MINLP Solver Technology

Stefan Vigerske
October 24, 2017
Outline

Introduction

Fundamental Methods

   Recap: Mixed-Integer Linear Programming

   Convex MINLP

   Nonconvex MINLP

   Bound Tightening

Acceleration – Selected Topics

   Optimization-based bound tightening

   Synergies with MIP and NLP

   Convexity

   Convexification

   Primal Heuristics
Introduction
Mixed-Integer Nonlinear Programs (MINLPs)

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

The functions \( g_k \in C^1([\ell, u], \mathbb{R}) \) can be

- convex
- nonconvex
Convex MINLP:

- Main **difficulty**: Integrality restrictions on variables
- Main **challenge**: Integrating techniques for MIP (branch-and-bound) and NLP (SQP, interior point, Kelley’ cutting plane, ...)
Solving MINLPs

Convex MINLP:

- Main difficulty: Integrality restrictions on variables
- Main challenge: Integrating techniques for MIP (branch-and-bound) and NLP (SQP, interior point, Kelley’ cutting plane, ...)

General MINLP = Convex MINLP plus Global Optimization:

- Main difficulty: Nonconvex nonlinearities
- Main challenges:
  - Convexification of nonconvex nonlinearities
  - Reduction of convexification gap (spatial branch-and-bound)
  - Numerical robustness
  - Diversity of problem class: MINLP is “The mother of all deterministic optimization problems” (Jon Lee, 2008)
## Solvers for Convex MINLP

<table>
<thead>
<tr>
<th>solver</th>
<th>citation</th>
<th>OA NLP-BB</th>
<th>LP/NLP Alignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlphaECP</td>
<td>Westerlund and Lundquist [2005],LASTUSILTA [2011]</td>
<td></td>
<td>ECP</td>
</tr>
<tr>
<td>AOA</td>
<td>Roelofs and Bisschop [2017] (AIMMS)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Bonmin</td>
<td>Bonami, Biegler, Conn, Cornuëjols, Grossmann, Laird, Lee, Lodi, Margot, Sawaya, and Wächter [2008]</td>
<td>✓ ✓ ✓</td>
<td></td>
</tr>
<tr>
<td>DICOPT</td>
<td>Kocis and Grossmann [1989]</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>FilMINT</td>
<td>Abhishek, Leyffer, and Linderoth [2010]</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Knitro</td>
<td>Byrd, Nocedal, and Waltz [2006]</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>MINLPBB</td>
<td>Leyffer [1998]</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>MINOTAUR</td>
<td>Mahajan, Leyffer, and Munson [2009]</td>
<td>(✓)</td>
<td></td>
</tr>
<tr>
<td>SBB</td>
<td>Bussieck and Drud [2001]</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>XPRESS-SLP</td>
<td>FICO [2008]</td>
<td>ECP</td>
<td>✓</td>
</tr>
</tbody>
</table>

- can often work as heuristic for nonconvex MINLP
## Solvers for General MINLP

### Deterministic:

<table>
<thead>
<tr>
<th>Solver</th>
<th>1st ver. citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>αBB</td>
<td>1995 Adjiman, Androulakis, and Floudas [1998a]</td>
</tr>
<tr>
<td>BARON</td>
<td>1996 Sahinidis [1996], Tawarmalani and Sahinidis [2005]</td>
</tr>
<tr>
<td>in gPROMS</td>
<td>1996 Smith and Pantelides [1997]</td>
</tr>
</tbody>
</table>

**Main restriction:** algebraic structure of problem must be available (see later)

**Interval-Arithmetic based:** avoid round-off errors, typically NLP only, e.g., COCONUT [Neumaier, 2001], Ibex, . . .

**Stochastic search:** LocalSolver, OQNLP [Ugray, Lasdon, Plummer, Glover, Kelly, and Martí, 2007], . . .
### Solvers for General MINLP

#### Deterministic:

<table>
<thead>
<tr>
<th>Solver</th>
<th>1st ver. citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>αBB</td>
<td>1995 Adjiman, Androulakis, and Floudas [1998a]</td>
</tr>
<tr>
<td>BARON</td>
<td>1996 Sahinidis [1996], Tawarmalani and Sahinidis [2005]</td>
</tr>
<tr>
<td>in gPROMS</td>
<td>1996 Smith and Pantelides [1997]</td>
</tr>
<tr>
<td>Couenne</td>
<td>2008 Belotti, Lee, Liberti, Margot, and Wächter [2009]</td>
</tr>
<tr>
<td>LindoAPI</td>
<td>2009 Lin and Schrage [2009]</td>
</tr>
<tr>
<td>SCIP</td>
<td>2012 Achterberg [2009], Vigerske and Gleixner [2017], Maher, Fischer, Gally,</td>
</tr>
<tr>
<td></td>
<td>Gamrath, Gleixner, Gottwald, Hendel, Koch, Lübbecke, Miltenberger, Müller,</td>
</tr>
<tr>
<td></td>
<td>Pfetsch, Puchert, Rehfeldt, Schenker, Schwarz, Serrano, Shinano, Weninger, Witt,</td>
</tr>
<tr>
<td></td>
<td>and Witzig [2017]</td>
</tr>
<tr>
<td>ANTIGONE</td>
<td>2013 Misener and Floudas [2014]</td>
</tr>
</tbody>
</table>

**Main restriction:** algebraic structure of problem must be available (see later)
## Solvers for General MINLP

### Deterministic:

<table>
<thead>
<tr>
<th>solver</th>
<th>1st ver. citation</th>
</tr>
</thead>
<tbody>
<tr>
<td>αBB</td>
<td>1995 Adjiman, Androulakis, and Floudas [1998a]</td>
</tr>
<tr>
<td>BARON</td>
<td>1996 Sahinidis [1996], Tawarmalani and Sahinidis [2005]</td>
</tr>
<tr>
<td>in gPROMS</td>
<td>1996 Smith and Pantelides [1997]</td>
</tr>
<tr>
<td>Couenne</td>
<td>2008 Belotti, Lee, Liberti, Margot, and Wächter [2009]</td>
</tr>
<tr>
<td>LindoAPI</td>
<td>2009 Lin and Schrage [2009]</td>
</tr>
<tr>
<td>SCIP</td>
<td>2012 Achterberg [2009], Vigerske and Gleixner [2017], Maher, Fischer, Gally,</td>
</tr>
<tr>
<td></td>
<td>Gamrath, Gleixner, Gottwald, Hendel, Koch, Lübbecke, Miltenberger, Müller,</td>
</tr>
<tr>
<td></td>
<td>Pfetsch, Puchert, Rehfeldt, Schenker, Schwarz, Serrano, Shinano, Weninger, Witt,</td>
</tr>
<tr>
<td></td>
<td>and Witzig [2017]</td>
</tr>
<tr>
<td>ANTIGONE</td>
<td>2013 Misener and Floudas [2014]</td>
</tr>
</tbody>
</table>

**Main restriction:** algebraic structure of problem must be available (see later)

**Interval-Arithmetic** based: avoid round-off errors, typically NLP only, e.g.,

- COCONUT [Neumaier, 2001], Ibex, . . .

**Stochastic search:** LocalSolver, OQNLP [Ugray, Lasdon, Plummer, Glover, Kelly, and Martí, 2007], . . .
Global MINLP Solver Progress: # Solved Instances and Solving Time

- 71 “non-trivial solvable” instances from MINLPLib
- time limit: 1800 seconds, gap limit: 1e-6

Overall speedup (virtual best solver): **15.12**
Fundamental Methods
Fundamental Methods

Recap: Mixed-Integer Linear Programming
For mixed-integer linear programs (MIP), that is,
\[
\min c^T x, \\
\text{s.t. } Ax \leq b, \\
x_i \in \mathbb{Z}, \quad i \in I,
\]
the dominant method of \textbf{Branch & Cut} combines cutting planes \cite{Gomory1958} & branch-and-bound \cite{LandDoig1960}.
Fundamental Methods

Convex MINLP
NLP-based Branch & Bound (NLP-BB)

MIP branch-and-bound
[Land and Doig, 1960]

**Bounding:** Solve convex NLP relaxation obtained by dropping integrality requirements.

**Branching:** Subdivide problem along variables $x_i$, $i \in \mathcal{I}$, that take fractional value in NLP solution.

MINLP branch-and-bound
[Leyffer, 1993]
NLP-based Branch & Bound (NLP-BB)

**MIP branch-and-bound**
[Land and Doig, 1960]

**MINLP branch-and-bound**
[Leyffer, 1993]

**Bounding**: Solve convex NLP relaxation obtained by dropping integrality requirements.

**Branching**: Subdivide problem along variables $x_i, i \in \mathcal{I}$, that take fractional value in NLP solution.

- However: Robustness and Warmstarting-capability of NLP solvers not as good as for LP solvers (simplex alg.)
Reduce Convex MINLP to MIP

Assume all functions $g_k(\cdot)$ of MINLP are convex on $[\ell, u]$.

Duran and Grossmann [1986]: MINLP and the following MIP have the same optimal solutions

$$\begin{align*}
\min & \quad c^T x, \\
\text{s.t.} & \quad g_k(\hat{x}) + \nabla g_k(\hat{x})^T (x - \hat{x}) \leq 0, \\
& \quad k \in [m], \quad \hat{x} \in R, \\
& \quad x_i \in \mathbb{Z}, \quad i \in I, \\
& \quad x \in [\ell, u],
\end{align*}$$

where $\hat{x} \in R$ are the solutions of the NLP subproblems obtained from MINLP by applying any possible fixing for $x_I$, i.e.,

$$\begin{align*}
\min & \quad c^T x \text{ s.t. } g(x) \leq 0, x \in [\ell, u], x_I \text{ fixed}.
\end{align*}$$

Example:

$$\begin{align*}
\min & \quad x + y \\
\text{s.t.} & \quad (x, y) \in \text{ellipsoid} \\
& \quad x \in \{0, 1, 2, 3\} \\
& \quad y \in [0, 3]
\end{align*}$$
Reduce Convex MINLP to MIP

Assume all functions $g_k(\cdot)$ of MINLP are convex on $[\ell, u]$.

Duran and Grossmann [1986]: MINLP and the following MIP have the same optimal solutions

$$\min c^T x,$$
$$\text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^T (x - \hat{x}) \leq 0,$$
$$k \in [m], \quad \hat{x} \in R,$$
$$x_i \in \mathbb{Z}, \quad i \in I,$$
$$x \in [\ell, u],$$

where $\hat{x} \in R$ are the solutions of the NLP subproblems obtained from MINLP by applying any possible fixing for $x_I$, i.e.,

$$\min c^T x \text{ s.t. } g(x) \leq 0, \quad x \in [\ell, u], \quad x_I \text{ fixed}.$$
Assume all functions $g_k(\cdot)$ of MINLP are convex on $[\ell, u]$.

Duran and Grossmann [1986]: MINLP and the following MIP have the same optimal solutions

$$
\begin{align*}
\min & \quad c^T x, \\
\text{s.t.} & \quad g_k(\hat{x}) + \nabla g_k(\hat{x})^T (x - \hat{x}) \leq 0, \\
& \quad k \in [m], \quad \hat{x} \in R, \\
& \quad x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\
& \quad x \in [\ell, u],
\end{align*}
$$

where $\hat{x} \in R$ are the solutions of the NLP subproblems obtained from MINLP by applying any possible fixing for $x_\mathcal{I}$, i.e.,

$$
\min c^T x \text{ s.t. } g(x) \leq 0, \quad x \in [\ell, u], \quad x_i \text{ fixed.}
$$

Example:

$$
\begin{align*}
\min \quad x + y \\
\text{s.t.} & \quad (x, y) \in \text{ellipsoid} \\
& \quad x \in \{0, 1, 2, 3\} \\
& \quad y \in [0, 3]
\end{align*}
$$
Assume all functions $g_k(\cdot)$ of MINLP are convex on $[\ell, u]$.

Duran and Grossmann [1986]: MINLP and the following MIP have the same optimal solutions

$$\begin{align*}
\min & \quad c^T x, \\
\text{s.t.} & \quad g_k(\hat{x}) + \nabla g_k(\hat{x})^T (x - \hat{x}) \leq 0, \\
& \quad k \in [m], \quad \hat{x} \in R, \\
& \quad x_i \in \mathbb{Z}, \quad i \in I, \\
& \quad x \in [\ell, u],
\end{align*}$$

where $\hat{x} \in R$ are the solutions of the NLP subproblems obtained from MINLP by applying any possible fixing for $x_I$, i.e.,

$$\min c^T x \text{ s.t. } g(x) \leq 0, \ x \in [\ell, u], \ x_I \text{ fixed.}$$

Example:

$$\begin{align*}
\min & \quad x + y \\
\text{s.t.} & \quad (x, y) \in \text{ellipsoid} \\
& \quad x \in \{0, 1, 2, 3\} \\
& \quad y \in [0, 3]
\end{align*}$$
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Outer Approximation (OA) algorithm

[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by alternatively solving MIP relaxations and NLP subproblems until MIP solution is feasible for MINLP.
Convex MINLP

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

MIP

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

Outer Approximation (OA) algorithm

[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by alternatively solving MIP relaxations and NLP subproblems until MIP solution is feasible for MINLP.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Outer Approximation (OA) algorithm

[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by alternatively solving MIP relaxations and NLP subproblems until MIP solution is feasible for MINLP.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Outer Approximation (OA) algorithm
[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by alternatively solving MIP relaxations and NLP subproblems until MIP solution is feasible for MINLP.
Convex MINLP

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

MIP

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

**Outer Approximation (OA) algorithm**

[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by **alternatively solving MIP relaxations and NLP subproblems** until MIP solution is feasible for MINLP.
Convex MINLP

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

MIP

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

Outer Approximation (OA) algorithm

[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by alternatively solving MIP relaxations and NLP subproblems until MIP solution is feasible for MINLP.
Convex MINLP

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

MIP

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

Outer Approximation (OA) algorithm

[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by alternatively solving MIP relaxations and NLP subproblems until MIP solution is feasible for MINLP.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Outer Approximation (OA) algorithm

[Duran and Grossmann, 1986]:

- Start with \( R := \emptyset \).
- Dynamically increase \( R \) by alternatively solving MIP relaxations and NLP subproblems until MIP solution is feasible for MINLP.
Convex MINLP

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

MIP

\[
\min c^T x, \\
\text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^T (x - \hat{x}) \leq 0, \quad \forall k \in [m], \quad \hat{x} \in \mathbb{R}, \\
x_i \in \mathbb{Z}, \quad \forall i \in \mathcal{I}, \\
x_i \in [\ell_i, u_i], \quad \forall i \in [n]
\]

**Extended Cutting Plane Method (ECP)**

[Kelley, 1960, Westerlund and Pettersson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Extended Cutting Plane Method (ECP)
[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
**Outer Approximation Method (OA), ECP, EHP**

**Convex MINLP**

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

**MIP**

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

**Extended Cutting Plane Method (ECP)**

[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
**Outer Approximation Method (OA), ECP, EHP**

**Convex MINLP**

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

**MIP**

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

**Extended Cutting Plane Method (ECP)**

[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
Convex MINLP

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

MIP

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

Extended Cutting Plane Method (ECP) 
[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Extended Cutting Plane Method (ECP) [Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize $g_k(\cdot)$ in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
Convex MINLP

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

MIP

\[
\min c^T x, \\
\text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^T(x - \hat{x}) \leq 0, \\
\quad \forall k \in [m], \quad \hat{x} \in R, \\
x_i \in \mathbb{Z}, \quad \forall i \in I, \\
x_i \in [\ell_i, u_i], \quad \forall i \in [n]
\]

Extended Cutting Plane Method (ECP)

[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
**Outer Approximation Method (OA), ECP, EHP**

**Convex MINLP**

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

**MIP**

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

**Extended Cutting Plane Method (ECP)**

[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve **MIP relaxation only**.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but **weaker MIP relaxation**.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Extended Cutting Plane Method (ECP)

[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Extended Cutting Plane Method (ECP)

[Kelley, 1960, Westerlund and Petterson, 1995]:

- Iteratively solve MIP relaxation only.
- Linearize \( g_k(\cdot) \) in MIP relaxation.
- No need to solve NLP, but weaker MIP relaxation.
Convex MINLP

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

MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set \( \{x \in [\ell, u] : g_k(x) \leq 0\} \) via linesearch.
- Linearize \( g_k(\cdot) \) in improved reference point.
- No need to solve NLP, but stronger MIP than ECP.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set
  \( \{x \in [\ell, u] : g_k(x) \leq 0\} \) via linesearch.
- Linearize \( g_k(\cdot) \) in improved reference point.
- No need to solve NLP, but stronger MIP than ECP.
Outer Approximation Method (OA), ECP, EHP

### Convex MINLP

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

### MIP

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

### Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- Iteratively solve **MIP relaxation only**.
- **Move MIP solution** onto NLP-feasible set \( \{ x \in [\ell, u] : g_k(x) \leq 0 \} \) via linesearch.
- Linearize \( g_k(\cdot) \) in improved reference point.
- No need to solve NLP, but **stronger MIP** than ECP.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set \( \{x \in [\ell, u] : g_k(x) \leq 0\} \) via linesearch.
- Linearize \( g_k(\cdot) \) in improved reference point.
- No need to solve NLP, but stronger MIP than ECP.
Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

MIP

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

Extended Hyperplane Method (EHP)
[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set \( \{x \in [\ell, u] : g_k(x) \leq 0\} \) via linesearch.
- Linearize \( g_k(\cdot) \) in improved reference point.
- No need to solve NLP, but stronger MIP than ECP.
**Convex MINLP**

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

**MIP**

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

**Extended Hyperplane Method (EHP)**

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- **Iteratively solve MIP relaxation only.**
- **Move MIP solution** onto NLP-feasible set \( \{x \in [\ell, u] : g_k(x) \leq 0\} \) via linesearch.
- **Linearize** \( g_k(\cdot) \) in improved reference point.
- **No need to solve NLP, but stronger MIP than ECP.**
Convex MINLP

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

MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set \( \{x \in [\ell, u] : g_k(x) \leq 0\} \) via linesearch.
- Linearize \( g_k(\cdot) \) in improved reference point.
- No need to solve NLP, but stronger MIP than ECP.
Convex MINLP

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

MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016]:

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set \(\{x \in [\ell, u] : g_k(x) \leq 0\}\) via linesearch.
- Linearize \(g_k(\cdot)\) in improved reference point.
- No need to solve NLP, but stronger MIP than ECP.
LP/NLP- or LP-based Branch & Bound

**OA/ECP/EHP**: Solving a sequence of MIP relaxations can be expensive and wasteful (no warmstarts)
**LP/NLP- or LP-based Branch & Bound**

**OA/ECP/EHP:** Solving a sequence of MIP relaxations can be expensive and wasteful (no warmstarts)

**LP/NLP-based Branch & Bound** [Quesada and Grossmann, 1992]:

- Integrate NLP-solves into MIP Branch & Bound.
- When LP relaxation is integer feasible, solve **NLP subproblem** (as in OA).
- Add linearization in NLP solution to LP relaxation and resolve LP.
LP/NLP- or LP-based Branch & Bound

**OA/ECP/EHP**: Solving a sequence of MIP relaxations can be expensive and wasteful (no warmstarts)

**LP/NLP-based Branch & Bound** [Quesada and Grossmann, 1992]:

- Integrate NLP-solves into MIP Branch & Bound.
- When LP relaxation is integer feasible, solve NLP subproblem (as in OA).
- Add linearization in NLP solution to LP relaxation and resolve LP.

**LP-based Branch & Bound:**

- Integrate Kelley’ Cutting Plane method into MIP Branch & Bound.
- Add linearization in LP solution to LP relaxation (as in ECP).
- Optional: Move LP solution onto NLP-feasible set \( \{x \in [\ell, u] : g_k(x) \leq 0\} \) via linesearch (as in EHP) [Maher, Fischer, Gally, Gamrath, Gleixner, Gottwald, Hendel, Koch, Lübbecke, Miltenberger, Müller, Pfetsch, Puchert, Reihefdt, Schenker, Schwarz, Serrano, Shinano, Weninger, Witt, and Witzig, 2017].
Fundamental Methods

Nonconvex MINLP
Nonconvex MINLP

**Now:** Let $g_k(\cdot)$ be nonconvex for some $k \in [m]$.

**Outer-Approximation:**
- Linearizations
  $$g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq 0$$
  may not be valid.

**NLP-based Branch & Bound:**
- Solving nonconvex NLP relaxation to global optimality can be as hard as original problem.
Nonconvex MINLP

Now: Let $g_k(\cdot)$ be nonconvex for some $k \in [m]$.

Outer-Approximation:

- Linearizations
  $$g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq 0$$
  may not be valid.

- Heuristics: add cuts as "soft-constraints"
  $$\min_{\alpha \geq 0} \alpha \text{ s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq \alpha$$

NLP-based Branch & Bound:

- Solving nonconvex NLP relaxation to global optimality can be as hard as original problem.
- Heuristic: Solve NLPs locally from multiple starting points.
Nonconvex MINLP

Now: Let $g_k(\cdot)$ be nonconvex for some $k \in [m]$.

Outer-Approximation:

- Linearizations
  \[ g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq 0 \]
  may not be valid.

- Heuristics: add cuts as “soft-constraints”
  \[ \min_{\alpha \geq 0} \alpha \text{ s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq \alpha \]

NLP-based Branch & Bound:

- Solving nonconvex NLP relaxation to global optimality can be as hard as original problem.
- Heuristic: Solve NLPs locally from multiple starting points.

Exact approach: Spatial Branch & Bound:

- Relax nonconvexity to obtain a tractable relaxation (LP or convex NLP).
- Branch on “nonconvexities” to enforce original constraints.
Nonconvex MINLP

Now: Let $g_k(\cdot)$ be nonconvex for some $k \in [m]$.

**Outer-Approximation:**

- Linearizations
  \[ g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq 0 \]
  may not be valid.

- Heuristics: add cuts as “soft-constraints”
  \[ \min_{\alpha \geq 0} \alpha \text{ s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq \alpha \]

**NLP-based Branch & Bound:**

- Solving nonconvex NLP relaxation to global optimality can be as hard as original problem.

- Heuristic: Solve NLPs locally from multiple starting points.

**Exact approach: Spatial Branch & Bound:**

- **Relax nonconvexity** to obtain a tractable relaxation (LP or convex NLP).

- **Branch on “nonconvexities”** to enforce original constraints.
Convex Relaxation

**Given:** \( X = \{ x \in [\ell, u] : g_k(x) \leq 0, k \in [m] \} \) (continuous relaxation of MINLP)

**Seek:** \( \text{conv}(X) \) – convex hull of \( X \)
Convex Relaxation

Given: \( X = \{ x \in \ell, u : g_k(x) \leq 0, k \in [m] \} \) (continuous relaxation of MINLP)

Seek: \( \text{conv}(X) \) – convex hull of \( X \)

- In practice, \( \text{conv}(X) \) is impossible to construct explicitly.
Convex Relaxation

**Given:** \( X = \{ x \in [\ell, u] : g_k(x) \leq 0, k \in [m] \} \) (continuous relaxation of MINLP)

**Seek:** \( \text{conv}(X) \) – convex hull of \( X \)

- In practice, \( \text{conv}(X) \) is impossible to construct explicitly.

**Relax I:** Convexify the feasible sets that are defined by each constraint individually, i.e.,

\[
\bigcap_{k \in [m]} \text{conv}\{x \in [\ell, u] : g_k(x) \leq 0\}
\]
Convex Relaxation

**Given:** \( X = \{ x \in [\ell, u] : g_k(x) \leq 0, k \in [m] \} \) (continuous relaxation of MINLP)

**Seek:** \( \text{conv}(X) \) – convex hull of \( X \)

- In practice, \( \text{conv}(X) \) is impossible to construct explicitly.

**Relax I:** Convexify the feasible sets that are defined by each constraint individually, i.e.,

\[
\bigcap_{k \in [m]} \text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \}
\]

- In practice, \( \text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \} \) is impossible to construct explicitly in general – but possible for certain cases.
Convex Relaxation

**Given:** $X = \{ x \in [\ell, u] : g_k(x) \leq 0, k \in [m] \}$ (continuous relaxation of MINLP)

**Seek:** $\text{conv}(X)$ – convex hull of $X$

- In practice, $\text{conv}(X)$ is **impossible** to construct explicitly.

**Relax I:** Convexify the feasible sets that are defined by each constraint individually, i.e.,

$$\bigcap_{k \in [m]} \text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \}$$

- In practice, $\text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \}$ is **impossible** to construct explicitly in general – but possible for certain cases.

**Relax II:** Convexify each nonconvex function $g_k(\cdot)$ individually, i.e.,

$$\{ x \in [\ell, u] : \text{"conv}(g_k)''(x) \leq 0 \}$$
Convex Relaxation

**Given:** $X = \{ x \in [\ell, u] : g_k(x) \leq 0, k \in [m] \}$ (continuous relaxation of MINLP)

**Seek:** $\text{conv}(X) = \text{convex hull of } X$

- In practice, $\text{conv}(X)$ is impossible to construct explicitly.

**Relax I:** Convexify the feasible sets that are defined by each constraint individually, i.e.,

$$\bigcap_{k \in [m]} \text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \}$$

- In practice, $\text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \}$ is impossible to construct explicitly in general – but possible for certain cases.

**Relax II:** Convexify each nonconvex function $g_k(\cdot)$ individually, i.e.,

$$\{ x \in [\ell, u] : \left. \text{conv}(g_k)''(x) \right| \leq 0 \}$$

- In practice, convex envelope is not known explicitly in general.
Convex Relaxation

**Given:** \[ X = \{ x \in [\ell, u] : g_k(x) \leq 0, \; k \in [m] \} \text{ (continuous relaxation of MINLP)} \]

**Seek:** \( \text{conv}(X) \) – convex hull of \( X \)

- In practice, \( \text{conv}(X) \) is impossible to construct explicitly.

**Relax I:** Convexify the feasible sets that are defined by each constraint individually, i.e.,
\[
\bigcap_{k \in [m]} \text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \}
\]

- In practice, \( \text{conv}\{ x \in [\ell, u] : g_k(x) \leq 0 \} \) is impossible to construct explicitly in general – but possible for certain cases.

**Relax II:** Convexify each nonconvex function \( g_k(\cdot) \) individually, i.e.,
\[
\{ x \in [\ell, u] : \text{“conv}(g_k)’’(x) \leq 0 \}
\]

- In practice, convex envelope is not known explicitly in general – except for many “simple functions”
Convex Envelopes for “simple” functions

concave functions

\[ x^k \quad (k \in 2\mathbb{Z} + 1) \]

\[ x \cdot y \quad (0 < y < \infty) \]

\[ x^2 \cdot y^2 \]

\[ -\sqrt{x} \cdot y^2 \]

\[ x/y \]
Factorable Functions [McCormick, 1976]

$g(x)$ is factorable if it can be expressed as a combination of functions from a finite set of operators, e.g., $\{+, \times, \div, \wedge, \sin, \cos, \exp, \log, \mid \cdot \mid \}$, whose arguments are variables, constants, or other factorable functions.

- Typically represented as expression trees or graphs (DAG).
- Excludes integrals $x \mapsto \int_{x_0}^{x} h(\zeta) d\zeta$ and black-box functions.

Example:

$$x_1 \log(x_2) + x_2^3$$
Reformulation of Factorable MINLP

Smith and Pantelides [1996, 1997]: By introducing new variables and equations, every factorable MINLP can be reformulated such that for every constraint function the convex envelope is known.

\[
x_1 \log(x_2) + x_2^3 \leq 0
\]
\[
x_1 \in [1, 2], \ x_2 \in [1, e]
\]
\[
\Rightarrow
\]
\[
y_1 + y_2 \leq 0
\]
\[
x_1 y_3 = y_1
\]
\[
x_3 = y_2
\]
\[
x_2 = y_3
\]
\[
\log(x_2) = y_3
\]
\[
x_1 \in [1, 2], \ x_2 \in [1, e]
\]
\[
y_1 \in [0, 2], \ y_2 \in [1, e^3]
\]
\[
y_3 \in [0, 1]
\]

- Bounds for new variables inherited from functions and their arguments, e.g., \(y_3 \in \log([1, e]) = [0, 1]\).
- Reformulation may not be unique, e.g., \(xyz = (xy)z = x(yz)\).
Smith and Pantelides [1996, 1997]: By introducing new variables and equations, every factorable MINLP can be reformulated such that for every constraint function the convex envelope is known.

\[ x_1 \log(x_2) + x_2^3 \leq 0 \]
\[ x_1 \in [1, 2], \ x_2 \in [1, e] \]

\[ \Rightarrow \]
\[ y_1 + y_2 \leq 0 \]
\[ x_1 y_3 = y_1 \]
\[ x_2^3 = y_2 \]
\[ \log(x_2) = y_3 \]
\[ x_1 \in [1, 2], \ x_2 \in [1, e] \]
\[ y_1 \in [0, 2], \ y_2 \in [1, e^3] \]
\[ y_3 \in [0, 1] \]

\[ \Rightarrow \]
\[ \text{Convex Relax} \]
\[ y_1 + y_2 \leq 0 \]
\[ 2y_3 + x_1 - 2 \leq y_1 \]
\[ y_3 \leq y_1 \]
\[ y_1 \leq 2y_3 \]
\[ y_1 \leq y_3 + x - 1 \]
\[ x_2^3 \leq y_2 \]
\[ y_2 \leq 1 + \frac{e^3 - 1}{e - 1} (x_2 - 1) \]
\[ \frac{1}{e - 1} (x_2 - 1) \leq y_3 \]
\[ y_3 \leq \log(x_2) \]
\[ x_1 \in [1, 2], \ x_2 \in [1, e] \]
\[ y_1 \in [0, 2], \ y_2 \in [1, e^3], \ y_3 \in [0, 1] \]

- Bounds for new variables inherited from functions and their arguments, e.g., \( y_3 \in \log([1, e]) = [0, 1] \).
- Reformulation may not be unique, e.g., \( xyz = (xy)z = x(yz) \).

17/42
The type of algebraic expressions that is understood and not broken up further is implementation specific.

Thus, not all functions are supported by any deterministic solver, e.g.,

- ANTIGONE, BARON, and SCIP do not support trigonometric functions.
- Couenne does not support max or min.
- No deterministic global solver supports external functions that are given by routines for point-wise evaluation of function and derivatives.

Example ANTIGONE [Misener and Floudas, 2014]:
Recall **Spatial Branch & Bound**:

- **Relax nonconvexity** to obtain a **tractable relaxation** (often an LP).
- **Branch on “nonconvexities”** to enforce original constraints.
Spatial Branching

Recall Spatial Branch & Bound:

✓ Relax nonconvexity to obtain a tractable relaxation (often an LP).
• Branch on “nonconvexities” to enforce original constraints.

The variable bounds determine the convex relaxation, e.g., for the constraint

\[ y = x^2, \quad x \in [\ell, u], \]

the convex relaxation is

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

Recall Spatial Branch & Bound:

- Relax nonconvexity to obtain a tractable relaxation (often an LP).
- Branch on “nonconvexities” to enforce original constraints.

The variable bounds determine the convex relaxation, e.g., for the constraint

\[ y = x^2, \quad x \in [\ell, u], \]

the convex relaxation is

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

Thus, branching on a nonlinear variable in a nonconvex term allows for tighter relaxations in sub-problems:
Fundamental Methods

Bound Tightening
Variable Bounds Tightening (Domain Propagation)

Tighten variable bounds \([\ell, u]\) such that

- the optimal value of the problem is not changed, or
- the set of optimal solutions is not changed, or
- the set of feasible solutions is not changed.

Formally:

\[
\min / \max \{x_k : x \in \mathcal{R}\}, \quad k \in [n],
\]

where \(\mathcal{R} = \{x \in [\ell, u] : g(x) \leq 0, x_i \in \mathbb{Z}, i \in I\}\) (MINLP-feasible set) or a relaxation thereof.

Bound tightening can tighten the LP relaxation without branching.

Belotti, Lee, Liberti, Margot, and Wächter [2009]: overview on bound tightening for MINLP
Feasibility-Based Bound Tightening

**Feasibility-based Bound Tightening (FBBT):**
Deduce variable bounds from single constraint and box $[\ell, u]$, that is
$$\mathcal{R} = \{x \in [\ell, u] : g_j(x) \leq 0\}$$ for some fixed $j \in [m]$.

- cheap and effective $\Rightarrow$ used for “probing”
Feasibility-Based Bound Tightening

Feasibility-based Bound Tightening (FBBT):
Deduce variable bounds from single constraint and box $[\ell, u]$, that is
\[ \mathcal{R} = \{ x \in [\ell, u] : g_j(x) \leq 0 \} \quad \text{for some fixed } j \in [m]. \]

- cheap and effective $\Rightarrow$ used for “probing”

**Linear Constraints:**

\[
b \leq \sum_{i:a_i>0} a_ix_i + \sum_{i:a_i<0} a_ix_i \leq c, \quad \ell \leq x \leq u
\]

\[
\Rightarrow \quad \frac{1}{a_j} x_j \leq \begin{cases} 
\frac{c}{a_j} - \sum_{i:a_i>0,i\neq j} a_i\ell_i - \sum_{i:a_i<0} a_iu_i, & \text{if } a_j > 0 \\
\frac{b}{a_j} - \sum_{i:a_i>0} a_iu_i - \sum_{i:a_i<0,i\neq j} a_i\ell_i, & \text{if } a_j < 0 
\end{cases}
\]

\[
\frac{1}{a_j} x_j \geq \begin{cases} 
\frac{b}{a_j} - \sum_{i:a_i>0,i\neq j} a_iu_i - \sum_{i:a_i<0} a_i\ell_i, & \text{if } a_j > 0 \\
\frac{c}{a_j} - \sum_{i:a_i>0} a_i\ell_i - \sum_{i:a_i<0,i\neq j} a_iu_i, & \text{if } a_j < 0 
\end{cases}
\]

- Belotti, Cafieri, Lee, and Liberti [2010]: fixed point of iterating FBBT on set of linear constraints can be computed by solving one LP
- Belotti [2013]: FBBT on two linear constraints simultaneously
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \\
x, y \in [1, 9]
\]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]
\[ x, y \in [1, 9] \]

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

\[ [1, 9] \times [1, 9] = [1, 81] \]

Application of Interval Arithmetics

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]

\[x, y \in [1, 9]\]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

\[
\sqrt{1, 9} = [1, 3] \quad \sqrt{1, 81} = [1, 9]
\]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [\infty, 7] \]
\[ x, y \in [1, 9] \]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

\[ [1, 3] + 2 [1, 9] + 2 [1, 3] = [5, 18] \]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]
\[
x, y \in [1, 9]
\]

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

[5, 7] − 2 [1, 9] − 2 [1, 3] = [−19, 3]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

**Example:**

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]

\[ x, y \in [1, 9] \]

**Forward propagation:**

- compute bounds on intermediate nodes (bottom-up)

**Backward propagation:**

- reduce bounds using reverse operations (top-down)

\[ ([5, 7] - [1, 3] - 2 [1, 3]) / 2 = [-2, 2] \]

Application of **Interval Arithmetics**

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]

\[x, y \in [1, 9]\]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

Backward propagation:
- reduce bounds using reverse operations (top-down)

\[
\frac{([5, 7] - [1, 3] - 2 [1, 2])}{2} = [-1, 2]
\]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:
\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]
\[x, y \in [1, 9]\]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

Backward propagation:
- reduce bounds using reverse operations (top-down)

\[[1, 2]^2 = [1, 4]\]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

**Example:**

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]

\(x, y \in [1, 9]\)

**Forward propagation:**

- compute bounds on intermediate nodes (bottom-up)

**Backward propagation:**

- reduce bounds using reverse operations (top-down)

\[
[1, 3]^2 = [1, 9] \quad [1, 4]/[1, 9] = [1/9, 4]
\]

Application of **Interval Arithmetics**

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]

\(x, y \in [1, 9]\)

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

\[\begin{align*}
[1, 2]^2 &= [1, 4] \\
[1, 4]/[1, 4] &= [1/4, 4]
\end{align*}\]

Application of Interval Arithmetics

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]
\[ x, y \in [1, 4] \]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

Backward propagation:
- reduce bounds using reverse operations (top-down)

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:
\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]
\[
x, y \in [1, 4]
\]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

Backward propagation:
- reduce bounds using reverse operations (top-down)

\[
\sqrt{[1, 4]} = [1, 2] \quad \sqrt{[1, 16]} = [1, 4]
\]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [\infty, 7] \]

\[ x, y \in [1, 4] \]

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

\[ [1, 2] + 2 [1, 4] + 2 [1, 2] = [5, 14] \]

Application of Interval Arithmetics

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]

\[ x, y \in [1, 4] \]

**Forward propagation:**

- compute bounds on intermediate nodes (bottom-up)

**Backward propagation:**

- reduce bounds using reverse operations (top-down)


Application of **Interval Arithmetics**

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]
\[ x, y \in [1, 4] \]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

Backward propagation:
- reduce bounds using reverse operations (top-down)

\[ ([5, 7] - [1, 2] - 2 [1, 2]) / 2 = [-0.5, 2] \]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]

\[ x, y \in [1, 4] \]

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

\[ ([5, 7] - [1, 2] - 2[1, 4]) / 2 = [-2.5, 2] \]

Application of Interval Arithmetics

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]

\[
x, y \in [1, 4]
\]

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

Application of Interval Arithmetics

[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:

\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]
\[ x, y \in [1, 4] \]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

Backward propagation:
- reduce bounds using reverse operations (top-down)

[1, 2]^2 = [1, 4]  \quad [1, 4]/[1, 4] = [1/4, 4]

Application of Interval Arithmetics
[Moore, 1966]
Feasibility-Based Bound Tightening on Expression “Tree”

Example:
\[ \sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7] \]
\[ x, y \in [1, 4] \]

Forward propagation:
- compute bounds on intermediate nodes (bottom-up)

Backward propagation:
- reduce bounds using reverse operations (top-down)

\[ [1, 2]^2 = [1, 4] \quad [1, 4]/[1, 4] = [1/4, 4] \]

Application of Interval Arithmetics
[Moore, 1966]
Problem: Overestimation
Example – reformulated:

\[ x' = \sqrt{x}, \quad y' = \sqrt{y} \]

\[ x' + 2x'y' + 2y' \in [-\infty, 7] \]

\[ x', y' \in [1, 3] \]

Simple FBBT:

\[ x' \leq 7 - 2x'y' - 2y' \]

\[ x' \leq (7 - x' - 2y')/(2y') \]
Example – reformulated:

\[
(x' = \sqrt{x}, \ y' = \sqrt{y})
\]

\[
x' + 2x'y' + 2y' \in [-\infty, 7]
\]

\[
x', y' \in [1, 3]
\]

Simple FBBT:

\[
x' \leq 7 - 2x'y' - 2y'
\]

\[
\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1
\]

\[
x' \leq (7 - x' - 2y')/(2y')
\]

\[
\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1)
\]
Example – reformulated:

\[(x' = \sqrt{x}, y' = \sqrt{y})\]

\[x' + 2x'y' + 2y' \in [-\infty, 7]\]

\[x', y' \in [1, 3]\]

Simple FBBT:

\[x' \leq 7 - 2x'y' - 2y'\]

\[\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = 3\]

\[x' \leq (7 - x' - 2y')/(2y')\]

\[\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = 2\]
**Example – reformulated:**

\[(x' = \sqrt{x}, \ y' = \sqrt{y})\]

\[x' + 2x'y' + 2y' \in [-\infty, 7]\]

\[x', y' \in [1, 3]\]

---

**Simple FBBT:**

\[x' \leq 7 - 2x'y' - 2y'\]
\[\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = 3\]

\[x' \leq (7 - x' - 2y')/(2y')\]
\[\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = 2\]

\[y' \leq (7 - 2x'y' - x')/2 \leq 2\]

\[y' \leq (7 - 2y' - x')/(2x') \leq 2\]
Example – reformulated:

\[(x' = \sqrt{x}, \ y' = \sqrt{y})\]

\[x' + 2x'y' + 2y' \in [-\infty, 7]\]

\[x', y' \in [1, 3]\]

Simple FBBT:

\[x' \leq 7 - 2x'y' - 2y'\]

\[\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = 3\]

\[x' \leq (7 - x' - 2y')/(2y')\]

\[\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = 2\]

\[y' \leq (7 - 2x'y' - x')/2 \leq 2\]

\[y' \leq (7 - 2y' - x')/(2x') \leq 2\]
**Example – reformulated:**

\[(x' = \sqrt{x}, \ y' = \sqrt{y})\]

\[x' + 2x'y' + 2y' \in [-\infty, 7]\]

\[x', y' \in [1, 3]\]

**Simple FBBT:**

\[x' \leq 7 - 2x'y' - 2y'\]

\[\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = 3\]

\[x' \leq (7 - x' - 2y')/(2y')\]

\[\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = 2\]

\[y' \leq (7 - 2x'y' - x')/2 \leq 2\]

\[y' \leq (7 - 2y' - x')/(2x') \leq 2\]

**Consider Bivariate Quadratic as one term**

[Vigerske, 2013, Vigerske and Gleixner, 2017]:

\[x' \leq \frac{7 - 2y'}{1 + 2y'}\]

\[y' \leq \frac{7 - x'}{2 + 2x'}\]
Example – reformulated:

\((x' = \sqrt{x}, y' = \sqrt{y})\)

\[ x' + 2x'y' + 2y' \in [-\infty, 7] \]

\[ x', y' \in [1, 3] \]

Simple FBBT:

\[
\begin{align*}
x' & \leq 7 - 2x'y' - 2y' \\
& \leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = 3 \\
x' & \leq (7 - x' - 2y')/(2y') \\
& \leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = 2 \\
y' & \leq (7 - 2x'y' - x')/2 \leq 2 \\
y' & \leq (7 - 2y' - x')/(2x') \leq 2
\end{align*}
\]

Consider Bivariate Quadratic as one term

[Vigerske, 2013, Vigerske and Gleixner, 2017]:

\[
\begin{align*}
x' & \leq \frac{7 - 2y'}{1 + 2y'} \\
y' & \leq \frac{7 - x'}{2 + 2x'}
\end{align*}
\]
Example – reformulated:

\[(x' = \sqrt{x}, \ y' = \sqrt{y})\]

\[x' + 2x'y' + 2y' \in [-\infty, 7]\]

\[x', y' \in [1, 3]\]

Simple FBBT:

\[x' \leq 7 - 2x'y' - 2y'\]

\[\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = 3\]

\[x' \leq (7 - x' - 2y')/(2y')\]

\[\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = 2\]

\[y' \leq (7 - 2x'y' - x')/2 \leq 2\]

\[y' \leq (7 - 2y' - x')/(2x') \leq 2\]

Consider Bivariate Quadratic as one term

[Vigerske, 2013, Vigerske and Gleixner, 2017]:

\[x' \leq \frac{7 - 2y'}{1 + 2y'} \leq \frac{7 - 2 \cdot 1}{1 + 2 \cdot 1} = 5\]

\[x' \leq \frac{7 - x'}{2 + 2x'} \leq \frac{7 - 1}{2 + 2 \cdot 1} = 3\]

\[y' \leq \frac{7 - x'}{2 + 2x'} \leq \frac{7 - 1}{2 + 2 \cdot 1} = 3\]
Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]

\[x, y \in [1, 9]\]

- **Common subexpressions** from different constraints may stronger boundtightening.
FBBT on Expression Graph

Example:

\[
\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]
\]

\[
x^2\sqrt{y} - 2xy + 3\sqrt{y} \in [0, 2]
\]

\[
x, y \in [1, 9]
\]

- **Common subexpressions** from different constraints may stronger bound tightening.
Acceleration – Selected Topics
Acceleration – Selected Topics

Optimization-based bound tightening
Recall: **Bound Tightening** $\equiv \min / \max \{x_k : x \in \mathcal{R}\}, \ k \in [n]$, where

$\mathcal{R} \supseteq \{x \in [\ell, u] : g(x) \leq 0, x_i \in \mathbb{Z}, i \in \mathcal{I}\}$
Recall: Bound Tightening $\equiv \min / \max \{ x_k : x \in \mathcal{R} \}$, $k \in [n]$, where
\[
\mathcal{R} \supseteq \{ x \in [\ell, u] : g(x) \leq 0, x_i \in \mathbb{Z}, i \in I \}
\]

Optimization-based Bound Tightening


- $\mathcal{R} = \{ x : Ax \leq b, c^T x \leq z^* \}$ linear relaxation (with obj. cutoff)
Optimization-based bound tightening

**Recall:** Bound Tightening $\equiv \min / \max \{x_k : x \in \mathcal{R}\}, \ k \in [n]$, where
$\mathcal{R} \supseteq \{x \in [\ell, u] : g(x) \leq 0, x_i \in \mathbb{Z}, \ i \in I\}$

**Optimization-based Bound Tightening**


- $\mathcal{R} = \{x : Ax \leq b, c^T x \leq z^*\}$ linear relaxation (with obj. cutoff)
Recall: **Bound Tightening** \( \equiv \min / \max \{ x_k : x \in R \}, \ k \in [n] \), where
\[ R \supseteq \{ x \in [\ell, u] : g(x) \leq 0, x_i \in \mathbb{Z}, i \in I \} \]

**Optimization-based Bound Tightening**


- \( R = \{ x : Ax \leq b, c^T x \leq z^* \} \) linear relaxation (with obj. cutoff)
- simple, but effective on nonconvex MINLP: relaxation depends on domains
- but: potentially many expensive LPs per node
Optimization-based bound tightening

Recall: Bound Tightening $\equiv \min / \max \{ x_k : x \in \mathcal{R} \}$, $k \in [n]$, where
\[ \mathcal{R} \supseteq \{ x \in [\ell, u] : g(x) \leq 0, x_i \in \mathbb{Z}, i \in \mathcal{I} \} \]

Optimization-based Bound Tightening


- $\mathcal{R} = \{ x : Ax \leq b, c^T x \leq z^* \}$ linear relaxation (with obj. cutoff)
- simple, but effective on nonconvex MINLP: relaxation depends on domains
- but: potentially many expensive LPs per node

Advanced implementation [Gleixner, Berthold, Müller, and Weltge, 2017]:

- solve OBBT LPs at root only, learn dual certificates $x_k \geq \sum_i r_i x_i + \mu z^* + \lambda^T b$
- propagate duality certificates during tree search ("approximate OBBT")
- greedy ordering for faster LP warmstarts, filtering of provably tight bounds
- 16\% faster (24\% on instances $\geq 100$ seconds) and less time outs
Acceleration – Selected Topics

Synergies with MIP and NLP
Many MIP techniques can be generalized for MINLP

- **MIP cutting planes applied** to LP relaxation, e.g., Gomory, Mixed-Integer Rounding, Flow Cover

Many MIP techniques can be generalized for MINLP

- **MIP cutting planes applied** to LP relaxation, e.g.,
  Gomory, Mixed-Integer Rounding, Flow Cover
- **MIP cutting planes generalized** to MINLP, e.g.,
  Disjunctive Cuts [Kilinç, Linderoth, and Luedtke, 2010,
- **MIP primal heuristics applied** to MIP relaxation;
  generates fixings and starting point for sub-NLP
- **MIP heuristics generalized** to MINLP, e.g.,
  Feasibility Pump, Large Neighborhood Search,
  NLP Diving [Bonami, Cornuéjols, Lodi, and Margot, 2009,
  Berthold, Heinz, Pfetsch, and Vigerske, 2011, Bonami and
  Gonçalves, 2012, Berthold, 2014a]
Many MIP techniques can be generalized for MINLP

- **MIP cutting planes applied** to LP relaxation, e.g., Gomory, Mixed-Integer Rounding, Flow Cover
- **MIP primal heuristics applied** to MIP relaxation; generates fixings and starting point for sub-NLP
- **Bound Tightening**
Many MIP techniques can be generalized for MINLP

- **MIP cutting planes applied** to LP relaxation, e.g., Gomory, Mixed-Integer Rounding, Flow Cover
- **MIP primal heuristics applied** to MIP relaxation; generates fixings and starting point for sub-NLP
- **Bound Tightening**
- **Symmetry detection and breaking** [Liberti, 2012, Liberti and Ostrowski, 2014]
NLP Solvers (finding local optima) are used in MINLP solver

- to find feasible points when integrality and linear constraints are satisfied
- to solve continuous relaxation in NLP-based B&B
NLP Solvers (finding local optima) are used in MINLP solver

- to find feasible points when integrality and linear constraints are satisfied
- to solve continuous relaxation in NLP-based B&B
- performance of NLP solver is problem-dependent
- some MINLP solvers interface several NLP solvers:
  - **ANTIGONE**: CONOPT, SNOPT
  - **BARON**: FilterSD, FilterSQP, GAMS/NLP (e.g., CONOPT), IPOPT, MINOS, SNOPT
  - **SCIP (next ver.)**: FilterSQP, IPOPT, WORHP

![Graph showing performance ratio and problems solved](image-url)
NLP Solvers (finding local optima) are used in MINLP solver

- to **find feasible points** when integrality and linear constraints are satisfied
- to solve **continuous relaxation** in NLP-based B&B
- performance of NLP solver is **problem-dependent**
- some MINLP solvers interface several NLP solvers:
  - **ANTIGONE:** CONOPT, SNOPT
  - **BARON:** FilterSD, FilterSQP, GAMS/NLP (e.g., CONOPT), IPOPT, MINOS, SNOPT
  - **SCIP (next ver.):** FilterSQP, IPOPT, WORHP
- strategy to select NLP solver becomes important: e.g., in SCIP, always choosing best NLP solver finds **10% more locally optimal points** and is **2–3x faster** than best single solver [Müller et al., 2017]
- **BARON** chooses according to solver performance
- “**fast fail**” on expensive NLPs, **warmstart** in B&B seem important [Müller et al., 2017, Mahajan et al., 2012]
Acceleration – Selected Topics

Convexity
Convexity Detection

Analyze the Hessian:

\[ f(x) \text{ convex on } [\ell, u] \iff \nabla^2 f(x) \succeq 0 \quad \forall x \in [\ell, u] \]

- \( f(x) \) quadratic: \( \nabla^2 f(x) \) constant \( \Rightarrow \) compute spectrum numerically
- general \( f \in C^2 \): estimate eigenvalues of Interval-Hessian [Nenov et al., 2004]
Convexity Detection

Analyze the Hessian:

\( f(x) \) convex on \([\ell, u]\) \iff \nabla^2 f(x) \succeq 0 \quad \forall x \in [\ell, u] \\

- \( f(x) \) quadratic: \( \nabla^2 f(x) \) constant \implies \text{compute spectrum numerically} \\
- general \( f \in C^2 \): estimate eigenvalues of Interval-Hessian [Nenov et al., 2004]

Analyze the Algebraic Expression:

\( f(x) \) convex \implies a \cdot f(x) \begin{cases} \text{convex, } a \geq 0 \\ \text{concave, } a \leq 0 \end{cases} \\

\( f(x), g(x) \) convex \implies f(x) + g(x) \text{ convex} \\
\( f(x) \) concave \implies \log(f(x)) \text{ concave} \\
\( f(x) = \prod_{i} x_i^{e_i}, x_i \geq 0 \implies f(x) \begin{cases} \text{convex, } e_i \leq 0 \forall i \\ \text{convex, } \exists j : e_i \leq 0 \forall i \neq j; \sum_i e_i \geq 1 \\ \text{concave, } e_i \geq 0 \forall i; \sum_i e_i \leq 1 \end{cases} \)

Consider a constraint $x^T Ax + b^T x \leq c$.

If $A$ has only one negative eigenvalue, it may be reformulated as a second-order cone constraint [Mahajan and Munson, 2010], e.g.,

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

- $\sqrt{\sum_{k=1}^{N} x_k^2}$ is a convex term that can easily be linearized
- BARON and SCIP recognize “obvious” SOCs $\left(\sum_{k=1}^{N} (\alpha_k x_k)^2 - (\alpha_{N+1} x_{N+1})^2 \leq 0\right)$

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

---

feasible region

not recognizing SOC

recognizing SOC
Acceleration – Selected Topics

Convexification
Convex Envelopes for Product Terms

**Bilinear** \( x \cdot y \) \((x \in [\ell_x, u_x], y \in [\ell_y, u_y]): \)

\[
\max \left\{ \frac{u_x y + u_y x - u_x u_y}{\ell_x y + \ell_y x - \ell_x \ell_y} \right\} \leq x \cdot y \leq \min \left\{ \frac{u_x y + \ell_y x - u_x \ell_y}{\ell_x y + u_y x - \ell_x u_y} \right\}
\]

Convex Envelopes for Product Terms

**Bilinear** \( x \cdot y \) \((x \in [\ell_x, u_x], \ y \in [\ell_y, u_y])\):

\[
\max \left\{ u_x y + u_y x - u_x u_y \right\} \leq x \cdot y \leq \min \left\{ \ell_x y + \ell_y x - \ell_x \ell_y \right\}
\]


**Trilinear** \( x \cdot y \cdot z \):

- Similar formulas by recursion, considering \((x \cdot y) \cdot z\), \(x \cdot (y \cdot z)\), and \((x \cdot z) \cdot y\)
  \(\Rightarrow\) **18 inequalities** for convex underestimator [Meyer and Floudas, 2004]
Convex Envelopes for Product Terms

**Bilinear** $x \cdot y$ \hspace{1cm} ($x \in [\ell_x, u_x]$, $y \in [\ell_y, u_y]$):

$$\max \left\{ u_x y + u_y x - u_x u_y \right\} \leq x \cdot y \leq \min \left\{ \ell_x y + \ell_y x - \ell_x \ell_y \right\}$$


**Trilinear** $x \cdot y \cdot z$:

- Similar formulas by recursion, considering $(x \cdot y) \cdot z$, $x \cdot (y \cdot z)$, and $(x \cdot z) \cdot y$ \hspace{1cm} $\Rightarrow$ **18 inequalities** for convex underestimator [Meyer and Floudas, 2004]
- if mixed signs (e.g., $\ell_x < 0 < u_x$) recursion may not provide convex envelope
- Meyer and Floudas [2004] derive the facets of the envelopes: for convex envelope, distinguish **9 cases**, each giving 5-6 linear inequalities
Convex Envelopes for Product Terms

**Bilinear** $x \cdot y$  \hspace{1cm} ($x \in [\ell_x, u_x], y \in [\ell_y, u_y]$):

$$\max \left\{ \frac{u_x y + u_y x - u_x u_y}{\ell_x y + \ell_y x - \ell_x \ell_y} \right\} \leq x \cdot y \leq \min \left\{ \frac{u_x y + \ell_y x - u_x \ell_y}{\ell_x y + u_y x - \ell_x u_y} \right\}$$


**Trilinear** $x \cdot y \cdot z$:

- Similar formulas by **recursion**, considering $(x \cdot y) \cdot z$, $x \cdot (y \cdot z)$, and $(x \cdot z) \cdot y$  
  $\Rightarrow$ **18 inequalities** for convex underestimator [Meyer and Floudas, 2004]
- if **mixed signs** (e.g., $\ell_x < 0 < u_x$) recursion may not provide convex envelope
- Meyer and Floudas [2004] derive the facets of the envelopes: for convex envelope, distinguish **9 cases**, each giving 5-6 linear inequalities

**Quadrilinear** $u \cdot v \cdot w \cdot x$:

- Cafieri, Lee, and Liberti [2010]: apply formulas for bilinear and trilinear to groupings $((u \cdot v) \cdot w) \cdot x$, $(u \cdot v) \cdot (w \cdot x)$, $(u \cdot v \cdot w) \cdot x$, $(u \cdot v) \cdot w \cdot x$ and compare strength numerically
Vertex-Polyhedral Functions

For a **vertex-polyhedral** function, the convex envelope is determined by the vertices of the box:

Given $f(\cdot)$ vertex-polyhedral over $[\ell, u] \subset \mathbb{R}^n$, value of convex envelope in $x$ is

$$\min_{\lambda \in \mathbb{R}^{2n}} \left\{ \sum_p \lambda_p f(v^p) : x = \sum_p \lambda_p v^p, \sum_p \lambda_p = 1, \lambda \geq 0 \right\} \quad (C)$$

$$= \max_{a \in \mathbb{R}^n, b \in \mathbb{R}} \left\{ a^T x + b : a^T v^p + b \leq f(v^p) \forall p \right\}, \quad (D)$$

where $\{v^p : p = 1, \ldots, 2^n\} = \text{vert}([\ell, u])$ are the vertices of the box $[\ell, u]$. 
Vertex-Polyhedral Functions

For a **vertex-polyhedral** function, the convex envelope is determined by the vertices of the box:

Given \( f(\cdot) \) vertex-polyhedral over \([\ell, u] \subset \mathbb{R}^n\), value of convex envelope in \( x \) is

\[
\min_{\lambda \in \mathbb{R}^{2^n}} \left\{ \sum_p \lambda_p f(v^p) : x = \sum_p \lambda_p v^p, \sum \lambda_p = 1, \lambda \geq 0 \right\} \tag{C}
\]

\[
= \max_{a \in \mathbb{R}^n, b \in \mathbb{R}} \left\{ a^T x + b : a^T v^p + b \leq f(v^p) \quad \forall p \right\} \tag{D}
\]

where \( \{v^p : p = 1, \ldots, 2^n\} = \text{vert}([\ell, u]) \) are the vertices of the box \([\ell, u]\).

The following function classes are vertex-polyhedral:

- **Multilinear functions**: \( f(x) = \sum_{l \in \mathcal{I}} a_l \prod_{i \in l} x_i, \ l \subseteq [n] \) [Rikun, 1997]
- **Edge-concave functions**: \( f(x) \) with \( \frac{\partial^2 f}{\partial x_i^2} \leq 0, \ i \in [n] \) [Tardella, 1988/89]
For a \textbf{vertex-polyhedral} function, the convex envelope is determined by the vertices of the box:

Given $f(\cdot)$ vertex-polyhedral over $[\ell, u] \subset \mathbb{R}^n$, value of convex envelope in $x$ is

$$\min_{\lambda \in \mathbb{R}^{2^n}} \left\{ \sum_p \lambda_p f(v^p) : x = \sum_p \lambda_p v^p, \sum \lambda_p = 1, \lambda \geq 0 \right\}$$

$$= \max_{a \in \mathbb{R}^n, b \in \mathbb{R}} \{ a^T x + b : a^T v^p + b \leq f(v^p) \forall p \}$$

where $\{v^p : p = 1, \ldots, 2^n\} = \text{vert}([\ell, u])$ are the vertices of the box $[\ell, u]$.

The following function classes are vertex-polyhedral:

- \textbf{Multilinear functions}: $f(x) = \sum_{I \in \mathcal{I}} a_I \prod_{i \in I} x_i$, $I \subseteq [n]$ [Rikun, 1997]

- \textbf{Edge-concave functions}: $f(x)$ with $\frac{\partial^2 f}{\partial x_i^2} \leq 0$, $i \in [n]$ [Tardella, 1988/89]

(C) and (D) allow to compute facets of convex envelope:

- naive: try every subset of $n + 1$ vertices: $\binom{2^{n+1}}{n}$ choices!

- Bao, Sahinidis, and Tawarmalani [2009], Meyer and Floudas [2005]:
  efficient methods for moderate $n$
Consider a function $x^T Ax + b^T x$ with $A \not\succeq 0$.

Let $\alpha \in \mathbb{R}^n$ be such that $A - \text{diag}(\alpha) \succeq 0$. Then

$$x^T Ax + b^T x + (u_x - x)^T \text{diag}(\alpha)(x - \ell_x)$$

is a convex underestimator of $x^T Ax + b^T x$ w.r.t. the box $[\ell, u]$. 

- can be generalized to twice continuously differentiable functions $g(x)$ by bounding the minimal eigenvalue of the Hessian $\nabla^2 H(x)$ for $x \in [\ell, u]$.

Consider a function $x^T Ax + b^T x$ with $A \not\succeq 0$.

Let $\alpha \in \mathbb{R}^n$ be such that $A - \text{diag}(\alpha) \succeq 0$. Then

$$x^T Ax + b^T x + (u_x - x)^T \text{diag}(\alpha)(x - \ell_x)$$

is a convex underestimator of $x^T Ax + b^T x$ w.r.t. the box $[\ell, u]$.

A simple choice for $\alpha$ is $\alpha_i = \lambda_1(A)$ (minimal eigenvalue of $A$), $i = 1, \ldots, n$. 

**α-Underestimators**
Consider a function $x^T Ax + b^T x$ with $A \not\succeq 0$.

Let $\alpha \in \mathbb{R}^n$ be such that $A - \text{diag}(\alpha) \succeq 0$. Then

$$x^T Ax + b^T x + (u_x - x)^T \text{diag}(\alpha)(x - \ell_x)$$

is a convex underestimator of $x^T Ax + b^T x$ w.r.t. the box $[\ell, u]$.

A simple choice for $\alpha$ is $\alpha_i = \lambda_1(A)$ (minimal eigenvalue of $A$), $i = 1, \ldots, n$.

- can be generalized to twice continuously differentiable functions $g(x)$ by bounding the minimal eigenvalue of the Hessian $\nabla^2 H(x)$ for $x \in [\ell_x, u_x]$ [Androulakis, Maranas, and Floudas, 1995, Adjiman and Floudas, 1996, Adjiman, Dallwig, Floudas, and Neumaier, 1998b]
Consider a function \( x^T A x + b^T x \) with \( A \not\succeq 0 \).

Let \( \alpha \in \mathbb{R}^n \) be such that \( A - \text{diag}(\alpha) \succeq 0 \). Then
\[
x^T A x + b^T x + (u_x - x)^T \text{diag}(\alpha) (x - \ell_x)
\]
is a convex underestimator of \( x^T A x + b^T x \) w.r.t. the box \([\ell, u]\).

A simple choice for \( \alpha \) is \( \alpha_i = \lambda_1(A) \) (minimal eigenvalue of \( A \)), \( i = 1, \ldots, n \).

- can be generalized to twice continuously differentiable functions \( g(x) \) by bounding the minimal eigenvalue of the Hessian \( \nabla^2 H(x) \) for \( x \in [\ell_x, u_x] \)
- underestimator is exact for \( x_i \in \{\ell_i, u_i\} \)
- thus, if \( x \) is a vector of binary variables \( (x_i^2 = x_i) \), then
\[
x^T A x + b^T x = x^T (A - \text{diag}(\alpha)) x + (b + \text{diag}(\alpha))^T x
\]
for \( x \in \{0, 1\}^n \) and \( A - \text{diag}(\alpha) \succeq 0 \). ⇒ used in CPLEX, Gurobi
Consider a function $x^T A x + b^T x$ with $A \not\succeq 0$.

- Let $\lambda_1, \ldots, \lambda_n$ be eigenvalues of $A$ and $v_1, \ldots, v_n$ be corresponding eigenvectors.

$$\Rightarrow x^T A x + b^T x + c = \sum_{i=1}^{n} \lambda_i (v_i^T x)^2 + b^T x + c.$$ (E)
Consider a function $x^T Ax + b^T x$ with $A \not\succeq 0$.

- Let $\lambda_1, \ldots, \lambda_n$ be eigenvalues of $A$ and $v_1, \ldots, v_n$ be corresponding eigenvectors.

\[
x^T Ax + b^T x + c = \sum_{i=1}^{n} \lambda_i (v_i^T x)^2 + b^T x + c.
\]

- Introducing auxiliary variables $z_i = v_i^T x$, the function becomes separable:

\[
\sum_{i=1}^{n} \lambda_i z_i^2 + b^T x + c
\]

- Underestimate concave functions $z_i \mapsto \lambda_i z_i^2$, $\lambda_i < 0$, as known.
Consider a function \( x^T A x + b^T x \) with \( A \preceq 0 \).

- Let \( \lambda_1, \ldots, \lambda_n \) be eigenvalues of \( A \) and \( v_1, \ldots, v_n \) be corresp. eigenvectors.

\[
\Rightarrow \quad x^T A x + b^T x + c = \sum_{i=1}^{n} \lambda_i (v_i^T x)^2 + b^T x + c. \tag{E}
\]

- introducing auxiliary variables \( z_i = v_i^T x \), function becomes separable:

\[
\sum_{i=1}^{n} \lambda_i z_i^2 + b^T x + c
\]

- underestimate concave functions \( z_i \mapsto \lambda_i z_i^2, \lambda_i < 0 \), as known

- one of the methods for nonconvex QP in CPLEX (keeps convex \( \lambda_i z_i^2 \) in objective and solves relaxation by QP simplex) [Bliek, Bonami, and Lodi, 2014]
Reformulation Linearization Technique (RLT)

Consider the QCQP

\[
\begin{align*}
\min & \quad x^T Q_0 x + b_0^T x \\
\text{s.t.} & \quad x^T Q_k x + b_k^T x \leq c_k \quad k = 1, \ldots, q \\
& \quad Ax \leq b \\
& \quad \ell \leq x \leq u
\end{align*}
\]

(quadraic)  
(quadraic)  
(linear)  
(linear)
Consider the QCQP

\[
\begin{align*}
\text{min} & \quad x^T Q_0 x + b_0^T x & \text{(quadratic)} \\
\text{s.t.} & \quad x^T Q_k x + b_k^T x \leq c_k & k = 1, \ldots, q \quad \text{(quadratic)} \\
& \quad Ax \leq b & \text{(linear)} \\
& \quad \ell \leq x \leq u & \text{(linear)} 
\end{align*}
\]

Introduce new variables \( X_{i,j} = x_i x_j \):

\[
\begin{align*}
\text{min} & \quad \langle Q_0, X \rangle + b_0^T x & \text{(linear)} \\
\text{s.t.} & \quad \langle Q_k, X \rangle + b_k^T x \leq c_k & k = 1, \ldots, q \quad \text{(linear)} \\
& \quad Ax \leq b & \text{(linear)} \\
& \quad \ell \leq x \leq u & \text{(linear)} \\
& \quad X = xx^T & \text{(quadratic)}
\end{align*}
\]
Consider the QCQP
\[
\begin{align*}
\min & \quad x^T Q_0 x + b_0^T x \\
\text{s.t.} & \quad x^T Q_k x + b_k^T x \leq c_k & \quad k = 1, \ldots, q \\
& \quad Ax \leq b \\
& \quad \ell \leq x \leq u
\end{align*}
\]

Introduce new variables \( X_{i,j} = x_i x_j \):
\[
\begin{align*}
\min & \quad \langle Q_0, X \rangle + b_0^T x \\
\text{s.t.} & \quad \langle Q_k, X \rangle + b_k^T x \leq c_k & \quad k = 1, \ldots, q \\
& \quad Ax \leq b \\
& \quad \ell \leq x \leq u \\
& \quad X = xx^T
\end{align*}
\]

Adams and Sherali [1986], Sherali and Alameddine [1992], Sherali and Adams [1999]:

- relax \( X = xx^T \) by linear inequalities that are derived from multiplications of pairs of linear constraints
Multiplying bounds $\ell_i \leq x_i \leq u_i$ and $\ell_j \leq x_j \leq u_j$ yields

\[
(x_i - \ell_i)(x_j - \ell_j) \geq 0 \\
(x_i - u_i)(x_j - u_j) \geq 0 \\
(x_i - \ell_i)(x_j - u_j) \leq 0 \\
(x_i - u_i)(x_j - \ell_j) \leq 0
\]
Multiplying bounds $\ell_i \leq x_i \leq u_i$ and $\ell_j \leq x_j \leq u_j$ and using $X_{i,j} = x_ix_j$ yields

\[
(x_i - \ell_i)(x_j - \ell_j) \geq 0 \quad \Rightarrow \quad X_{i,j} \geq \ell_i x_j + \ell_j x_i - \ell_i \ell_j
\]

\[
(x_i - u_i)(x_j - u_j) \geq 0 \quad \Rightarrow \quad X_{i,j} \geq u_i x_j + u_j x_i - u_i u_j
\]

\[
(x_i - \ell_i)(x_j - u_j) \leq 0 \quad \Rightarrow \quad X_{i,j} \leq \ell_i x_j + u_j x_i - \ell_i u_j
\]

\[
(x_i - u_i)(x_j - \ell_j) \leq 0 \quad \Rightarrow \quad X_{i,j} \leq u_i x_j + \ell_j x_i - u_i \ell_j
\]
Multiplying bounds \( \ell_i \leq x_i \leq u_i \) and \( \ell_j \leq x_j \leq u_j \) and using \( X_{i,j} = x_i x_j \) yields

\[
(x_i - \ell_i)(x_j - \ell_j) \geq 0 \quad \Rightarrow \quad X_{i,j} \geq \ell_i x_j + \ell_j x_i - \ell_i \ell_j
\]

\[
(x_i - u_i)(x_j - u_j) \geq 0 \quad \Rightarrow \quad X_{i,j} \geq u_i x_j + u_j x_i - u_i u_j
\]

\[
(x_i - \ell_i)(x_j - u_j) \leq 0 \quad \Rightarrow \quad X_{i,j} \leq \ell_i x_j + u_j x_i - \ell_i u_j
\]

\[
(x_i - u_i)(x_j - \ell_j) \leq 0 \quad \Rightarrow \quad X_{i,j} \leq u_i x_j + \ell_j x_i - u_i \ell_j
\]

- these are exactly the McCormick inequalities that we have seen earlier
Multiplying bounds $\ell_i \leq x_i \leq u_i$ and $\ell_j \leq x_j \leq u_j$ and using $X_{i,j} = x_i x_j$ yields

\[
\begin{align*}
(x_i - \ell_i)(x_j - \ell_j) &\geq 0 \quad \Rightarrow \quad X_{i,j} \geq \ell_i x_j + \ell_j x_i - \ell_i \ell_j \\
(x_i - u_i)(x_j - u_j) &\geq 0 \quad \Rightarrow \quad X_{i,j} \geq u_i x_j + u_j x_i - u_i u_j \\
(x_i - \ell_i)(x_j - u_j) &\leq 0 \quad \Rightarrow \quad X_{i,j} \leq \ell_i x_j + u_j x_i - \ell_i u_j \\
(x_i - u_i)(x_j - \ell_j) &\leq 0 \quad \Rightarrow \quad X_{i,j} \leq u_i x_j + \ell_j x_i - u_i \ell_j
\end{align*}
\]

- these are exactly the McCormick inequalities that we have seen earlier
- the resulting linear relaxation is

\[
\begin{align*}
\min \langle Q_0, X \rangle + b_0^T x \\
\text{s.t. } \langle Q_k, X \rangle + b_k^T x &\leq c_k \quad k = 1, \ldots, q \\
Ax &\leq b, \quad \ell \leq x \leq u \\
X_{i,j} &\geq \ell_i x_j + \ell_j x_i - \ell_i \ell_j \quad i, j = 1, \ldots, n, i \leq j \\
X_{i,j} &\geq u_i x_j + u_j x_i - u_i u_j \quad i, j = 1, \ldots, n, i \leq j \\
X_{i,j} &\leq \ell_i x_j + u_j x_i - \ell_i u_j \quad i, j = 1, \ldots, n, \\
X &\geq X^T
\end{align*}
\]
Multiplying bounds $\ell_i \leq x_i \leq u_i$ and $\ell_j \leq x_j \leq u_j$ and using $X_{i,j} = x_i x_j$ yields

\[
(x_i - \ell_i)(x_j - \ell_j) \geq 0 \quad \Rightarrow \quad X_{i,j} \geq \ell_i x_j + \ell_j x_i - \ell_i \ell_j \\
(x_i - u_i)(x_j - u_j) \geq 0 \quad \Rightarrow \quad X_{i,j} \geq u_i x_j + u_j x_i - u_i u_j \\
(x_i - \ell_i)(x_j - u_j) \leq 0 \quad \Rightarrow \quad X_{i,j} \leq \ell_i x_j + u_j x_i - \ell_i u_j \\
(x_i - u_i)(x_j - \ell_j) \leq 0 \quad \Rightarrow \quad X_{i,j} \leq u_i x_j + \ell_j x_i - u_i \ell_j
\]

- these are exactly the McCormick inequalities that we have seen earlier
- the resulting linear relaxation is

\[
\min \langle Q_0, X \rangle + b_0^T x \\
\text{s.t. } \langle Q_k, X \rangle + b_k^T x \leq c_k \quad k = 1, \ldots, q \\
Ax \leq b, \quad \ell \leq x \leq u \\
X_{i,j} \geq \ell_i x_j + \ell_j x_i - \ell_i \ell_j \quad i, j = 1, \ldots, n, i \leq j \\
X_{i,j} \geq u_i x_j + u_j x_i - u_i u_j \quad i, j = 1, \ldots, n, i \leq j \\
X_{i,j} \leq \ell_i x_j + u_j x_i - \ell_i u_j \quad i, j = 1, \ldots, n, \\
X = X^T
\]

- these inequalities are used by all solvers
- not every solver introduces $X_{i,j}$ variables explicitly
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[(A_k^T x - b_k)(x_j - l_j) \geq 0 \Rightarrow \sum_{i=1}^{n} A_{k,i}x_i(x_j - l_j) - b_k(x_j - l_j) \geq 0\]
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[
(A_k^T x - b_k)(x_j - \ell_j) \geq 0 \implies \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i \ell_j) - b_k(x_j - \ell_j) \geq 0
\]
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[(A_k^T x - b_k)(x_j - \ell_j) \geq 0 \Rightarrow \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i \ell_j) - b_k(x_j - \ell_j) \geq 0\]

\[(A_k^T x - b_k)(A_{k'}^T x - b_{k'}) \geq 0 \Rightarrow A_k^T A_{k'}^T x - b_k A_{k'}^T x - b_{k'} A_k^T x + b_k b_{k'} \geq 0\]
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[(A_k^T x - b_k)(x_j - \ell_j) \geq 0 \Rightarrow \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i \ell_j) - b_k(x_j - \ell_j) \geq 0\]

\[(A_k^T x - b_k)(A_{k'}^T x - b_{k'}) \geq 0 \Rightarrow A_k^T X A_{k'}^T - (b_k A_{k'} + b_{k'} A_k^T)x + b_k b_{k'} \geq 0\]
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[(A_k^T x - b_k)(x_j - \ell_j) \geq 0 \quad \Rightarrow \quad \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i\ell_j) - b_k(x_j - \ell_j) \geq 0\]

\[(A_k^T x - b_k)(A_{k'}^T x - b_{k'}) \geq 0 \quad \Rightarrow \quad A_k^T X A_{k'}^T - (b_k A_{k'} + b_{k'} A_k^T)x + b_k b_{k'} \geq 0\]

**ANTIGONE [Misener and Floudas, 2012]:**

- **linear inequality** \((A_k^T x - b_k \leq 0) \times \text{variable bound} \ (x_j - \ell_j \geq 0)\)
  \[\Rightarrow\] consider for cut generation and bound tightening
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[(A_k^T x - b_k)(x_j - \ell_j) \geq 0 \Rightarrow \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i \ell_j) - b_k(x_j - \ell_j) \geq 0\]

\[(A_k^T x - b_k)(A_{k'}^T x - b_{k'}) \geq 0 \Rightarrow A_k^T X A_{k'}^T - (b_k A_{k'} + b_{k'} A_k^T)x + b_k b_{k'} \geq 0\]

**ANTIGONE [Misener and Floudas, 2012]:**

- **linear inequality** \((A_k^T x - b_k \leq 0) \times \text{variable bound } (x_j - \ell_j \geq 0)\)
  \(\Rightarrow\) consider for cut generation and bound tightening

- **linear (in)equality** \((A_k^T x - b_k \leq/=0) \times \text{linear (in)equality } (A_{k'}^T x - b_{k'} \leq/=0)\)
  \(\Rightarrow\) consider for cut generation and bound tightening
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[(A^T_k x - b_k)(x_j - \ell_j) \geq 0 \implies \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i \ell_j) - b_k(x_j - \ell_j) \geq 0\]

\[(A^T_k x - b_k)(A^T_{k'} x - b_{k'}) \geq 0 \implies A^T_k X A^T_{k'} - (b_k A_{k'} + b_{k'} A^T_k)x + b_k b_{k'} \geq 0\]

ANTIGONE [Misener and Floudas, 2012]:

- **linear inequality** \((A^T_k x - b_k \leq 0) \times \text{variable bound} (x_j - \ell_j \geq 0)\)
  \(\implies\) consider for cut generation and bound tightening

- **linear (in)equality** \((A^T_k x - b_k \leq/=0) \times \text{linear (in)equality} (A^T_{k'} x - b_{k'} \leq/=0)\)
  \(\implies\) consider for cut generation and bound tightening

- **linear equality** \((A^T_k x - b_k = 0) \times \text{variable} (x_j)\)
  \(\implies\) add to the model (strong cut in context of QAP, pooling, ...)
  
  (close to bilinear term elimination of Liberti and Pantelides [2006])
Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

\[(A_k^T x - b_k)(x_j - \ell_j) \geq 0 \Rightarrow \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i\ell_j) - b_k(x_j - \ell_j) \geq 0\]

\[(A_k^T x - b_k)(A_{k'}^T x - b_{k'}) \geq 0 \Rightarrow A_k^T X A_{k'}^T - (b_k A_{k'} + b_{k'} A_k^T)x + b_k b_{k'} \geq 0\]

**ANTIGONE [Misener and Floudas, 2012]:**

- **linear inequality** \((A_k^T x - b_k \leq 0) \times \text{variable bound} (x_j - \ell_j \geq 0)\)
  \(\Rightarrow\) consider for cut generation and bound tightening
- **linear (in)equality** \((A_k^T x - b_k \leq/=0) \times \text{linear (in)equality} (A_{k'}^T x - b_{k'} \leq/=0)\)
  \(\Rightarrow\) consider for cut generation and bound tightening
- **linear equality** \((A_k^T x - b_k = 0) \times \text{variable} (x_j)\)
  \(\Rightarrow\) add to the model (strong cut in context of QAP, pooling, ...)
  (close to bilinear term elimination of Liberti and Pantelides [2006])
- in all cases, consider only products that **do not add new nonlinear terms**
  (avoid \(X_{i,j}\) without corresponding \(x_i x_j\))
- learn useful RLT cuts in the first levels of branch-and-bound
Semidefinite Programming (SDP) Relaxation

\[
\begin{align*}
\min & \quad x^T Q_0 x + b_0^T x \\
\text{s.t.} & \quad x^T Q_k x + b_k^T x \leq c_k \\
A & \quad x \leq b \\
\ell & \quad x \leq u_x \\
\end{align*}
\]

\[
\begin{align*}
\min & \quad \langle Q_0, X \rangle + b_0^T x \\
\text{s.t.} & \quad \langle Q_k, X \rangle + b_k^T x \leq c_k \\
A & \quad x \leq b \\
\ell & \quad x \leq u_x \\
X & \quad = xx^T
\end{align*}
\]

- relaxing \( X - xx^T = 0 \) to \( X - xx^T \succeq 0 \), which is equivalent to

\[
\tilde{X} := \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0,
\]

yields a semidefinite programming relaxation
Semidefinite Programming (SDP) Relaxation

\[
\min x^T Q_0 x + b_0^T x \\
\text{s.t. } x^T Q_k x + b_k^T x \leq c_k \\
\quad A x \leq b \\
\quad \ell_x \leq x \leq u_x \\
\Leftrightarrow \\
\min \langle Q_0, X \rangle + b_0^T x \\
\text{s.t. } \langle Q_k, X \rangle + b_k^T x \leq c_k \\
\quad A x \leq b \\
\quad \ell_x \leq x \leq u_x \\
X = xx^T
\]

- relaxing \( X - xx^T = 0 \) to \( X - xx^T \succeq 0 \), which is equivalent to

\[
\tilde{X} := \begin{pmatrix} 1 & x^T \\ x & X \end{pmatrix} \succeq 0,
\]

yields a semidefinite programming relaxation

- SDP is computationally demanding, so approximate by linear inequalities:
  for \( \tilde{X}^* \not\succeq 0 \) compute eigenvector \( v \) with eigenvalue \( \lambda < 0 \), then

\[
\langle v, \tilde{X} v \rangle \geq 0
\]

is a valid cut that cuts off \( \tilde{X}^* \) [Sherali and Fraticelli, 2002]

- available in Couenne and Lindo API (non-default)
- Qualizza, Belotti, and Margot [2009] (Couenne): sparsify cut by setting entries of \( v \) to 0
Anstreicher [2009]:

- the SDP relaxation does not dominate the RLT relaxation
- the RLT relaxation does not dominate the SDP relaxation
- \textbf{combining both} relaxations can produce substantially better bounds
Anstreicher [2009]:

- the SDP relaxation does not dominate the RLT relaxation
- the RLT relaxation does not dominate the SDP relaxation
- combining both relaxations can produce substantially better bounds

Anstreicher [2012]:

- the SDP relaxation dominates the $\alpha$-BB underestimators
Acceleration – Selected Topics

Primal Heuristics
Sub-NLP Heuristics

Given a solution satisfying all integrality constraints,

- fix all integer variables in the MINLP
- call an NLP solver to find a local solution to the remaining NLP
Sub-NLP Heuristics

Given a solution satisfying all integrality constraints,

- **fix all integer variables** in the MINLP
- call an **NLP solver** to find a **local solution** to the remaining NLP
- variable fixings given by integer-feasible solution to LP relaxation
- additionally, SCIP runs its MIP heuristics on MIP relaxation (rounding, diving, feas. pump, LNS, . . .)
Given a solution satisfying all integrality constraints,

- **fix all integer variables** in the MINLP
- call an **NLP solver** to find a **local solution** to the remaining NLP
- variable fixings given by integer-feasible solution to LP relaxation
- additionally, SCIP runs its MIP heuristics on MIP relaxation (rounding, diving, feas. pump, LNS, ...)  

**NLP-Diving**: solve NLP relaxation, restrict bounds on fractional variable, repeat
Sub-NLP Heuristics

Given a solution satisfying all integrality constraints,

- fix all integer variables in the MINLP
- call an NLP solver to find a local solution to the remaining NLP
- variable fixings given by integer-feasible solution to LP relaxation
- additionally, SCIP runs its MIP heuristics on MIP relaxation (rounding, diving, feas. pump, LNS, . . .)

**NLP-Diving**: solve NLP relaxation, restrict bounds on fractional variable, repeat

**Multistart**: run local NLP solver from random starting points to increase likelihood of finding global optimum

Smith, Chinneck, and Aitken [2013]: sample many random starting points, move them cheaply towards feasible region (average gradients of violated constraints), cluster, run NLP solvers from (few) center of cluster (in SCIP [Maher et al., 2017])
"Undercover" (SCIP) [Berthold and Gleixner, 2014]:

- **Fix nonlinear variables**, so problem becomes MIP (pass to SCIP)
- not always necessary to fix all nonlinear variables, e.g., consider \(x \cdot y\)
- find a **minimal set of variables** to fix by solving a Set Covering Problem
“Undercover” (SCIP) [Berthold and Gleixner, 2014]:

- Fix nonlinear variables, so problem becomes MIP (pass to SCIP)
- not always necessary to fix all nonlinear variables, e.g., consider $x \cdot y$
- find a minimal set of variables to fix by solving a Set Covering Problem

Large Neighborhood Search [Berthold, Heinz, Pfetsch, and Vigerske, 2011]:

- RENS [Berthold, 2014b]: fix integer variables with integral value in LP relaxation
- RINS, DINS, Crossover, Local Branching
Iterative Rounding Heuristic (Couenne) [Nannicini and Belotti, 2012]:

1. find a local optimal solution to the NLP relaxation
2. find the nearest integer feasible solution to the MIP relaxation
3. fix integer variables in MINLP and solve remaining sub-NLP locally
4. forbid found integer variable values in MIP relaxation (no-good-cuts) and reiterate
Rounding Heuristics

Iterative Rounding Heuristic (Couenne) [Nannicini and Belotti, 2012]:

1. find a local optimal solution to the NLP relaxation
2. find the nearest integer feasible solution to the MIP relaxation
3. fix integer variables in MINLP and solve remaining sub-NLP locally
4. forbid found integer variable values in MIP relaxation (no-good-cuts) and reiterate

Feasibility Pump (Couenne) [Belotti and Berthold, 2017]:

- alternately find feasible solutions to MIP and NLP relaxations
- solution of NLP relaxation is “rounded” to solution of MIP relaxation (by various methods trading solution quality with computational effort)
- solution of MIP relaxation is projected onto NLP relaxation (local search)
- various choices for objective functions and accuracy of MIP relaxation
- D’Ambrosio et al. [2010, 2012]: previous work on Feasibility Pump for nonconvex MINLP
Thank you for your attention!

Consider contributing your NLP and MINLP instances to MINLPLib\(^1\)!

Some recent MINLP reviews:
- Burer and Letchford [2012]
- Belotti, Kirches, Leyffer, Linderoth, Luedtke, and Mahajan [2013]
- Boukouvala, Misener, and Floudas [2016]

Some recent books:
- Lee and Leyffer [2012]
- Locatelli and Schoen [2013]

\(^1\)http://www.gamsworld.org/minlp/minlplib2/html/index.html
References


