$ by incident rows/columns

independent of fix. values.
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

\[
\alpha_k = 1 \quad \text{f.a. } i \in \{1, \ldots, m\}, k \in \{1, \ldots, n\}, A_{kk}^i \neq 0, L_k \neq U_k, \tag{5}
\]

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

\[
L_k \neq U_k, L_j \neq U_j. \tag{6}
\]

To find a minimum cover, we solve the covering problem

\[
\min \left\{ \sum_{k=1}^{n} \alpha_k : (5), (6), \alpha \in \{0, 1\}^n \right\}. \tag{7}
\]
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

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

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

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

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 variable 

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

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

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

- (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 $x_1^{p_1} \cdot \cdot x_n^{p_n}$, $p_1, \ldots, p_n \in \mathbb{N}_0$, are

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

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

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”.

12 / 34
1. Introduction: MINLP, MIQCP, MIBCP
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

**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)
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)
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 1.2.1.1 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 1.2.1.1 with CPLEX 12.1 and Ipopt 3.7 (ma27)
  - for sub-MIP: emphasis feasibility and fast presolving settings, node limit 500

- **Comparing with**
  - SCIP 1.2.1.1 with CPLEX 12.1 and Ipopt 3.7 (ma27) (incl. nonlinear RENS)
  - settings: default, node limit 1, no time limit

- **Reported**:
  - nonlinear nonzeros/variable, % variables fixed by Undercover,
  - solution values of Undercover (*: sub-MIP optimal), plain SCIP, and best known solution value from MINLPLib website
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>best known</th>
</tr>
</thead>
<tbody>
<tr>
<td>netmod_dol1</td>
<td>0.00</td>
<td>0.30</td>
<td>0*</td>
<td>-0.26321</td>
<td>-0.56</td>
</tr>
<tr>
<td>netmod_dol2</td>
<td>0.00</td>
<td>0.38</td>
<td>-0.078020*</td>
<td>-0.50562</td>
<td>-0.56</td>
</tr>
<tr>
<td>netmod_kar1</td>
<td>0.01</td>
<td>0.88</td>
<td>0*</td>
<td>0</td>
<td>-0.4198</td>
</tr>
<tr>
<td>netmod_kar2</td>
<td>0.01</td>
<td>0.88</td>
<td>0*</td>
<td>0</td>
<td>-0.4198</td>
</tr>
<tr>
<td>space25</td>
<td>0.12</td>
<td>1.04</td>
<td>–</td>
<td>–</td>
<td>484.33</td>
</tr>
<tr>
<td>ex1266</td>
<td>0.40</td>
<td>3.03</td>
<td>16.3*</td>
<td>–</td>
<td>16.3</td>
</tr>
<tr>
<td>util</td>
<td>0.07</td>
<td>3.13</td>
<td>999.58*</td>
<td>1000.5</td>
<td>999.58</td>
</tr>
<tr>
<td>feedtray2</td>
<td>10.70</td>
<td>3.26</td>
<td>–</td>
<td>–</td>
<td>0</td>
</tr>
<tr>
<td>ex1265</td>
<td>0.38</td>
<td>3.52</td>
<td>15.1*</td>
<td>–</td>
<td>10.3</td>
</tr>
<tr>
<td>ex1263</td>
<td>0.34</td>
<td>3.88</td>
<td>30.1*</td>
<td>–</td>
<td>19.6</td>
</tr>
<tr>
<td>tln12</td>
<td>1.70</td>
<td>3.99</td>
<td>–</td>
<td>–</td>
<td>90.5</td>
</tr>
<tr>
<td>ex1264</td>
<td>0.36</td>
<td>4.26</td>
<td>11.1*</td>
<td>–</td>
<td>8.6</td>
</tr>
</tbody>
</table>

- 9 instances feasible
- ex1266 and util optimal
### 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>best known</th>
</tr>
</thead>
<tbody>
<tr>
<td>waste</td>
<td>1.10</td>
<td>5.65</td>
<td><strong>608.76</strong></td>
<td>–</td>
<td>598.92</td>
</tr>
<tr>
<td>space25a</td>
<td>0.29</td>
<td>5.84</td>
<td>–</td>
<td>–</td>
<td>484.33</td>
</tr>
<tr>
<td>nuclear14a</td>
<td>4.98</td>
<td>6.43</td>
<td>–</td>
<td>–</td>
<td>-1.1280</td>
</tr>
<tr>
<td>nuclear14b</td>
<td>2.42</td>
<td>6.43</td>
<td>–</td>
<td>–</td>
<td>-1.1135</td>
</tr>
<tr>
<td>tln7</td>
<td>1.53</td>
<td>6.67</td>
<td><strong>30.3</strong></td>
<td>–</td>
<td>15</td>
</tr>
<tr>
<td>tln6</td>
<td>1.47</td>
<td>7.69</td>
<td><strong>20.3</strong></td>
<td>–</td>
<td>15.3</td>
</tr>
<tr>
<td>tloss</td>
<td>1.47</td>
<td>7.89</td>
<td><strong>16.3</strong></td>
<td>–</td>
<td>16.3</td>
</tr>
<tr>
<td>tln5</td>
<td>1.39</td>
<td>9.09</td>
<td><strong>15.1</strong></td>
<td>–</td>
<td>10.3</td>
</tr>
<tr>
<td>sep1</td>
<td>0.40</td>
<td>10.53</td>
<td>-510.08*</td>
<td>–</td>
<td>-510.08</td>
</tr>
<tr>
<td>tltr</td>
<td>1.10</td>
<td>12.50</td>
<td><strong>74.2</strong></td>
<td>–</td>
<td>48.067</td>
</tr>
</tbody>
</table>

- 7 instances feasible
- tloss and sep1 optimal
## 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>best known</th>
</tr>
</thead>
<tbody>
<tr>
<td>nous1</td>
<td>2.39</td>
<td>19.44</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>0.626</td>
</tr>
<tr>
<td>meanvarx</td>
<td>0.19</td>
<td>23.33</td>
<td>15.925*</td>
<td>14.369</td>
<td>14.369</td>
</tr>
<tr>
<td>product2</td>
<td>0.37</td>
<td>26.15</td>
<td>–</td>
<td>–</td>
<td>-2102.4</td>
</tr>
<tr>
<td>product</td>
<td>0.17</td>
<td>30.87</td>
<td>–</td>
<td>–</td>
<td>-2142.9</td>
</tr>
<tr>
<td>spectra2</td>
<td>3.43</td>
<td>35.71</td>
<td>31.981*</td>
<td>13.978</td>
<td>13.978</td>
</tr>
<tr>
<td>fac3</td>
<td>0.81</td>
<td>78.26</td>
<td>13065e4*</td>
<td>7213e4</td>
<td>3198e4</td>
</tr>
<tr>
<td>nvs19</td>
<td>8.00</td>
<td>88.89</td>
<td>–</td>
<td>0</td>
<td>-1098.4</td>
</tr>
<tr>
<td>nvs23</td>
<td>9.00</td>
<td>90.00</td>
<td>–</td>
<td>0</td>
<td>-1125.2</td>
</tr>
<tr>
<td>du-opt5</td>
<td>0.95</td>
<td>94.74</td>
<td>3407.1*</td>
<td>14.168</td>
<td>8.0737</td>
</tr>
<tr>
<td>du-opt</td>
<td>0.95</td>
<td>95.24</td>
<td>4233.9*</td>
<td>4233.9</td>
<td>3.5563</td>
</tr>
</tbody>
</table>

5 instances feasible
Computational results for MIQCPs

Feasible solutions

- Undercover: 21 instances
- SCIP: 13 instances
- All: 24 instances

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.
Computational results for MIQCPs

- **Feasible solutions**
  - Undercover: 21 instances
  - SCIP: 13 instances
  - All: 24 instances

- **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: MINLP, MIQCP, MIBCP
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
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.

$\Rightarrow$ “re-cover”: solve the covering problem again considering all bound changes from fix-and-propagate.
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-MILP
NLP postprocessing

▷ 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
<table>
<thead>
<tr>
<th>instance</th>
<th>nnz/var</th>
<th>% cov</th>
<th>UC-LP</th>
<th>UC-NLP</th>
<th>best known</th>
</tr>
</thead>
<tbody>
<tr>
<td>netmod_dol1</td>
<td>0.00</td>
<td>0.30</td>
<td>0*</td>
<td>–</td>
<td>-0.56</td>
</tr>
<tr>
<td>netmod_dol2</td>
<td>0.00</td>
<td>0.38</td>
<td>-0.078020*</td>
<td>–</td>
<td>-0.56</td>
</tr>
<tr>
<td>netmod_kar1</td>
<td>0.01</td>
<td>0.88</td>
<td>0*</td>
<td>-0.33974</td>
<td>-0.4198</td>
</tr>
<tr>
<td>netmod_kar2</td>
<td>0.01</td>
<td>0.88</td>
<td>0*</td>
<td>-0.33974</td>
<td>-0.4198</td>
</tr>
<tr>
<td>ex1266</td>
<td>0.40</td>
<td>3.03</td>
<td>16.3</td>
<td>17.3*</td>
<td>16.3</td>
</tr>
<tr>
<td>ex1264</td>
<td>0.36</td>
<td>4.26</td>
<td>11.1</td>
<td>9.6*</td>
<td>8.6</td>
</tr>
<tr>
<td>tln7</td>
<td>1.53</td>
<td>6.67</td>
<td>30.3*</td>
<td>29.4*</td>
<td>15</td>
</tr>
<tr>
<td>tln6</td>
<td>1.47</td>
<td>7.69</td>
<td>20.3*</td>
<td>31.3*</td>
<td>15.3</td>
</tr>
<tr>
<td>tloss</td>
<td>1.47</td>
<td>7.89</td>
<td>16.3*</td>
<td>21.5*</td>
<td>16.3</td>
</tr>
<tr>
<td>meanvarx</td>
<td>0.19</td>
<td>23.33</td>
<td>15.925*</td>
<td>–</td>
<td>14.369</td>
</tr>
<tr>
<td>spectra2</td>
<td>3.43</td>
<td>35.71</td>
<td>31.981*</td>
<td>19.301*</td>
<td>13.978</td>
</tr>
<tr>
<td>fac3</td>
<td>0.81</td>
<td>78.26</td>
<td>13065e4*</td>
<td>3479e4*</td>
<td>3198e4</td>
</tr>
<tr>
<td>du-opt5</td>
<td>0.95</td>
<td>94.74</td>
<td>3407.1*</td>
<td>20.438*</td>
<td>8.0737</td>
</tr>
<tr>
<td>du-opt</td>
<td>0.95</td>
<td>95.24</td>
<td>4233.9*</td>
<td>10.172*</td>
<td>3.5563</td>
</tr>
</tbody>
</table>

- 3 instances feasible (not counting NLP error on product2, waste)
- on feasible instances: NLP 8 times, LP 3 times better, 6 times equal
If the sub-MIP is infeasible, this is typically detected
- during fix-and-propagate, or
- via infeasible root LP.
Avoiding/exploiting infeasibility

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

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

Then we can perform conflict analysis: generate conflict clauses valid 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.
Avoiding/exploiting infeasibility

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

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

Then we can perform **conflict analysis**: generate conflict clauses **valid 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, i.e. the heuristic is unsuccessful, at least this gives us valid conflicts to prune the search tree in the original problem.
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)
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 $\mapsto$ MIP (so far, in general very restrictive)
  - MINLP $\mapsto$ MIQCP
  - nonconvex MINLP $\mapsto$ 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:

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

subject to linear/combinatorial constraints,

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

$WH = \sum_i A_i$,

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:

$$\text{minimise} \quad W + H + \sum_i w_i + \sum_i h_i$$
$$\text{subject to} \quad \text{linear/combinatorial constraints,}$$
$$\text{bilinear} \quad \Rightarrow \quad w_i h_i = A_i \quad \text{for } i = 1, \ldots, n,$$
$$\text{bilinear} \quad \Rightarrow \quad WH = \sum_i A_i,$$

bounded widths and heights.

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

Given

- a fixed number $n$ of rectangles
- with fixed areas $A_1, \ldots, A_n$

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

\[
\begin{align*}
\text{minimise} & \quad W + H + \sum_i w_i + \sum_i h_i \\
\text{subject to} & \quad \text{linear/combinatorial constraints,}
\end{align*}
\]

\begin{align*}
\text{univariate nonlinear} & \quad w_i = \frac{A_i}{h_i} \quad \text{for } i = 1, \ldots, n, \\
\text{univariate nonlinear} & \quad W = \frac{(\sum_i A_i)}{H},
\end{align*}

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:

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

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 $\leadsto$ 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</td>
</tr>
<tr>
<td>1,3,5,12</td>
<td>2.70</td>
<td>2.58</td>
<td></td>
</tr>
</tbody>
</table>
1 Introduction: MINLP, MIQCP, MIBCP
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
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
- implementation and experiments for general MINLPs
- experiments on specific problems
- tuning for efficient use within the branch-and-bound tree


Undercover

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

Ambros M. Gleixner

Zuse Institute Berlin • MATHEON • Berlin Mathematical School

joint work with Timo Berthold

European Workshop on Mixed-Integer Nonlinear Programming, 16 April 2010