Primal heuristic for MINLPs in SCIP

Ambros Gleixner, and Felipe Serrano

Zuse Institute Berlin · serrano@zib.de
SCIP Optimization Suite · http://scip.zib.de

Workshop on Discrepancy Theory and Integer Programming
Amsterdam · June 12, 2018
Outline

Introduction: LP-based Branch and Bound
   Spatial Branch and bound

Heuristics
   Sub-NLP
   NLP diving
   Multi-start
   MPEC
   Undercover
   RENS

Conclusion
Mixed-Integer Nonlinear Programs (MINLPs)

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad g_k(x) \leq 0 \quad \forall k \in [m] \\
& \quad x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I} \subseteq [n] \\
& \quad x_i \in [\ell_i, u_i] \quad \forall i \in [n]
\end{align*}
\]

The functions \( g_k \in C^1([\ell, u], \mathbb{R}) \) can be convex or nonconvex.
One way of solving MINLPs to global optimality

- Methods for finding (good) feasible solutions
  - Primal heuristics

- Proof that there is no better solution
  - LP-based spatial branch and bound
LP based spatial Branch & Bound

- Build a (extended formulation of a) polyhedral relaxation $\mathcal{R}$

- Solve $\mathcal{R}$ and get solution $x^*$

- If $x^*$ is feasible we are done. If not,

- Try to strengthen $\mathcal{R}$ by separating $x^*$

- When not possible, branch possibly on continuous variables (spatially)
Building polyhedral relaxations: the problem

• We can start with the variable’s bounds as our relaxation.
• Then we have to solve the separation problem: Given \( \{x \in [l, u] : g(x) \leq 0\} \) and \( \bar{x} \) s.t. \( g(\bar{x}) > 0 \) either
  • Find a separating inequality or
  • prove that none exists.
Building polyhedral relaxations: the problem

- We can start with the variable’s bounds as our relaxation.
- Then we have to solve the separation problem: Given \( \{ x \in [l, u] : g(x) \leq 0 \} \) and \( \bar{x} \) s.t. \( g(\bar{x}) > 0 \) either
  - Find a separating inequality or
  - prove that none exists.
- Very expensive for general \( g \)
Building polyhedral relaxations: the problem

- We can start with the variable’s bounds as our relaxation.
- Then we have to solve the separation problem: Given \( \{x \in [l, u] : g(x) \leq 0\} \) and \( \bar{x} \) s.t. \( g(\bar{x}) > 0 \) either
  - Find a separating inequality or
  - prove that none exists.
- Very expensive for general \( g \)
- However, when \( g \) is convex, it is as easy as computing a gradient:
  \[
g(\bar{x}) + \nabla g(\bar{x})(x - \bar{x}) \leq 0
  \]
Building polyhedral relaxations: the problem

- We can start with the variable’s bounds as our relaxation.
- Then we have to solve the separation problem: Given \( \{x \in [l, u] : g(x) \leq 0\} \) and \( \bar{x} \) s.t. \( g(\bar{x}) > 0 \) either
  - Find a separating inequality or
  - prove that none exists.
- Very expensive for general \( g \)
- However, when \( g \) is convex, it is as easy as computing a gradient:
  \[
  g(\bar{x}) + \nabla g(\bar{x})(x - \bar{x}) \leq 0
  \]
- Idea: find convex underestimator \( \hat{g} \) of \( g(x) \)
Building polyhedral relaxations: the problem

- We can start with the variable’s bounds as our relaxation.
- Then we have to solve the separation problem: Given \( \{x \in [l, u] : g(x) \leq 0\} \) and \( \bar{x} \) s.t. \( g(\bar{x}) > 0 \) either
  - Find a separating inequality or
  - prove that none exists.
- Very expensive for general \( g \)
- However, when \( g \) is convex, it is as easy as computing a gradient:
  \[
g(\bar{x}) + \nabla g(\bar{x})(x - \bar{x}) \leq 0
  \]
- Idea: find convex underestimator \( \hat{g} \) of \( g(x) \)
- Then \( \{x \in [l, u] : g(x) \leq 0\} \subseteq \{x \in [l, u] : \hat{g}(x) \leq 0\} \)
- If \( \hat{g}(\bar{x}) > 0 \) we can separate.
Building polyhedral relaxations: an example

- $x^2 + y^2 + 2 \exp(xy^3) \leq 3$ with $x, y \in [-2, 2]$
Building polyhedral relaxations: an example

- $x^2 + y^2 + 2\exp(xy^3) \leq 3$ with $x, y \in [-2, 2]$
- $\exp(\cdot) > 0 \Rightarrow x^2 + y^2 \leq x^2 + y^2 + 2\exp(xy^3)$
Building polyhedral relaxations: an example

- $x^2 + y^2 + 2\exp(xy^3) \leq 3$ with $x, y \in [-2, 2]$
- $\exp(\cdot) > 0 \Rightarrow x^2 + y^2 \leq x^2 + y^2 + 2\exp(xy^3)$

- Admittedly, ad-hoc argument
- In practice: if functions are simple enough, we know convex/concave envelopes
- If function is not simple enough, make it simpler!
Building polyhedral relaxation in SCIP

- \( x^2 + y^2 + 2 \exp(xy^3) \leq 3 \)
- Introduce auxiliary variables
  - \( z_1 = y^3 \)
  - \( z_2 = xz_1 \)
  - \( x^2 + y^2 + 2 \exp(z_2) \leq 3 \)
- We can find polyhedral relaxations of \( z_1 = y^3 \)
- For \( z_2 = xz_1 \) we have McCormick inequalities:
  \[
  \max \{ xz_1 + z_1x - xz_1, x\bar{z}_1 + z_1\bar{x} - \bar{x}\bar{z}_1 \} \leq z_2 \leq \min \{ x\bar{z}_1 + z_1x - x\bar{z}_1, xz_1 + z_1\bar{x} - \bar{x}z_1 \}
  \]
- Finally, \( x^2 + y^2 + 2 \exp(z_2) \) is convex
Spatial Branch and bound

Solutions might not be separable: \( \text{conv}\{(x, y) : x^2 = y, x \in [\ell, u]\} \) is

\[
x^2 \leq y \leq \ell^2 + \frac{u^2 - \ell^2}{u - \ell} (x - \ell) \quad \forall x \in [\ell, u].
\]
Spatial Branch and bound

Solutions might not be separable: \( \text{conv}\{ (x, y) : x^2 = y, x \in [\ell, u] \} \) is

\[
x^2 \leq y \leq \ell^2 + \frac{u^2 - \ell^2}{u - \ell} (x - \ell) \quad \forall x \in [\ell, u].
\]

Branching on a nonlinear variable in a nonconvex constraint allows for tighter relaxations:
Outline

Introduction: LP-based Branch and Bound
Spatial Branch and bound

Heuristics
Sub-NLP
NLP diving
Multi-start
MPEC
Undercover
RENS

Conclusion
Sub-NLP

- Idea: fix integer variables to integer values and run a local NLP solver
- Good for: MINLP

Let $\bar{x}$ be LP-optimum of the current node’s relaxation

- If there is a $i \in I$ such that $\bar{x}_i \notin \mathbb{Z} \rightarrow$ STOP
- Fix $x_i$ to $\bar{x}_i$.
- Solve remaining NLP to local optimality using $\bar{x}$ as initial point.
**Sub-NLP**

- Idea: fix integer variables to integer values and run a local NLP solver
- Good for: MINLP

Let $\bar{x}$ be LP-optimum of the current node’s relaxation
- If there is a $i \in I$ such that $\bar{x}_i \not\in \mathbb{Z} \rightarrow$ STOP
- Fix $x_i$ to $\bar{x}_i$.
- Solve remaining NLP to local optimality using $\bar{x}$ as initial point.

Extends MIP heuristics to MINLP
- SCIP runs its MIP heuristics on MIP relaxation of MINLP
- use heuristic’s proposed solution as $\bar{x}$
NLP diving

- Idea: Solve NLP relaxations, fixing an integer variable after each NLP
- Good for: MINLP

- solve NLP relaxation
- fix an integer variable
- propagate
- repeat
- if fixing is infeasible, backtrack
  - undo last fixing and fix to another value
  - if infeasible again, abort
Multi-start Heuristic [Chinneck et al. 2013]

- Idea: use different starting points for NLP solver
- Good for: NLPs without too many integer variables

\[
4 \leq x^2 + y^2 \leq 9 \\
4 \leq (x - 2)^2 + y^2 \leq 9 \\
x \in [-4, 6] \\
y \in [-4, 4]
\]
Multi-start in SCIP

Input: Nonlinear Constraints $g_i(x) \leq 0$

1. generate random points in $[\ell, u]$
Multi-start in SCIP

Input: Nonlinear Constraints $g_i(x) \leq 0$

1. generate random points in $[\ell, u]$
Multi-start in SCIP

Input: Nonlinear Constraints \( g_i(x) \leq 0 \)

1. generate random points in \([\ell, u]\)
2. for each point \( x_k \):
   - \( s_i^k := \frac{-g_i(x_k)}{||\nabla g_i(x_k)||^2} \nabla g_i(x_k) \ \forall i \)
   - Update:
     \[
     x_k^+ = x_k + \frac{1}{n} \sum_{i=1}^{n} s_i^k
     \]
Multi-start in SCIP

Input: Nonlinear Constraints \( g_i(x) \leq 0 \)

1. generate random points in \([\ell, u]\)

2. for each point \(x_k\):
   - \( s_i^k := \frac{-g_i(x_k)}{||\nabla g_i(x_k)||^2} \nabla g_i(x_k) \ \forall i \)
   - Update:
     \[
x_k^+ = x_k + \frac{1}{n} \sum_{i=1}^{n} s_i^k
\]
Multi-start in SCIP

Input: Nonlinear Constraints $g_i(x) \leq 0$

1. generate random points in $[\ell, u]$
2. for each point $x_k$:
   - $s_i^k := \frac{-g_i(x_k)}{||\nabla g_i(x_k)||^2} \nabla g_i(x_k) \forall i$
   - Update:
     \[
     x_k^+ = x_k + \frac{1}{n} \sum_{i=1}^{n} s_i^k
     \]
Multi-start in SCIP

Input: Nonlinear Constraints $g_i(x) \leq 0$

1. generate random points in $[\ell, u]$
2. for each point $x_k$:
   - $s_i^k := \frac{-g_i(x_k)}{||\nabla g_i(x_k)||^2} \nabla g_i(x_k) \forall i$
   - Update:
     $$x_k^+ = x_k + \frac{1}{n} \sum_{i=1}^{n} s_i^k$$

3. identify clusters of points $C_1, \ldots , C_p$
Multi-start in SCIP

Input: Nonlinear Constraints $g_i(x) \leq 0$

1. Generate random points in $[\ell, u]$
2. For each point $x_k$:
   - $s^k_i := \frac{-g_i(x_k)}{||\nabla g_i(x_k)||^2} \nabla g_i(x_k)$ $\forall i$
   - Update:
     $$x_k^+ = x_k + \frac{1}{n} \sum_{i=1}^{n} s^k_i$$
3. Identify clusters of points $C_1, \ldots, C_p$
4. For each cluster
   - Build convex combination
     $$y := \frac{1}{|C_j|} \sum_{x \in C_j} x$$
   - Round each fractional integer variable to closest integer
   - Use resulting point as starting point for NLP
MPEC Heuristic [Schewe and Schmidt 2016]

- Idea: write $x \in \{0, 1\}$ as $x(1 - x) = 0$, relax to $x(1 - x) \leq \varepsilon$ and solve sequences of NLP with $\varepsilon \to 0$
- Good for: Mixed Binary NLPs
- MPEC stands for Mathematical Program with Equilibrium Constraints

\[
\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad g_j(x) \leq 0 \quad \forall j \in \{1, \ldots, m\} \\
& \quad x_i \in [\ell_i, u_i] \quad \forall i \in \{1, \ldots, k\} \\
& \quad x_i \in \{0, 1\} \quad \forall i \in \{k + 1, \ldots, n\}
\end{align*}
\]
MPEC Heuristic [Schewe and Schmidt 2016]

- Idea: write $x \in \{0, 1\}$ as $x(1 - x) = 0$, relax to $x(1 - x) \leq \varepsilon$ and solve sequences of NLP with $\varepsilon \rightarrow 0$
- Good for: Mixed Binary NLPs
- MPEC stands for Mathematical Program with Equilibrium Constraints

$$\begin{align*}
\min & \quad f(x) \\
\text{s.t.} & \quad g_j(x) \leq 0 \quad \forall j \in \{1, \ldots, m\} \\
& \quad x_i \in [\ell_i, u_i] \quad \forall i \in \{1, \ldots, k\} \\
& \quad x_i \in [0, 1] \quad \forall i \in \{k + 1, \ldots, n\} \\
& \quad x_i(1 - x_i) \leq 0 \quad \forall i \in \{k + 1, \ldots, n\}
\end{align*}$$

(NLP)
MPEC Heuristic [Schewe and Schmidt 2016]

- Idea: write $x \in \{0, 1\}$ as $x(1 - x) = 0$, relax to $x(1 - x) \leq \varepsilon$ and solve sequences of NLP with $\varepsilon \rightarrow 0$
- Good for: Mixed Binary NLPs
- MPEC stands for Mathematical Program with Equilibrium Constraints

$$\begin{align*}
\min f(x) \\
\text{s.t.} & \quad g_j(x) \leq 0 \quad \forall j \in \{1, \ldots, m\} \\
& \quad x_i \in [\ell_i, u_i] \quad \forall i \in \{1, \ldots, k\} \\
& \quad x_i \in [0, 1] \quad \forall i \in \{k + 1, \ldots, n\} \\
& \quad x_i(1 - x_i) \leq \varepsilon \quad \forall i \in \{k + 1, \ldots, n\}
\end{align*}$$

$(NLP_\varepsilon)$
Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, 1/4]$ and $x \in \mathbb{R}^n$

1. $x^* :=$ solution of $NLP_\varepsilon$ using $x$ as starting point
Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$
1. $x^* :=$ solution of $NLP_\varepsilon$ using $x$ as starting point
2. if $x^*$ is feasible for $NLP_\varepsilon$ and $x_i^* \in \{0, 1\} \rightarrow$ STOP

3. if $x^*$ is feasible for $NLP_\varepsilon$ but not binary: 
   - $x^* := x$ 
   - GOTO 1.

4. $x^*$ is infeasible for $NLP_\varepsilon$ but satisfies 
   - $x_i^*(1-x_i^*)$ 
   - STOP

5. $x^*$ is infeasible for $NLP_\varepsilon$ and there are 
   - reset or fix $x_i$ for binary variables with 
   - $x_i^*(1-x_i^*)$ 
   - $x_i^* := \begin{cases} 0; & x_i^* > 0 \\ 1; & \text{otherwise} \end{cases}$
   - GOTO 1.
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$
1. $x^* :=$ solution of $NLP_\varepsilon$ using $x$ as starting point
2. if $x^*$ is feasible for $NLP_\varepsilon$ and $x_i^* \in \{0, 1\} \rightarrow$ STOP
3. if $x^*$ is feasible for $NLP_\varepsilon$ but not binary:
   • $x^* \rightarrow$ STOP
   • $x^* \rightarrow$ STOP
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$
1. $x^* :=$ solution of $NLP_\varepsilon$ using $x$ as starting point
2. if $x^*$ is feasible for $NLP_\varepsilon$ and $x_i^* \in \{0, 1\} \rightarrow$ STOP
3. if $x^*$ is feasible for $NLP_\varepsilon$ but not binary:
   - $\varepsilon := \frac{\varepsilon}{2}$; GOTO 1. with $x := x^*$
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$

1. $x^* :=$ solution of $NLP_{\varepsilon}$ using $x$ as starting point

2. if $x^*$ is feasible for $NLP_{\varepsilon}$ and $x^*_i \in \{0, 1\} \rightarrow$ STOP

3. if $x^*$ is feasible for $NLP_{\varepsilon}$ but not binary:
   - $\varepsilon := \frac{\varepsilon}{2}$; GOTO 1. with $x := x^*$

4. $x^*$ is infeasible for $NLP_{\varepsilon}$ but satisfies $x^*_i(1 - x^*_i) \leq \varepsilon \rightarrow$ STOP
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$
1. $x^* :=$ solution of $NLP_{\varepsilon}$ using $x$ as starting point
2. if $x^*$ is feasible for $NLP_{\varepsilon}$ and $x^*_i \in \{0, 1\} \rightarrow$ STOP
3. if $x^*$ is feasible for $NLP_{\varepsilon}$ but not binary:
   - $\varepsilon := \frac{\varepsilon}{2}$; GOTO 1. with $x := x^*$
4. $x^*$ is infeasible for $NLP_{\varepsilon}$ but satisfies $x^*_i(1 - x^*_i) \leq \varepsilon \rightarrow$ STOP
5. $x^*$ is infeasible for $NLP_{\varepsilon}$ and there are $x^*_i(1 - x^*_i) > \varepsilon$
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$

1. $x^* :=$ solution of $NLP_\varepsilon$ using $x$ as starting point

2. if $x^*$ is feasible for $NLP_\varepsilon$ and $x_i^* \in \{0, 1\} \rightarrow$ STOP

3. if $x^*$ is feasible for $NLP_\varepsilon$ but not binary:
   - $\varepsilon := \frac{\varepsilon}{2}$; GOTO 1. with $x := x^*$

4. $x^*$ is infeasible for $NLP_\varepsilon$ but satisfies $x_i^*(1 - x_i^*) \leq \varepsilon \rightarrow$ STOP

5. $x^*$ is infeasible for $NLP_\varepsilon$ and there are $x_i^*(1 - x_i^*) > \varepsilon$
   - reset or fix $x_i$ for binary variables with $x_i^*(1 - x_i^*) > \varepsilon$
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$
1. $x^* := \text{solution of } NLP_\varepsilon \text{ using } x \text{ as starting point}$
2. if $x^*$ is feasible for $NLP_\varepsilon$ and $x_i^* \in \{0, 1\} \rightarrow \text{STOP}$
3. if $x^*$ is feasible for $NLP_\varepsilon$ but not binary:
   • $\varepsilon := \frac{\varepsilon}{2}$; GOTO 1. with $x := x^*$
4. $x^*$ is infeasible for $NLP_\varepsilon$ but satisfies $x_i^*(1 - x_i^*) \leq \varepsilon \rightarrow \text{STOP}$
5. $x^*$ is infeasible for $NLP_\varepsilon$ and there are $x_i^*(1 - x_i^*) > \varepsilon$
   • reset or fix $x_i$ for binary variables with $x_i^*(1 - x_i^*) > \varepsilon$
   • $x_i := \begin{cases} 0, & x_i^* > 0.5 \\ 1, & \text{otherwise} \end{cases}$
MPEC in SCIP

Algorithm

0. choose $\varepsilon \in [0, \frac{1}{4}]$ and $x \in \mathbb{R}^n$

1. $x^* :=$ solution of $NLP_\varepsilon$ using $x$ as starting point

2. if $x^*$ is feasible for $NLP_\varepsilon$ and $x^*_i \in \{0, 1\} \rightarrow$ STOP

3. if $x^*$ is feasible for $NLP_\varepsilon$ but not binary:
   - $\varepsilon := \frac{\varepsilon}{2}$; GOTO 1. with $x := x^*$

4. $x^*$ is infeasible for $NLP_\varepsilon$ but satisfies $x^*_i(1 - x^*_i) \leq \varepsilon \rightarrow$ STOP

5. $x^*$ is infeasible for $NLP_\varepsilon$ and there are $x^*_i(1 - x^*_i) > \varepsilon$
   - reset or fix $x_i$ for binary variables with $x^*_i(1 - x^*_i) > \varepsilon$
   - $x_i := \begin{cases} 0, & x^*_i > 0.5 \\ 1, & \text{otherwise} \end{cases}$
   - GOTO 1.
Undercover [Berthold and Gleixner 2014]

- Idea: fix variables so that MINLP transforms into a MIP
- Good for: MIQCQP

- Solve a relaxation (LP, NLP)
- Identify minimum set $C$ of variables to be fixed in order to obtain a MIP
- Fix variables in $C$ to relaxation’s solution.
- Solve MIP
- Postprocess: Fix integer values and solve NLP
Undercover in SCIP

- Identifying minimum set $\mathcal{C}$ of variables to be fixed
  - If $\frac{\partial^2}{\partial x_i \partial x_j} g_k(x) \neq 0$ then $x_i$ or $x_j$ must be fixed
  - Build graph with nodes $x_i$ and arcs $\{x_i, x_j\}$ if $\frac{\partial}{\partial x_i x_j} g_k(x) \neq 0$
  - Minimum vertex cover yields the set we are looking for.

- Fix variables in $\mathcal{C}$ to relaxation’s solution
  - Fix a variable, if variable must be integer, fix to rounded value.
  - Propagate bounds: to avoid ”obvious” infeasible fixings
  - If next fixing value is outside bounds, choose closest bound or resolve relaxation
  - If infeasible, backtrack: undo last fixing and try new value
RENs: The optimal rounding [Berthold 2014]

- RENS stands for Relaxation Enforced Neighborhood Search
- Idea: restrict integer variables around LP-relaxation’s solution and solve remaining MINLP
- Good for: MIP, MINLP

Let $\bar{x}$ be the LP solution
- If $|\{i \in \mathcal{I} : \bar{x}_i \in \mathbb{Z}\}| > p|\mathcal{I}|$ for some $p \in [0, 1]$ then
- Change bounds of $x_i$ to $[\lfloor \bar{x}_i \rfloor, \lceil \bar{x}_i \rceil]$
- Solve smaller MINLP

This gives the feasible rounding with best objective value
Computational results for RENS

Table 1  Computing optimal roundings (aggregated results)

<table>
<thead>
<tr>
<th></th>
<th>Integrality</th>
<th>Succ</th>
<th>Prim. gap</th>
<th>Comp. effort</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&gt;90 %</td>
<td>Avg (%)</td>
<td>Mean</td>
<td>Std dev</td>
</tr>
<tr>
<td>MIP + cuts</td>
<td>55/159</td>
<td>71.7</td>
<td>95/159</td>
<td>8.22</td>
</tr>
<tr>
<td>MIP − cuts</td>
<td>62/159</td>
<td>73.6</td>
<td>80/159</td>
<td>10.88</td>
</tr>
<tr>
<td>MIQCP (LP)</td>
<td>9/70</td>
<td>59.9</td>
<td>49/70</td>
<td>11.61</td>
</tr>
<tr>
<td>MIQCP (NLP)</td>
<td>1/70</td>
<td>13.8</td>
<td>48/70</td>
<td>1.30</td>
</tr>
<tr>
<td>MINLP (LP)</td>
<td>6/105</td>
<td>63.5</td>
<td>65/105</td>
<td>13.80</td>
</tr>
<tr>
<td>MINLP (NLP)</td>
<td>1/105</td>
<td>15.0</td>
<td>73/105</td>
<td>4.60</td>
</tr>
</tbody>
</table>

- Basic LP solutions often show high integrality
- Success rate seems to decreases when more integer variables are fixed
- Roundability of MIP and MINLP are similar
Computational results for RENS

- Basic LP solutions often show high integrality
- Success rate seems to decreases when more integer variables are fixed
- Roundability of MIP and MINLP are similar

**Fig. 3** Moving averages of success rate, MIPLIB instances, after cuts
Outline

Introduction: LP-based Branch and Bound
  Spatial Branch and bound

Heuristics
  Sub-NLP
  NLP diving
  Multi-start
  MPEC
  Undercover
  RENS

Conclusion
Conclusion

Summary

• very brief introduction to LP-based branch and bound
• overview of MINLP heuristics in SCIP
Conclusion

Summary

• very brief introduction to LP-based branch and bound
• overview of MINLP heuristics in SCIP

What is missing in SCIP

• Non-linear Feasibility Pump
• Problem specific heuristics
Conclusion

Summary

• very brief introduction to LP-based branch and bound
• overview of MINLP heuristics in SCIP

What is missing in SCIP

• Non-linear Feasibility Pump
• Problem specific heuristics

Thank you for your attention
Primal heuristic for MINLPs in SCIP

Ambros Gleixner, and Felipe Serrano

Zuse Institute Berlin · serrano@zib.de
SCIP Optimization Suite · http://scip.zib.de

Workshop on Discrepancy Theory and Integer Programming
Amsterdam · June 12, 2018
Timo Berthold.  
RENS – the optimal rounding.  

Timo Berthold and Ambros M. Gleixner.  
Undercover: a primal MINLP heuristic exploring a largest sub-MIP.  

Lars Schewe and Martin Schmidt.  
Computing feasible points for minlps with mpecs.  
Laurence Smith, John Chinneck, and Victor Aitken. 
Improved constraint consensus methods for seeking feasibility in nonlinear programs. 
doi: 10.1007/s10589-012-9473-z.