MINLP Solver Technology

Stefan Vigerske

October 24, 2017

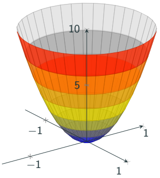
Outline

Introduction

Mixed-Integer Nonlinear Programs (MINLPs)

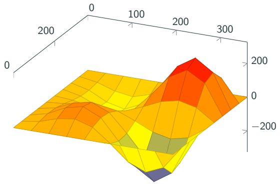
$$
\begin{aligned}
\min\ & c^{\mathsf{T}} x \\
\text{s.t. } & g_k(x) \le 0 && \forall k \in [m] \\
& x_i \in \mathbb{Z} && \forall i \in \mathcal{I} \subseteq [n] \\
& x_i \in [\ell_i, u_i] && \forall i \in [n]
\end{aligned}
$$

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



convex

or



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 determinstic optimization problems" (Jon Lee, 2008)

Solvers for Convex MINLP

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

- can often work as heuristic for nonconvex MINLP

Deterministic:

solver	1st ver.	citation
αBB	1995	Adjiman, Androulakis, and Floudas [1998a]
BARON	1996	Sahinidis [1996], Tawarmalani and Sahinidis [2005]
in gPROMS	1996	Smith and Pantelides [1997]

Solvers for General MINLP

Deterministic:

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

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

Solvers for General MINLP

Deterministic:

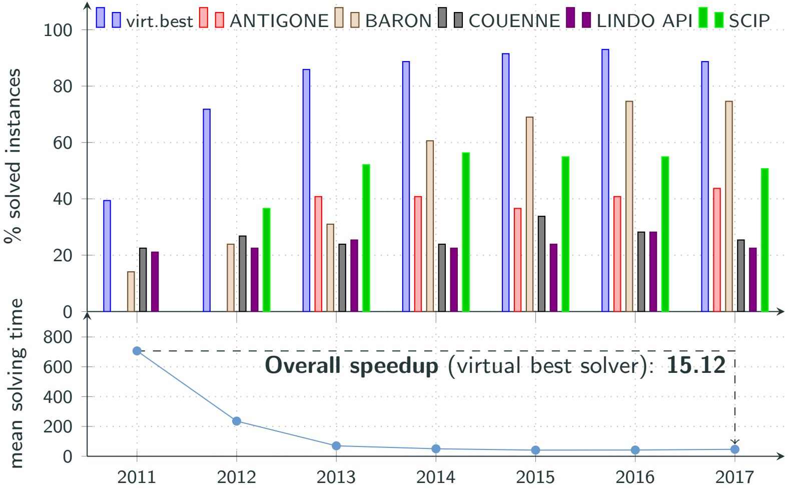
solver	1st ver.	citation
αBB	1995	Adjiman, Androulakis, and Floudas [1998a]
BARON	1996	Sahinidis [1996], Tawarmalani and Sahinidis [2005]
in gPROMS	1996	Smith and Pantelides [1997]
Couenne	2008	Belotti, Lee, Liberti, Margot, and Wächter [2009]
LindoAPI	2009	Lin and Schrage [2009]
SCIP	2012	Achterberg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

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

date	ANTIGONE	BARON	COUENNE	LINDO API	SCIP
08/2011	–	**9.3**.1	**0.3**	**6.1**.1.588	–
11/2012	–	**11.5**.2	0.4	7.0.1.497	2.1.2
07/2013	**1.1**	**12.3**.3	0.4	**8.0**.1283.385	3.0
09/2014	1.1	**14.0**.3	0.4	8.0.1694.550	**3.1**
11/2015	1.1	**15.9**.22	**0.5**	9.0.2120.225	**3.2**
09/2016	1.1	**16.8**.24	0.5	9.0.2217.293	3.2
08/2017	1.1	**17.8**.7	0.5	**11.0**.3742.269	**4.0**

Fundamental Methods

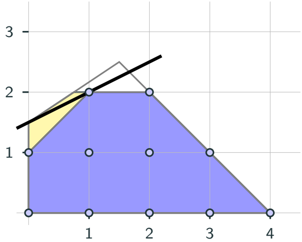
Fundamental Methods

Recap: Mixed-Integer Linear Programming

For mixed-integer linear programs (MIP), that is,

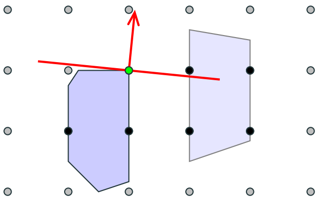
$$\begin{aligned} &\min c^{\mathsf{T}} x, \\ &\text{s.t. } Ax \le b, \\ &\quad x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \end{aligned}$$

the dominant method of **Branch & Cut** combines



cutting planes
[Gomory, 1958]

&

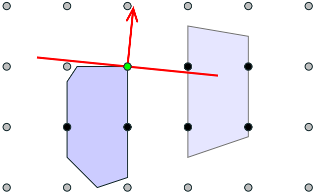


branch-and-bound
[Land and Doig, 1960]

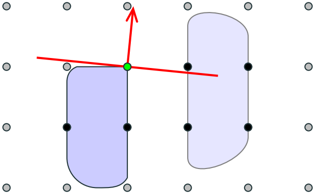
Fundamental Methods

Convex MINLP

NLP-based Branch & Bound (NLP-BB)



$\Rightarrow$



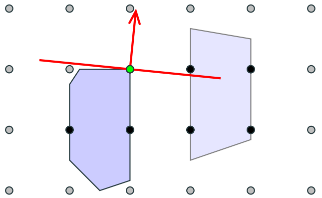
MIP branch-and-bound
[Land and Doig, 1960]

MINLP branch-and-bound
[Leyffer, 1993]

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

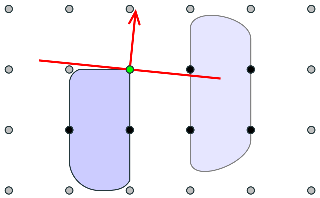
Branching: Subdivide problem along variables x_i, $i \in \mathcal{I}$, that take fractional value in NLP solution.

NLP-based Branch & Bound (NLP-BB)



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

$\Rightarrow$



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

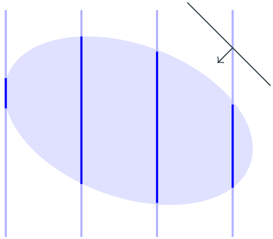
$$\begin{aligned} \min\ & c^\top x, \\ \text{s.t. } & g_k(\hat{x}) + \nabla g_k(\hat{x})^\top (x - \hat{x}) \leq 0, \\ & \quad k \in [m], \quad \hat{x} \in R, \\ & x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ & x \in [\ell, u], \end{aligned}$$

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.,

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

Example:

$$\begin{aligned} \min\ & x + y \\ \text{s.t. } & (x, y) \in \text{ellipsoid} \\ & x \in \{0, 1, 2, 3\} \\ & y \in [0, 3] \end{aligned}$$



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

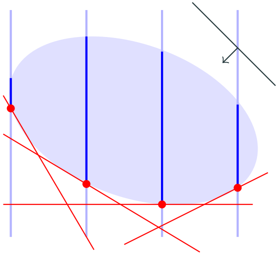
$$\begin{aligned} &\min\ c^{\mathsf{T}} x, \\ &\text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^{\mathsf{T}}(x - \hat{x}) \le 0, \\ &\qquad\qquad k \in [m], \quad \hat{x} \in R, \\ &\qquad x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ &\qquad x \in [\ell, u], \end{aligned}$$

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.,

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

Example:

$$\begin{aligned} &\min\ x + y \\ &\text{s.t. } (x, y) \in \text{ellipsoid} \\ &\qquad x \in \{0, 1, 2, 3\} \\ &\qquad y \in [0, 3] \end{aligned}$$



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

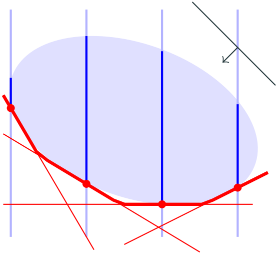
$$\begin{aligned} &\min c^{\mathsf{T}} x, \\ &\text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^{\mathsf{T}}(x - \hat{x}) \le 0, \\ &\qquad k \in [m], \quad \hat{x} \in R, \\ &\quad x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ &\quad x \in [\ell, u], \end{aligned}$$

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.,

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

Example:

$$\begin{aligned} &\min x + y \\ &\text{s.t. } (x, y) \in \text{ellipsoid} \\ &\qquad x \in \{0, 1, 2, 3\} \\ &\qquad y \in [0, 3] \end{aligned}$$



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

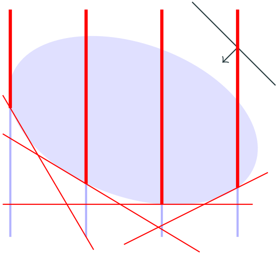
$$\begin{aligned} \min\ & c^{\top} x, \\ \text{s.t. } & g_k(\hat{x}) + \nabla g_k(\hat{x})^{\top}(x - \hat{x}) \le 0, \\ & \quad k \in [m], \quad \hat{x} \in R, \\ & x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ & x \in [\ell, u], \end{aligned}$$

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.,

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

Example:

$$\begin{aligned} \min\ & x + y \\ \text{s.t. } & (x, y) \in \text{ellipsoid} \\ & x \in \{0, 1, 2, 3\} \\ & y \in [0, 3] \end{aligned}$$



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$$\equiv$$

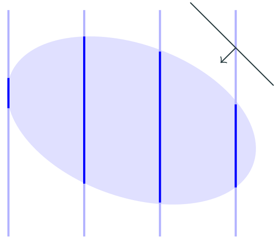
MIP

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

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

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

$\equiv$

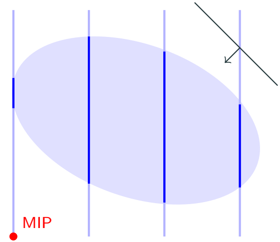
MIP

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

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

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

$\equiv$

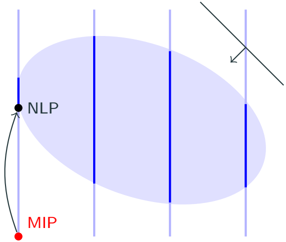
MIP

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

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

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

$\equiv$

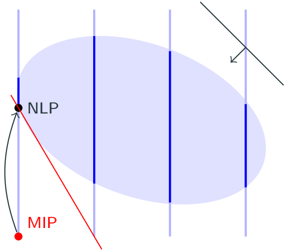
MIP

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

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

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

$\equiv$

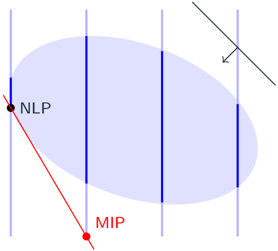
MIP

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

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

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

$$\equiv$$

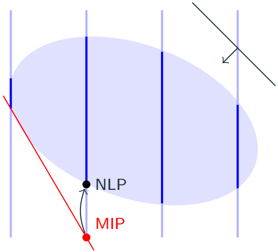
MIP

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

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

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

$$\equiv$$

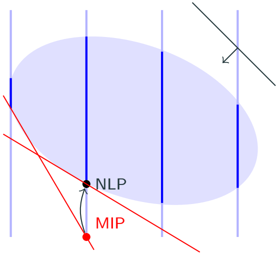
MIP

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

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

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

$$\equiv$$

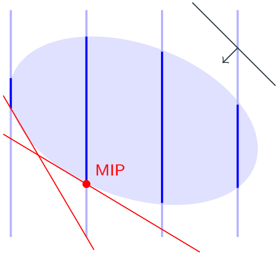
MIP

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

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

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

$$\equiv$$

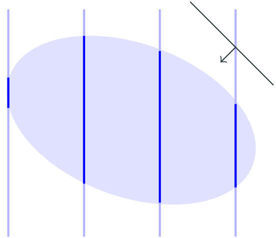
MIP

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

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

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

$$\equiv$$

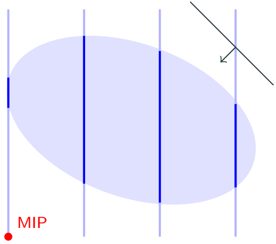
MIP

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

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

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

$$\equiv$$

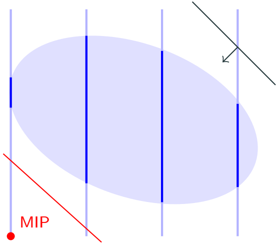
MIP

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

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

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

$$\equiv$$

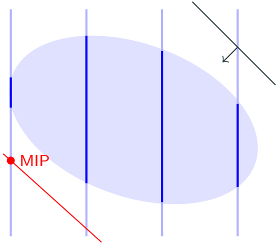
MIP

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

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

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

$$\equiv$$

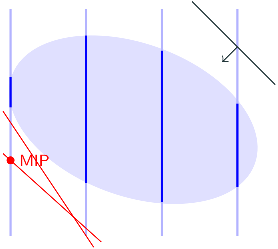
MIP

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

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

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

$\equiv$

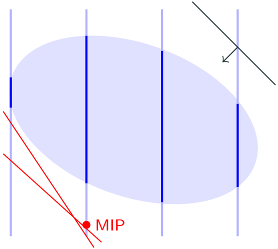
MIP

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

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

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

$\equiv$

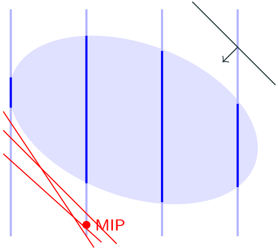
MIP

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

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

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

$\equiv$

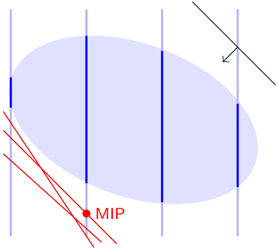
MIP

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

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

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

$$\equiv$$

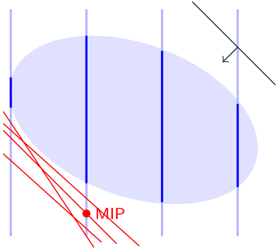
MIP

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

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

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

$\equiv$

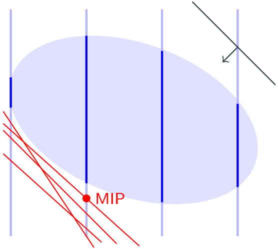
MIP

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

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

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

$\equiv$

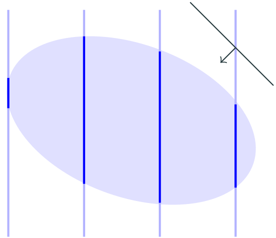
MIP

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

Extended Hyperplane Method (EHP)

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

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set $\{x \in [\ell, u] : g_k(x) \le 0\}$ 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

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

$$\equiv$$

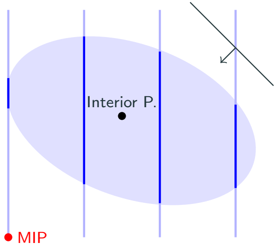
MIP

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

Extended Hyperplane Method (EHP)

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

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



Convex MINLP

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

$$\equiv$$

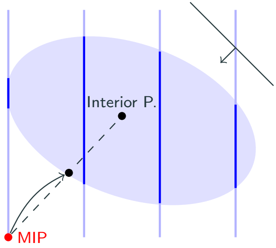
MIP

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

Extended Hyperplane Method (EHP)

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

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set $\{x \in [\ell, u] : g_k(x) \le 0\}$ 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

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

$\equiv$

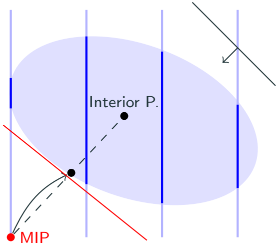
MIP

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

Extended Hyperplane Method (EHP)

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

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set $\{x \in [\ell, u] : g_k(x) \le 0\}$ 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

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

$\equiv$

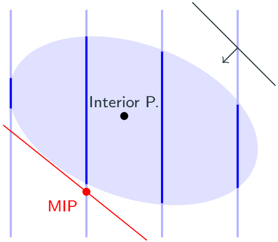
MIP

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

Extended Hyperplane Method (EHP)

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

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set $\{x \in [\ell, u] : g_k(x) \le 0\}$ 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

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

$\equiv$

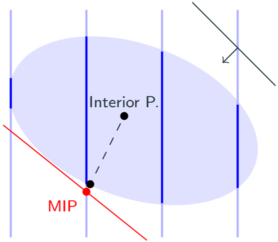
MIP

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

Extended Hyperplane Method (EHP)

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

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



Convex MINLP

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

$\equiv$

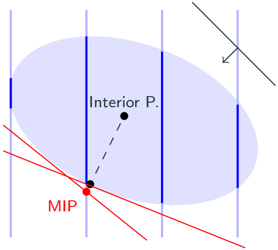
MIP

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

Extended Hyperplane Method (EHP)

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

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



Convex MINLP

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

$\equiv$

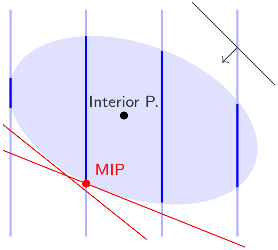
MIP

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

Extended Hyperplane Method (EHP)

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

- Iteratively solve MIP relaxation only.
- Move MIP solution onto NLP-feasible set $\{x \in [\ell, u] : g_k(x) \le 0\}$ 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:

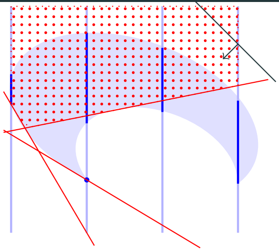
- Integrate Kelley' Cutting Plane method into MIP Branch & Bound.
- Add linearization in LP solution to LP relaxation (as in ECP).
- Optional: Move LP solution onto NLP-feasible set $\{x \in [\ell, u] : g_k(x) \leq 0\}$ via linesearch (as in EHP) [Maher, Fischer, Gally, Gamrath, Gleixner, Gottwald, Hendel,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

Fundamental Methods

Nonconvex MINLP

Nonconvex MINLP

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

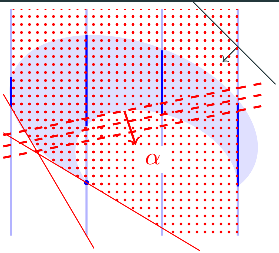


Outer-Approximation:

- 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.



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

Outer-Approximation:

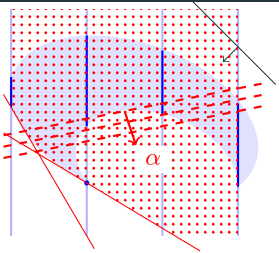
- Linearizations
  $g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq 0$
  may not be valid.
- Heuristics: add cuts as "soft-constraints"
  $\min_{\alpha \geq 0} \alpha$ s.t. $g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq \alpha$

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:

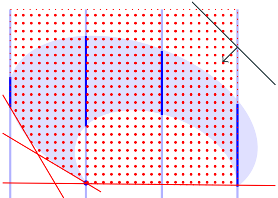
- Linearizations
  $g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq 0$
  may not be valid.
- Heuristics: add cuts as "soft-constraints"
  $\min_{\alpha \geq 0} \alpha$ s.t. $g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq \alpha$

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.

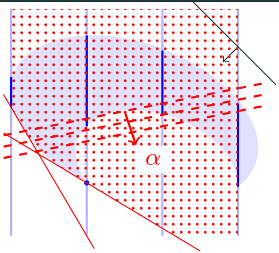
Exact approach: **Spatial Branch & Bound**:

- **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:

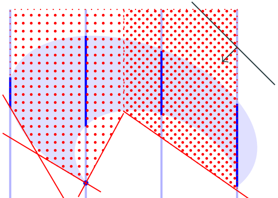
- Linearizations
  $g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq 0$
  may not be valid.
- Heuristics: add cuts as "soft-constraints"
  $\min_{\alpha \geq 0} \alpha$ s.t. $g_k(\hat{x}) + \nabla g_k(\hat{x})(x - \hat{x}) \leq \alpha$

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: $\mathrm{conv}(X)$ – 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: $\mathrm{conv}(X)$ – convex hull of X

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

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

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

- In practice, $\mathrm{conv}\{x \in [\ell, u] \,:\, g_k(x) \leq 0\}$ 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: $\mathrm{conv}(X)$ – convex hull of X

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

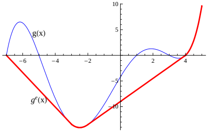
Relax I: Convexify the feasible sets that are defined by each constraint individually, i.e.,

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

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

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

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



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

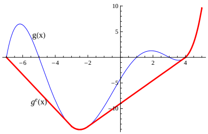
- In practice, $\text{conv}(X)$ is impossible to construct explicitly.

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.

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

Convex Relaxation

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

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

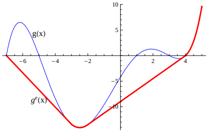
- In practice, $\mathrm{conv}(X)$ is impossible to construct explicitly.

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

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

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

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

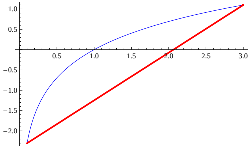


$$\{x \in [\ell, u] \,:\, \text{“}\mathrm{conv}(g_k)\text{”}(x) \le 0\}$$

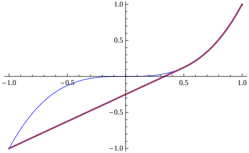
- In practice, convex envelope is not known explicitly in general
  – **except for many “simple functions”**

Convex Envelopes for “simple” functions

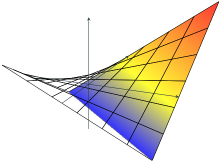
concave functions



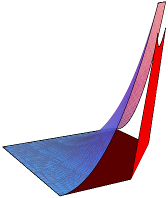
$x^k \quad (k \in 2\mathbb{Z}+1)$



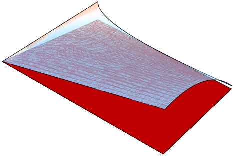
$x \cdot y$



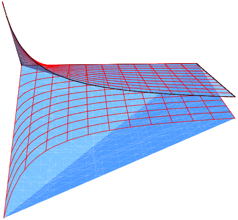
$x^2 \cdot y^2$



$-\sqrt{x} \cdot y^2$



$x/y \quad (0 < y < \infty)$



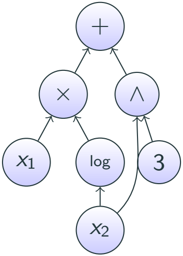
Factorable Functions [McCormick, 1976]

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

$$
\begin{array}{l}
x_1 \log(x_2) + x_2^3 \le 0 \\
x_1 \in [1,2],\ x_2 \in [1,e]
\end{array}
\quad \overset{}{\underset{\text{Reform}}{\Longrightarrow}} \quad
\begin{array}{l}
y_1 + y_2 \le 0 \\
x_1 y_3 = y_1 \\
x_2^3 = y_2 \\
\log(x_2) = y_3 \\
x_1 \in [1,2],\ x_2 \in [1,e] \\
y_1 \in [0,2],\ y_2 \in [1,e^3] \\
y_3 \in [0,1]
\end{array}
$$

- 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., $xyz = (xy)z = x(yz)$.

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.

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

$\Longrightarrow$ Reform

$$\begin{aligned} y_1 + y_2 &\le 0 \\ x_1 y_3 &= y_1 \\ x_2^3 &= y_2 \\ \log(x_2) &= y_3 \end{aligned}$$
$$x_1 \in [1,2],\ x_2 \in [1,e]$$
$$y_1 \in [0,2],\ y_2 \in [1,e^3]$$
$$y_3 \in [0,1]$$

$\Longrightarrow$ Convex Relax

$$\begin{aligned} y_1 + y_2 &\le 0 \\ 2y_3 + x_1 - 2 &\le y_1 \\ y_3 &\le y_1 \\ y_1 &\le 2y_3 \\ y_1 &\le y_3 + x - 1 \\ x_2^3 &\le y_2 \\ y_2 &\le 1 + \frac{e^3-1}{e-1}(x_2 - 1) \\ \frac{1}{e-1}(x_2 - 1) &\le y_3 \\ y_3 &\le \log(x_2) \end{aligned}$$
$$x_1 \in [1,2],\ x_2 \in [1,e]$$
$$y_1 \in [0,2],\ y_2 \in [1,e^3],\ \ y_3 \in [0,1]$$

- 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., $xyz = (xy)z = x(yz)$.

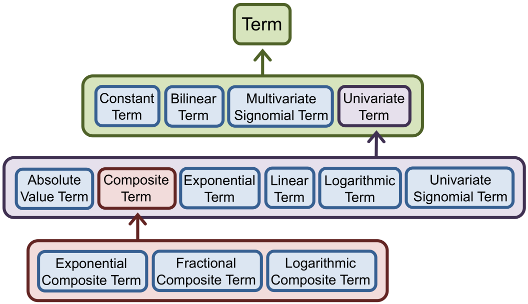
Factorable Reformulation in Practice

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]:



Spatial Branching

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**:

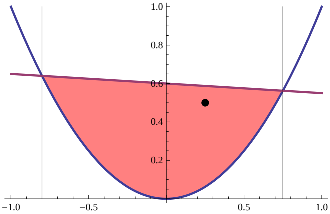
- ✓ **Relax nonconvexity** to obtain a tractable relaxation (often an LP).
- **Branch on "nonconvexities"** to enforce original constraints.

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

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

the convex relaxation is

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



Spatial Branching

Recall **Spatial Branch & Bound**:

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

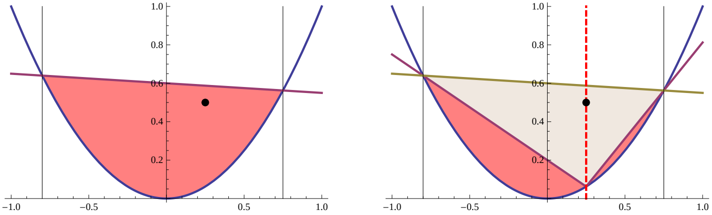
The variable bounds determine the convex relaxation, e.g., for the constraint

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

the convex relaxation is

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

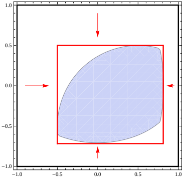
Thus, branching on a nonlinear variable in a nonconvex term allows for tighter relaxations in sub-problems:



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:

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

where $\mathcal{R} = \{x \in [\ell, u] : g(x) \leq 0, x_i \in \mathbb{Z}, i \in \mathcal{I}\}$ (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

Feasbility-based Bound Tightening (FBBT):

Deduce variable bounds from single constraint and box $[\ell, u]$, that is

$$\mathcal{R} = \{x \in [\ell, u] \ : \ g_j(x) \leq 0\} \qquad \text{for some fixed } j \in [m].$$

- cheap and effective ⇒ used for "probing"

Feasibility-Based Bound Tightening

Feasbility-based Bound Tightening (FBBT):
Deduce variable bounds from single constraint and box $[\ell, u]$, that is

$$\mathcal{R} = \{x \in [\ell, u] \;:\; g_j(x) \leq 0\} \qquad \text{for some fixed } j \in [m].$$

- cheap and effective $\Rightarrow$ used for "probing"

Linear Constraints:

$$b \leq \sum_{i:a_i>0} a_i x_i + \sum_{i:a_i<0} a_i x_i \leq c, \qquad \ell \leq x \leq u$$

$$\Rightarrow \qquad x_j \leq \frac{1}{a_j} \begin{cases} c - \sum\limits_{i:a_i>0, i\neq j} a_i \ell_i - \sum\limits_{i:a_i<0} a_i u_i, & \text{if } a_j > 0 \\ b - \sum\limits_{i:a_i>0} a_i u_i - \sum\limits_{i:a_i<0, i\neq j} a_i \ell_i, & \text{if } a_j < 0 \end{cases}$$

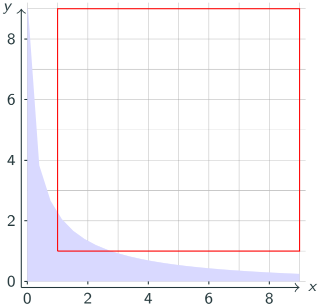
$$x_j \geq \frac{1}{a_j} \begin{cases} b - \sum\limits_{i:a_i>0, i\neq j} a_i u_i - \sum\limits_{i:a_i<0} a_i \ell_i, & \text{if } a_j > 0 \\ c - \sum\limits_{i:a_i>0} a_i \ell_i - \sum\limits_{i:a_i<0, i\neq j} a_i u_i, & \text{if } a_j < 0 \end{cases}$$

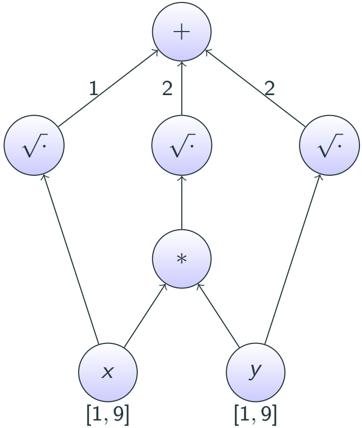
- Belotti, Cafieri, Lee, and Liberti [2010]: fixed point of iterating FBBT on set of linear constraints can be computed by solving one LP
- Belotti [2013]: FBBT on **two** linear constraints simultaneously

Example:

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

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



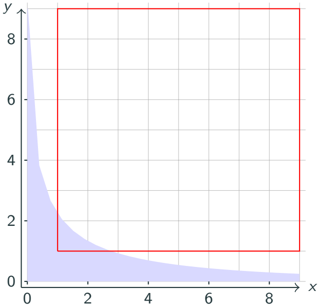


Feasibility-Based Bound Tightening on Expression “Tree”

Example:

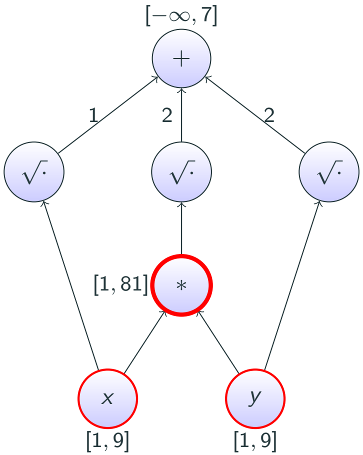
$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$

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



Forward propagation:

- compute bounds on intermediate nodes (bottom-up)



$[1, 9] * [1, 9] = [1, 81]$

Application of **Interval Arithmetics**

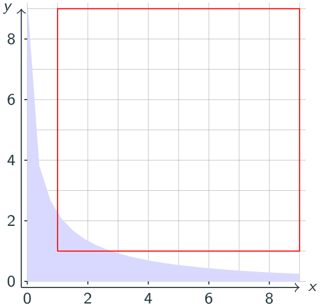
[Moore, 1966]

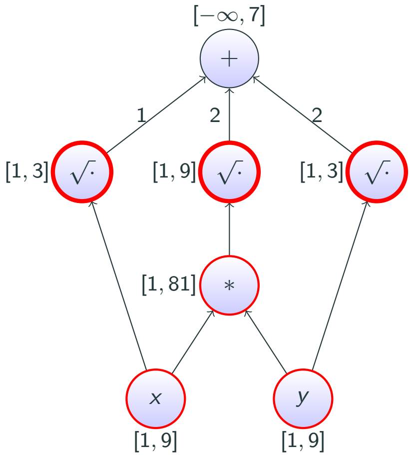
Feasibility-Based Bound Tightening on Expression "Tree"

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)

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

Application of **Interval Arithmetics**

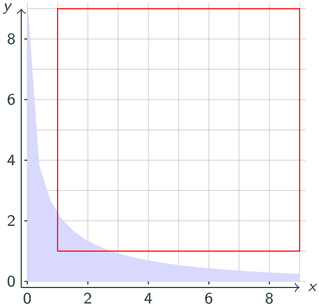
[Moore, 1966]

Feasibility-Based Bound Tightening on Expression "Tree"

Example:

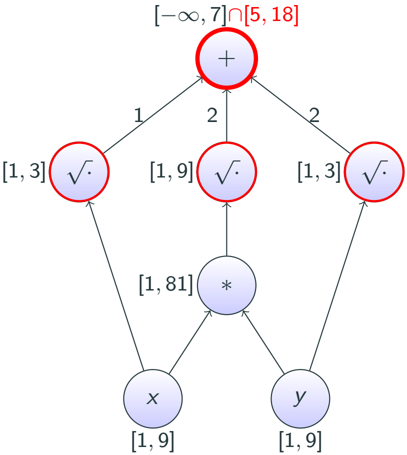
$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$

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



Forward propagation:

- compute bounds on intermediate nodes (bottom-up)



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

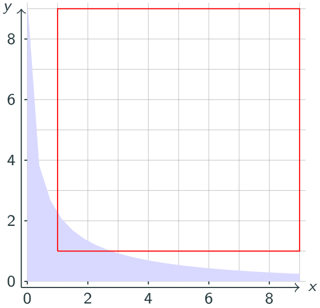
Application of **Interval Arithmetics**

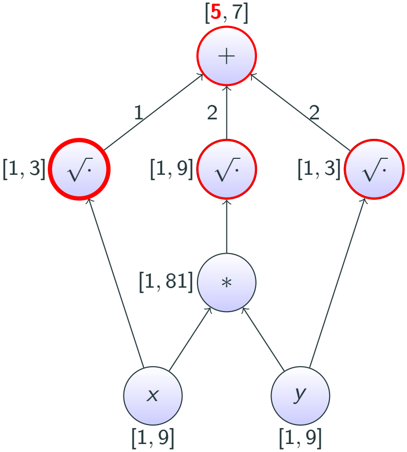
[Moore, 1966]

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)

$$[5, 7] - 2\,[1, 9] - 2\,[1, 3] = [-19, 3]$$

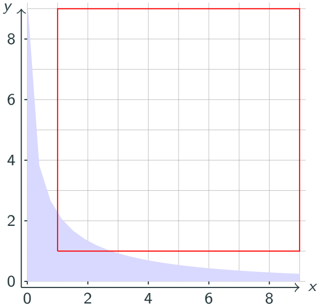
Application of **Interval Arithmetics**

[Moore, 1966]

Example:

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

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

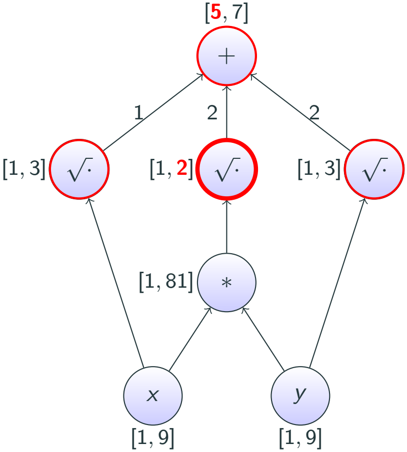


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

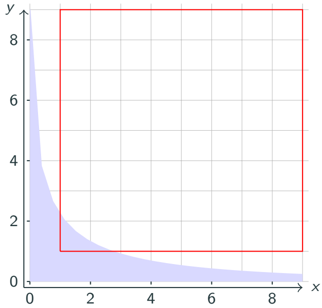
Application of **Interval Arithmetics**

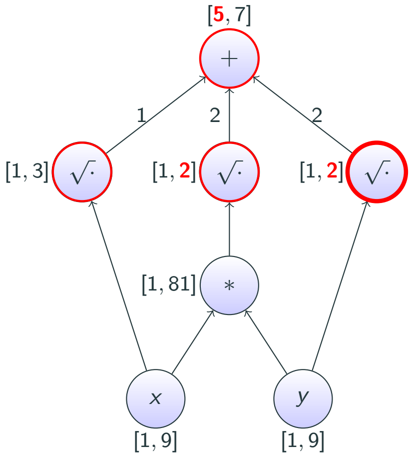
[Moore, 1966]

Example:

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

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





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

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

Application of **Interval Arithmetics**

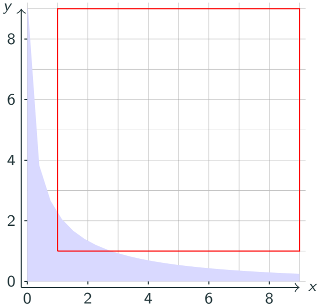
[Moore, 1966]

Feasibility-Based Bound Tightening on Expression "Tree"

Example:

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

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

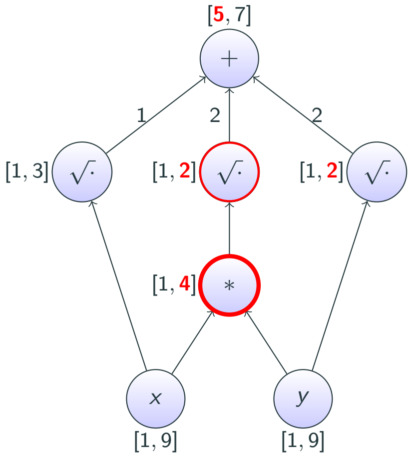


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

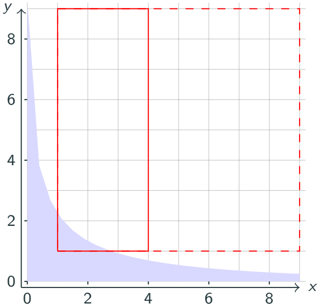
Application of **Interval Arithmetics**

[Moore, 1966]

Example:

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

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

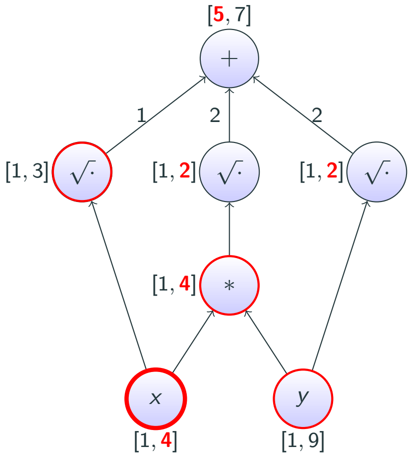


Forward propagation:

- 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**

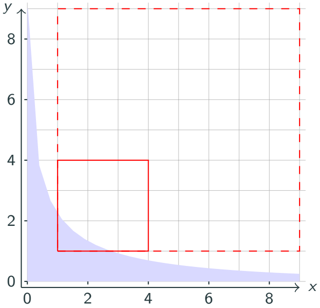
[Moore, 1966]

Feasibility-Based Bound Tightening on Expression “Tree”

Example:

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

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

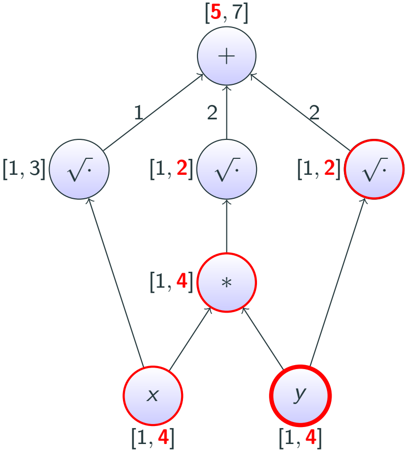


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



$[1, 2]^2 = [1, 4] \qquad [1, 4]/[1, 4] = [1/4, 4]$

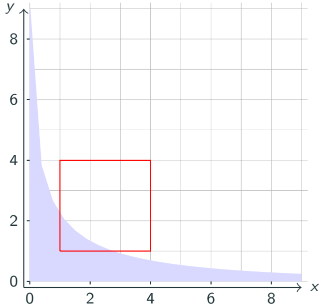
Application of **Interval Arithmetics**

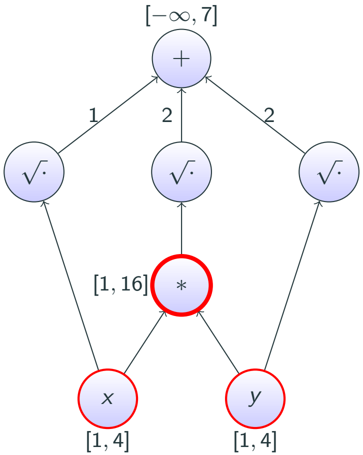
[Moore, 1966]

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)

$[1, 4] * [1, 4] = [1, 16]$

Backward propagation:

- reduce bounds using reverse operations (top-down)

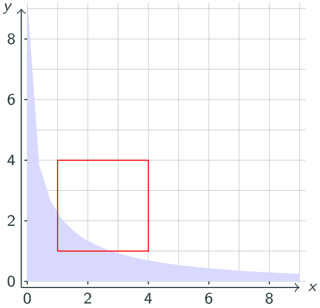
Application of **Interval Arithmetics**

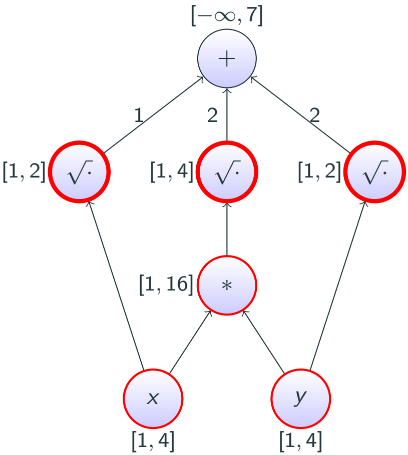
[Moore, 1966]

Example:

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

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





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

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

Application of **Interval Arithmetics**

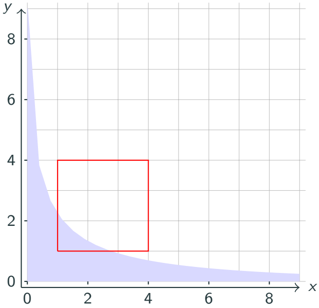
[Moore, 1966]

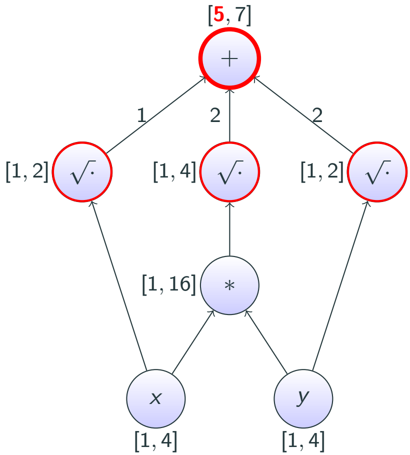
Feasibility-Based Bound Tightening on Expression "Tree"

Example:

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

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





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

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

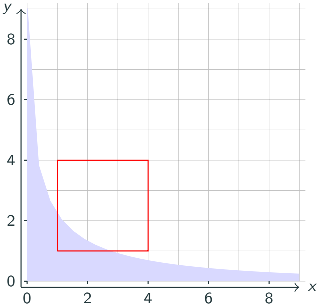
Application of **Interval Arithmetics**

[Moore, 1966]

Example:

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

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

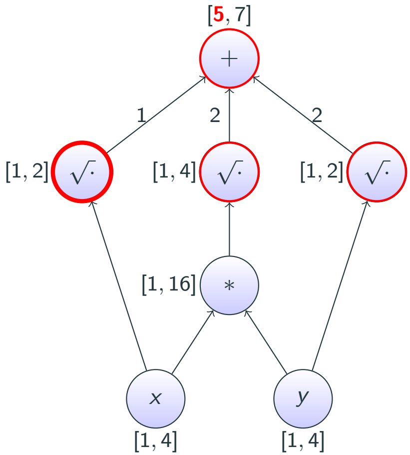


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



$$[5, 7] - 2\,[1, 4] - 2\,[1, 2] = [-7, 3]$$

Application of **Interval Arithmetics**

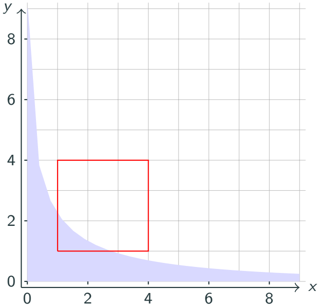
[Moore, 1966]

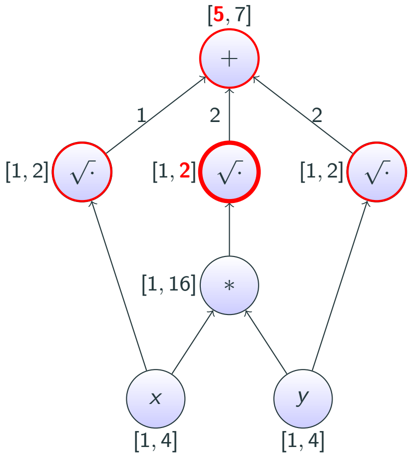
Feasibility-Based Bound Tightening on Expression "Tree"

Example:

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

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





Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

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

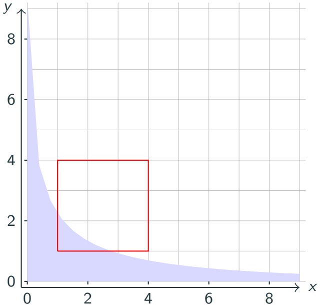
Application of **Interval Arithmetics**

[Moore, 1966]

Example:

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

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

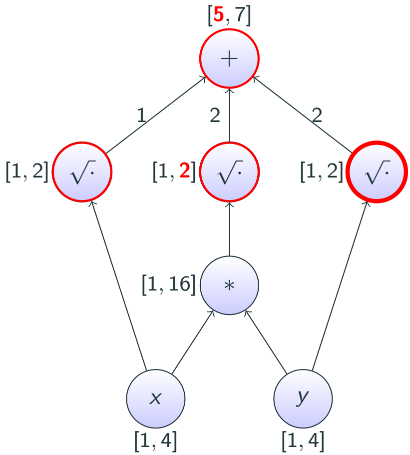


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

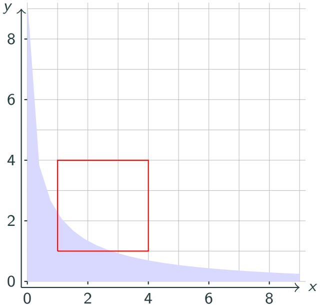
Application of **Interval Arithmetics**

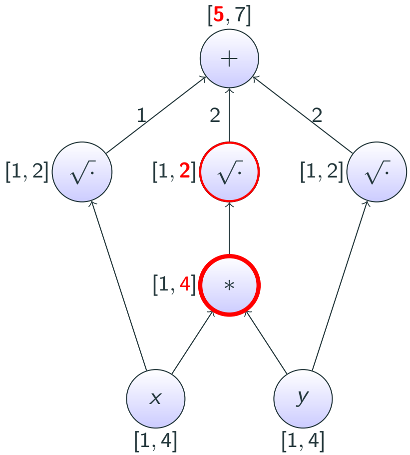
[Moore, 1966]

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)

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

Application of **Interval Arithmetics**

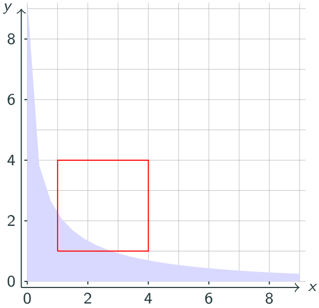
[Moore, 1966]

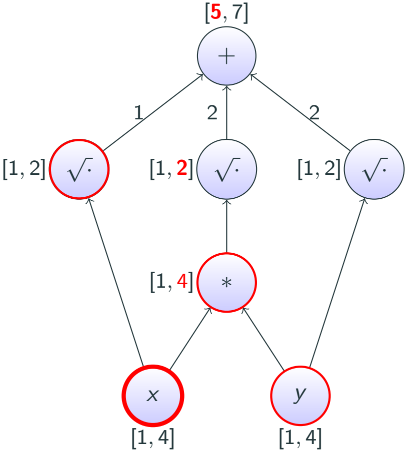
Feasibility-Based Bound Tightening on Expression "Tree"

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**

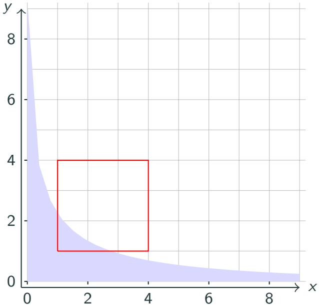
[Moore, 1966]

Feasibility-Based Bound Tightening on Expression “Tree”

Example:

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

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

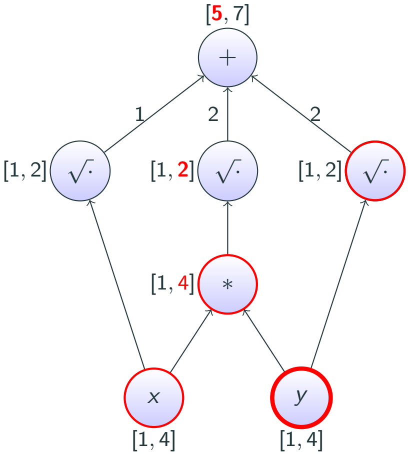


Forward propagation:

- 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

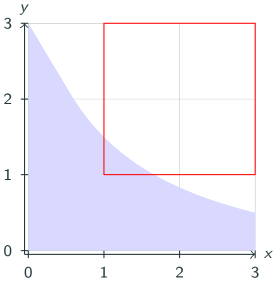
FBBT with Bivariate Quadratics

Example – reformulated:

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

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

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



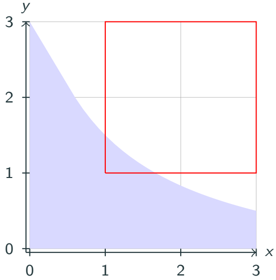
Simple FBBT:

$$x' \leq 7 - 2x'y' - 2y'$$

$$x' \leq (7 - x' - 2y')/(2y')$$

Example – reformulated:
($x' = \sqrt{x}$, $y' = \sqrt{y}$)

$$x' + 2x'y' + 2y' \in [-\infty, 7]$$
$$x', y' \in [1, 3]$$



Simple FBBT:

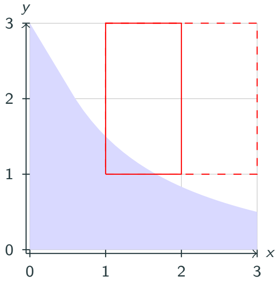
$$\begin{aligned} x' &\leq 7 - 2x'y' - 2y' \\ &\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 \\ x' &\leq (7 - x' - 2y')/(2y') \\ &\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) \end{aligned}$$

Example – reformulated:

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

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

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



Simple FBBT:

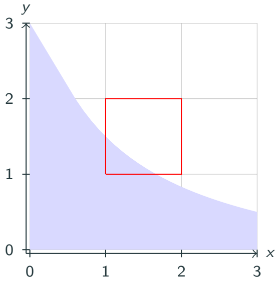
$$\begin{aligned} x' &\le 7 - 2x'y' - 2y' \\ &\le 7 - 2\cdot 1 \cdot 1 - 2 \cdot 1 = \mathbf{3} \\ x' &\le (7 - x' - 2y')/(2y') \\ &\le (7 - 1 - 2\cdot 1)/(2\cdot 1) = \mathbf{2} \end{aligned}$$

FBBT with Bivariate Quadratics

Example – reformulated:

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

$$x' + 2x'y' + 2y' \in [-\infty, 7]$$
$$x', y' \in [1, 3]$$

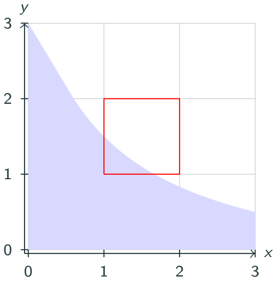


Simple FBBT:

$$\begin{aligned} x' &\le 7 - 2x'y' - 2y' \\ &\le 7 - 2\cdot 1\cdot 1 - 2\cdot 1 = \mathbf{3} \\ x' &\le (7 - x' - 2y')/(2y') \\ &\le (7 - 1 - 2\cdot 1)/(2\cdot 1) = \mathbf{2} \\ y' &\le (7 - 2x'y' - x')/2 \le \mathbf{2} \\ y' &\le (7 - 2y' - x')/(2x') \le \mathbf{2} \end{aligned}$$

Example – reformulated:
($x' = \sqrt{x}$, $y' = \sqrt{y}$)

$$x' + 2x'y' + 2y' \in [-\infty, 7]$$
$$x', y' \in [1, 3]$$



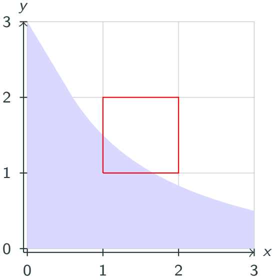
Simple FBBT:

$$\begin{aligned} x' &\leq 7 - 2x'y' - 2y' \\ &\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = \mathbf{3} \\ x' &\leq (7 - x' - 2y')/(2y') \\ &\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = \mathbf{2} \\ y' &\leq (7 - 2x'y' - x')/2 \leq \mathbf{2} \\ y' &\leq (7 - 2y' - x')/(2x') \leq \mathbf{2} \end{aligned}$$

FBBT with Bivariate Quadratics

Example – reformulated:
($x' = \sqrt{x}$, $y' = \sqrt{y}$)

$$x' + 2x'y' + 2y' \in [-\infty, 7]$$
$$x', y' \in [1, 3]$$



Simple FBBT:

$$\begin{aligned} x' &\leq 7 - 2x'y' - 2y' \\ &\leq 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = \mathbf{3} \\ x' &\leq (7 - x' - 2y')/(2y') \\ &\leq (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = \mathbf{2} \\ y' &\leq (7 - 2x'y' - x')/2 \leq \mathbf{2} \\ y' &\leq (7 - 2y' - x')/(2x') \leq \mathbf{2} \end{aligned}$$

Consider Bivariate Quadratic as **one term**
[Vigerske, 2013, Vigerske and Gleixner, 2017]:

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

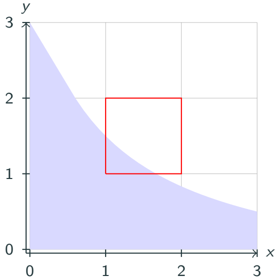
FBBT with Bivariate Quadratics

Example – reformulated:

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

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

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

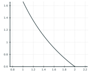


Simple FBBT:

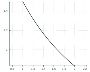
$$\begin{aligned} x' &\le 7 - 2x'y' - 2y' \\ &\le 7 - 2 \cdot 1 \cdot 1 - 2 \cdot 1 = \mathbf{3} \\ x' &\le (7 - x' - 2y')/(2y') \\ &\le (7 - 1 - 2 \cdot 1)/(2 \cdot 1) = \mathbf{2} \\ y' &\le (7 - 2x'y' - x')/2 \le \mathbf{2} \\ y' &\le (7 - 2y' - x')/(2x') \le \mathbf{2} \end{aligned}$$

Consider Bivariate Quadratic as **one term** [Vigerske, 2013, Vigerske and Gleixner, 2017]:

$$x' \le \frac{7 - 2y'}{1 + 2y'}$$



$$y' \le \frac{7 - x'}{2 + 2x'}$$



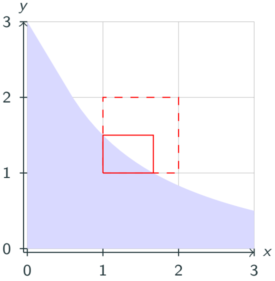
FBBT with Bivariate Quadratics

Example – reformulated:

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

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

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



Simple FBBT:

$$\begin{aligned} x' &\le 7 - 2x'y' - 2y' \\ &\le 7 - 2\cdot 1 \cdot 1 - 2 \cdot 1 = \mathbf{3} \\ x' &\le (7 - x' - 2y')/(2y') \\ &\le (7 - 1 - 2\cdot 1)/(2 \cdot 1) = \mathbf{2} \\ y' &\le (7 - 2x'y' - x')/2 \le \mathbf{2} \\ y' &\le (7 - 2y' - x')/(2x') \le \mathbf{2} \end{aligned}$$

Consider Bivariate Quadratic as **one term**

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

$$x' \le \frac{7 - 2y'}{1 + 2y'} \le \frac{7 - 2\cdot 1}{1 + 2 \cdot 1} = \mathbf{\frac{5}{3}}$$

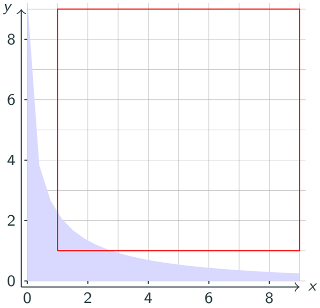
$$y' \le \frac{7 - x'}{2 + 2x'} \le \frac{7 - 1}{2 + 2 \cdot 1} = \mathbf{\frac{3}{2}}$$

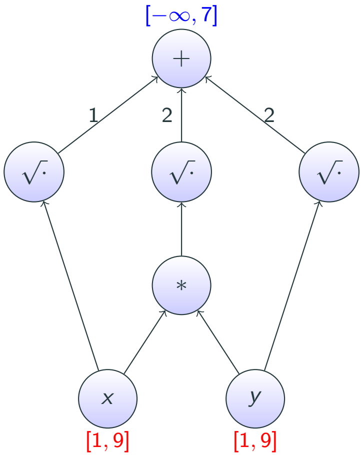
FBBT on Expression Graph

Example:

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

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



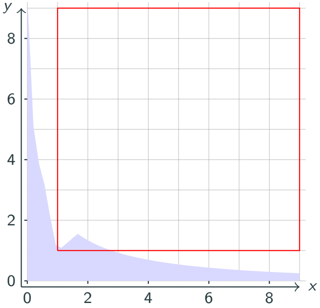


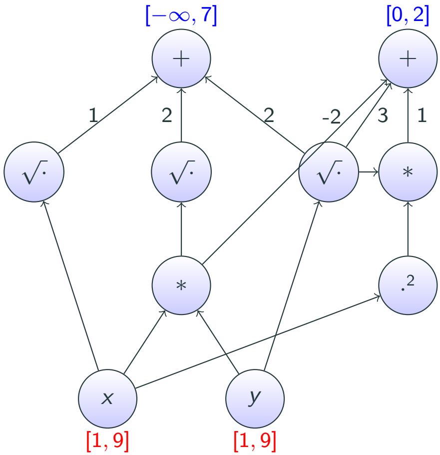
- Common subexpressions from different constraints may stronger boundtightening.

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 boundtightening.

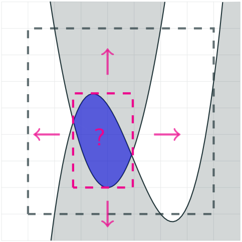
Acceleration – Selected Topics

Acceleration – Selected Topics

Optimization-based bound tightening

Optimization-based bound tightening

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



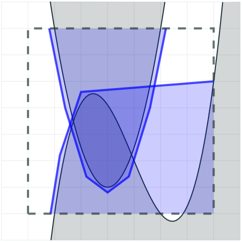
Optimization-based bound tightening

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

[Quesada and Grossmann, 1993, Maranas and Floudas, 1997, Smith and Pantelides, 1999, ...]:

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



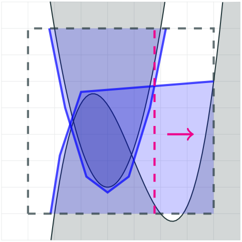
Optimization-based bound tightening

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

Optimization-based Bound Tightening

[Quesada and Grossmann, 1993, Maranas and Floudas, 1997, Smith and Pantelides, 1999, . . .]:

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



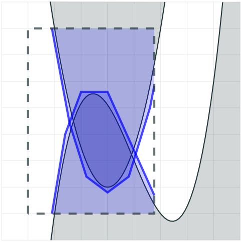
Optimization-based bound tightening

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

Optimization-based Bound Tightening

[Quesada and Grossmann, 1993, Maranas and Floudas, 1997, Smith and Pantelides, 1999, ...]:

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



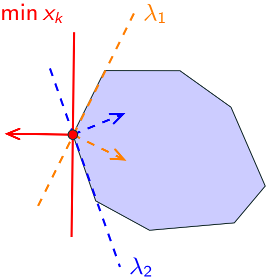
Optimization-based bound tightening

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

[Quesada and Grossmann, 1993, Maranas and Floudas, 1997, Smith and Pantelides, 1999, ...]:

- $\mathcal{R} = \{x : Ax \leq b, c^{\mathsf{T}}x \leq z^*\}$ 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 $x_k \geq \sum_i r_i x_i + \mu z^* + \lambda^{\mathsf{T}} b$
- 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

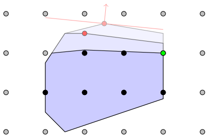
Acceleration – Selected Topics

Synergies with MIP and NLP

MIP ⊂ MINLP

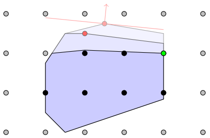
Many MIP techniques can be generalized for MINLP

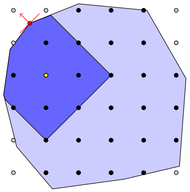
- MIP cutting planes applied to LP relaxation, e.g., Gomory, Mixed-Integer Rounding, Flow Cover
- MIP cutting planes generalized to MINLP, e.g., Disjunctive Cuts [Kilinç, Linderoth, and Luedtke, 2010, Belotti, 2012, Bonami, Linderoth, and Lodi, 2012]



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 [Kılınç, Linderoth, and Luedtke, 2010, Belotti, 2012, Bonami, Linderoth, and Lodi, 2012]
- 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]

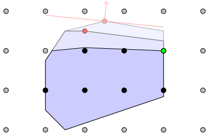


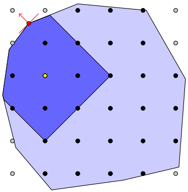


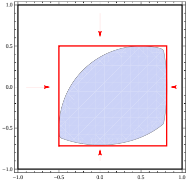
MIP $\subset$ MINLP

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, Belotti, 2012, Bonami, Linderoth, and Lodi, 2012]
- 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]
- Bound Tightening

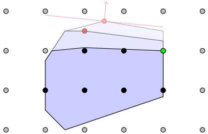


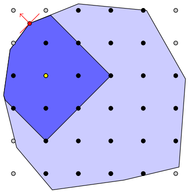


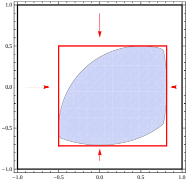


Many MIP techniques can be generalized for MINLP

- MIP cutting planes applied to LP relaxation, e.g., Gomory, Mixed-Integer Rounding, Flow Cover
- MIP cutting planes generalized to MINLP, e.g., Disjunctive Cuts [Kilinç, Linderoth, and Luedtke, 2010, Belotti, 2012, Bonami, Linderoth, and Lodi, 2012]
- 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]
- Bound Tightening
- Symmetry detection and breaking [Liberti, 2012, Liberti and Ostrowski, 2014]



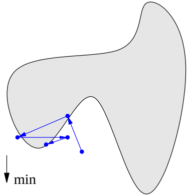




NLP ⊂ MINLP

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



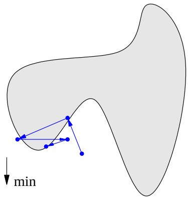
NLP Solvers (finding local optima) are used in MINLP solver

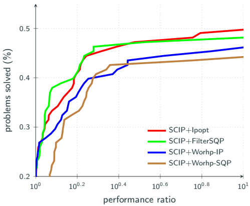
- to find feasible points when integrality and linear constraints are satisfied
- to solve continuous relaxation in NLP-based B&B
- performance of NLP solver is problem-dependent
- some MINLP solvers interface several NLP solvers:

  ANTIGONE: CONOPT, SNOPT

  BARON: FilterSD, FilterSQP, GAMS/NLP (e.g., CONOPT), IPOPT, MINOS, SNOPT

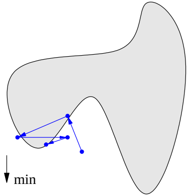
  SCIP (next ver.): FilterSQP, IPOPT, WORHP

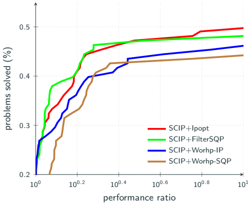




NLP Solvers (finding local optima) are used in MINLP solver

- 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:

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

- $f(x)$ 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:

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

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

Analyze the Algebraic Expression:

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

$$f(x), g(x) \text{ convex} \Rightarrow f(x) + g(x) \text{ convex}$$

$$f(x) \text{ concave} \Rightarrow \log(f(x)) \text{ concave}$$

$$f(x) = \prod_i x_i^{e_i}, x_i \geq 0 \Rightarrow f(x) \begin{cases} \text{convex}, & e_i \leq 0 \ \forall i \\ \text{convex}, & \exists j : e_i \leq 0 \ \forall i \neq j; \ \sum_i e_i \geq 1 \\ \text{concave}, & e_i \geq 0 \ \forall i; \ \sum_i e_i \leq 1 \end{cases}$$

[Maranas and Floudas, 1995, Bao, 2007, Fourer, Maheshwari, Neumaier, Orban, and Schichl, 2009, Vigerske, 2013]

Second Order Cones (SOC)

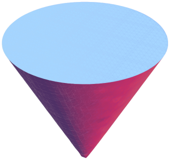
Consider a constraint $\quad x^{\mathsf{T}}Ax + b^{\mathsf{T}}x \leq c.$

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

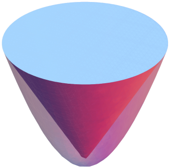
$$\sum_{k=1}^{N} x_k^2 - x_{N+1}^2 \leq 0, x_{N+1} \geq 0 \qquad \Leftrightarrow \qquad \sqrt{\sum_{k=1}^{N} x_k^2} \leq x_{N+1}$$

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

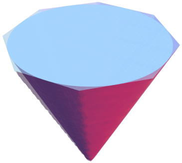
Example: $x^2 + y^2 - z^2 \leq 0$ in $[-1,1] \times [-1,1] \times [0,1]$



feasible region



not recognizing SOC



recognizing SOC

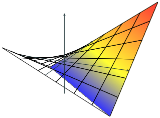
(initial relaxation)

Acceleration – Selected Topics

Convexification

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

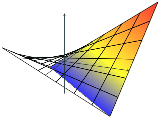
$$\max \left\{ \begin{array}{c} u_x y + u_y x - u_x u_y \\ \ell_x y + \ell_y x - \ell_x \ell_y \end{array} \right\} \leq x \cdot y \leq \min \left\{ \begin{array}{c} u_x y + \ell_y x - u_x \ell_y \\ \ell_x y + u_y x - \ell_x u_y \end{array} \right\}$$



[McCormick, 1976, Al-Khayyal and Falk, 1983]

Convex Envelopes for Product Terms

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



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

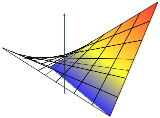
[McCormick, 1976, Al-Khayyal and Falk, 1983]

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 $x \cdot y$ $\quad (x \in [\ell_x, u_x],\ y \in [\ell_y, u_y])$:



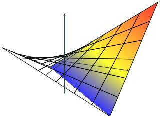
$$\max \left\{ \begin{array}{c} u_x y + u_y x - u_x u_y \\ \ell_x y + \ell_y x - \ell_x \ell_y \end{array} \right\} \leq x{\cdot}y \leq \min \left\{ \begin{array}{c} u_x y + \ell_y x - u_x \ell_y \\ \ell_x y + u_y x - \ell_x u_y \end{array} \right\}$$

[McCormick, 1976, Al-Khayyal and Falk, 1983]

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

Convex Envelopes for Product Terms



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

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

[McCormick, 1976, Al-Khayyal and Falk, 1983]

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

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

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

where $\{v^p : p = 1, \ldots, 2^n\} = \mathrm{vert}([\ell, u])$ 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

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

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

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

The following function classes are vertex-polyhedral:

- Multilinear functions: $f(x) = \sum_{I \in \mathcal{I}} a_I \prod_{i \in I} x_i$, $I \subseteq [n]$ [Rikun, 1997]
- Edge-concave functions: $f(x)$ with $\frac{\partial^2 f}{\partial x_i^2} \leq 0$, $i \in [n]$ [Tardella, 1988/89]

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

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

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

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

The following function classes are vertex-polyhedral:

- Multilinear functions: $f(x) = \sum_{I \in \mathcal{I}} a_I \prod_{i \in I} x_i$, $I \subseteq [n]$ [Rikun, 1997]
- Edge-concave functions: $f(x)$ with $\frac{\partial^2 f}{\partial x_i^2} \leq 0$, $i \in [n]$ [Tardella, 1988/89]

(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

α-Underestimators

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

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

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

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

α-Underestimators

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

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

$$x^\top A x + b^\top x + (u_x - x)^\top \operatorname{diag}(\alpha)(x - \ell_x)$$

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

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

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

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

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

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

A simple choice for α 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]

α-Underestimators

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

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

$$x^\mathsf{T} A x + b^\mathsf{T} x + (u_x - x)^\mathsf{T} \,\mathrm{diag}(\alpha)(x - \ell_x)$$

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

A simple choice for α 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]
- 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

$$x^\mathsf{T} A x + b^\mathsf{T} x = x^\mathsf{T} (A - \mathrm{diag}(\alpha)) x + (b + \mathrm{diag}(\alpha))^\mathsf{T} x$$

for $x \in \{0,1\}^n$ and $A - \mathrm{diag}(\alpha) \succeq 0$. $\Rightarrow$ used in CPLEX, Gurobi

Eigenvalue Reformulation

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

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

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

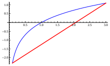
Consider a function $x^\mathsf{T} A x + b^\mathsf{T} x$ with $A \not\succeq 0$.

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

$$\Rightarrow \qquad x^\mathsf{T} A x + b^\mathsf{T} x + c = \sum_{i=1}^{n} \lambda_i (v_i^\mathsf{T} x)^2 + b^\mathsf{T} x + c. \qquad \text{(E)}$$

- introducing auxiliary variables $z_i = v_i^\mathsf{T} x$, function becomes separable:

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

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

Eigenvalue Reformulation

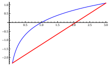
Consider a function $x^\mathsf{T} A x + b^\mathsf{T} x$ with $A \not\succeq 0$.

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

$$\Rightarrow \qquad x^\mathsf{T} A x + b^\mathsf{T} x + c = \sum_{i=1}^{n} \lambda_i (v_i^\mathsf{T} x)^2 + b^\mathsf{T} x + c. \qquad \text{(E)}$$

- introducing auxiliary variables $z_i = v_i^\mathsf{T} x$, function becomes separable:

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

- 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

$$\begin{aligned} \min\ & x^{\mathsf{T}} Q_0 x + b_0^{\mathsf{T}} x & & \text{(quadratic)} \\ \text{s.t. } & x^{\mathsf{T}} Q_k x + b_k^{\mathsf{T}} x \le c_k \qquad k = 1, \dots, q & & \text{(quadratic)} \\ & Ax \le b & & \text{(linear)} \\ & \ell \le x \le u & & \text{(linear)} \end{aligned}$$

Reformulation Linearization Technique (RLT)

Consider the QCQP

$$\begin{aligned}
\min\ & x^\top Q_0 x + b_0^\top x && && \text{(quadratic)}\\
\text{s.t. } & x^\top Q_k x + b_k^\top x \le c_k && k = 1, \dots, q && \text{(quadratic)}\\
& Ax \le b && && \text{(linear)}\\
& \ell \le x \le u && && \text{(linear)}
\end{aligned}$$

Introduce new variables $X_{i,j} = x_i x_j$:

$$\begin{aligned}
\min\ & \langle Q_0, X \rangle + b_0^\top x && && \text{(linear)}\\
\text{s.t. } & \langle Q_k, X \rangle + b_k^\top x \le c_k && k = 1, \dots, q && \text{(linear)}\\
& Ax \le b && && \text{(linear)}\\
& \ell \le x \le u && && \text{(linear)}\\
& X = xx^\top && && \text{(quadratic)}
\end{aligned}$$

Reformulation Linearization Technique (RLT)

Consider the QCQP

$$\begin{aligned}
\min\ & x^\mathsf{T} Q_0 x + b_0^\mathsf{T} x & & \text{(quadratic)}\\
\text{s.t. } & x^\mathsf{T} Q_k x + b_k^\mathsf{T} x \le c_k \qquad k = 1, \ldots, q & & \text{(quadratic)}\\
& Ax \le b & & \text{(linear)}\\
& \ell \le x \le u & & \text{(linear)}
\end{aligned}$$

Introduce new variables $X_{i,j} = x_i x_j$:

$$\begin{aligned}
\min\ & \langle Q_0, X\rangle + b_0^\mathsf{T} x & & \text{(linear)}\\
\text{s.t. } & \langle Q_k, X\rangle + b_k^\mathsf{T} x \le c_k \qquad k = 1, \ldots, q & & \text{(linear)}\\
& Ax \le b & & \text{(linear)}\\
& \ell \le x \le u & & \text{(linear)}\\
& X = xx^\mathsf{T} & & \text{(quadratic)}
\end{aligned}$$

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

- relax $X = xx^\mathsf{T}$ by linear inequalities that are derived from multiplications of pairs of linear constraints

RLT: Multiplying Bound Constraints

Multiplying bounds $\ell_i \leq x_i \leq u_i$ and $\ell_j \leq x_j \leq u_j$ yields

$$(x_i - \ell_i)(x_j - \ell_j) \geq 0$$
$$(x_i - u_i)(x_j - u_j) \geq 0$$
$$(x_i - \ell_i)(x_j - u_j) \leq 0$$
$$(x_i - u_i)(x_j - \ell_j) \leq 0$$

RLT: Multiplying Bound Constraints

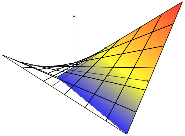
Multiplying bounds $\ell_i \le x_i \le u_i$ and $\ell_j \le x_j \le u_j$ and using $X_{i,j} = x_i x_j$ yields

$$\begin{aligned}
(x_i - \ell_i)(x_j - \ell_j) \ge 0 \quad &\Rightarrow \quad X_{i,j} \ge \ell_i x_j + \ell_j x_i - \ell_i \ell_j \\
(x_i - u_i)(x_j - u_j) \ge 0 \quad &\Rightarrow \quad X_{i,j} \ge u_i x_j + u_j x_i - u_i u_j \\
(x_i - \ell_i)(x_j - u_j) \le 0 \quad &\Rightarrow \quad X_{i,j} \le \ell_i x_j + u_j x_i - \ell_i u_j \\
(x_i - u_i)(x_j - \ell_j) \le 0 \quad &\Rightarrow \quad X_{i,j} \le u_i x_j + \ell_j x_i - u_i \ell_j
\end{aligned}$$

RLT: Multiplying Bound Constraints

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

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

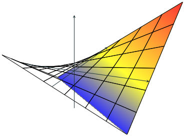


- these are exactly the McCormick inequalities that we have seen earlier

RLT: Multiplying Bound Constraints

Multiplying bounds $\ell_i \le x_i \le u_i$ and $\ell_j \le x_j \le u_j$ and using $X_{i,j} = x_i x_j$ yields

$$
\begin{aligned}
(x_i - \ell_i)(x_j - \ell_j) \ge 0 \quad &\Rightarrow \quad X_{i,j} \ge \ell_i x_j + \ell_j x_i - \ell_i \ell_j \\
(x_i - u_i)(x_j - u_j) \ge 0 \quad &\Rightarrow \quad X_{i,j} \ge u_i x_j + u_j x_i - u_i u_j \\
(x_i - \ell_i)(x_j - u_j) \le 0 \quad &\Rightarrow \quad X_{i,j} \le \ell_i x_j + u_j x_i - \ell_i u_j \\
(x_i - u_i)(x_j - \ell_j) \le 0 \quad &\Rightarrow \quad X_{i,j} \le u_i x_j + \ell_j x_i - u_i \ell_j
\end{aligned}
$$



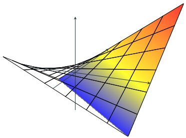
- these are exactly the McCormick inequalities that we have seen earlier
- the resulting linear relaxation is

$$
\begin{aligned}
\min\ & \langle Q_0, X \rangle + b_0^\mathsf{T} x \\
\text{s.t. } & \langle Q_k, X \rangle + b_k^\mathsf{T} x \le c_k \qquad k = 1, \ldots, q \\
& Ax \le b, \quad \ell \le x \le u \\
& X_{i,j} \ge \ell_i x_j + \ell_j x_i - \ell_i \ell_j \qquad i,j = 1, \ldots, n, i \le j \\
& X_{i,j} \ge u_i x_j + u_j x_i - u_i u_j \qquad i,j = 1, \ldots, n, i \le j \\
& X_{i,j} \le \ell_i x_j + u_j x_i - \ell_i u_j \qquad i,j = 1, \ldots, n, \\
& X = X^\mathsf{T}
\end{aligned}
$$

RLT: Multiplying Bound Constraints

Multiplying bounds $\ell_i \le x_i \le u_i$ and $\ell_j \le x_j \le u_j$ and using $X_{i,j} = x_i x_j$ yields

$$(x_i - \ell_i)(x_j - \ell_j) \ge 0 \quad \Rightarrow \quad X_{i,j} \ge \ell_i x_j + \ell_j x_i - \ell_i \ell_j$$
$$(x_i - u_i)(x_j - u_j) \ge 0 \quad \Rightarrow \quad X_{i,j} \ge u_i x_j + u_j x_i - u_i u_j$$
$$(x_i - \ell_i)(x_j - u_j) \le 0 \quad \Rightarrow \quad X_{i,j} \le \ell_i x_j + u_j x_i - \ell_i u_j$$
$$(x_i - u_i)(x_j - \ell_j) \le 0 \quad \Rightarrow \quad X_{i,j} \le u_i x_j + \ell_j x_i - u_i \ell_j$$



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

$$\begin{aligned}
\min\ & \langle Q_0, X\rangle + b_0^{\mathsf{T}} x \\
\text{s.t. } & \langle Q_k, X\rangle + b_k^{\mathsf{T}} x \le c_k && k = 1, \dots, q \\
& Ax \le b, \quad \ell \le x \le u \\
& X_{i,j} \ge \ell_i x_j + \ell_j x_i - \ell_i \ell_j && i,j = 1,\dots,n, i \le j \\
& X_{i,j} \ge u_i x_j + u_j x_i - u_i u_j && i,j = 1,\dots,n, i \le j \\
& X_{i,j} \le \ell_i x_j + u_j x_i - \ell_i u_j && i,j = 1,\dots,n, \\
& X = X^{\mathsf{T}}
\end{aligned}$$

- these inequalities are used by all solvers
- not every solver introduces $X_{i,j}$ variables explicitly

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

$$(A_k^\top x - b_k)(x_j - \ell_j) \geq 0 \quad \Rightarrow \quad \sum_{i=1}^{n} A_{k,i} x_i (x_j - \ell_j) - b_k (x_j - \ell_j) \geq 0$$

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

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

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

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

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

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

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

$$(A_k^\top x - b_k)(A_{k'}^\top x - b_{k'}) \geq 0 \quad \Rightarrow \quad A_k^\top X A_{k'}^\top - (b_k A_{k'} + b_{k'} A_k^\top)x + b_k b_{k'} \geq 0$$

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

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

$$(A_k^\top x - b_k)(A_{k'}^\top x - b_{k'}) \geq 0 \quad \Rightarrow \quad A_k^\top X A_{k'}^\top - (b_k A_{k'} + b_{k'} A_k^\top)x + b_k b_{k'} \geq 0$$

ANTIGONE [Misener and Floudas, 2012]:

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

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

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

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

ANTIGONE [Misener and Floudas, 2012]:

- linear inequality $(A_k^T x - b_k \leq 0)$ $\times$ variable bound $(x_j - \ell_j \geq 0)$
  $\Rightarrow$ consider for cut generation and bound tightening
- linear (in)equality $(A_k^T x - b_k {\leq}/{=} 0)$ $\times$ linear (in)equality $(A_{k'}^T x - b_{k'} {\leq}/{=} 0)$
  $\Rightarrow$ consider for cut generation and bound tightening

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

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

$$(A_k^\top x - b_k)(A_{k'}^\top x - b_{k'}) \geq 0 \quad \Rightarrow \quad A_k^\top X A_{k'}^\top - (b_k A_{k'} + b_{k'} A_k^\top)x + b_k b_{k'} \geq 0$$

ANTIGONE [Misener and Floudas, 2012]:

- linear inequality $(A_k^T x - b_k \leq 0)$ $\times$ variable bound $(x_j - \ell_j \geq 0)$
  $\Rightarrow$ consider for cut generation and bound tightening
- linear (in)equality $(A_k^T x - b_k \leq/{=} 0)$ $\times$ linear (in)equality $(A_{k'}^T x - b_{k'} \leq/{=} 0)$
  $\Rightarrow$ consider for cut generation and bound tightening
- linear equality $(A_k^\top x - b_k = 0)$ $\times$ 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])

RLT: Multiplying Bounds and Inequalities

Additional inequalities are derived by multiplying pairs of linear equations and bound constraints:

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

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

ANTIGONE [Misener and Floudas, 2012]:

- linear inequality $(A_k^T x - b_k \leq 0)$ $\times$ variable bound $(x_j - \ell_j \geq 0)$
  $\Rightarrow$ consider for cut generation and bound tightening
- linear (in)equality $(A_k^T x - b_k {\leq}/{=} 0)$ $\times$ linear (in)equality $(A_{k'}^T x - b_{k'} {\leq}/{=} 0)$
  $\Rightarrow$ consider for cut generation and bound tightening
- linear equality $(A_k^\mathsf{T} x - b_k = 0)$ $\times$ 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

$$\begin{aligned} \min\ & x^\top Q_0 x + b_0^\top x \\ \text{s.t. } & x^\top Q_k x + b_k^\top x \le c_k \\ & Ax \le b \\ & \ell_x \le x \le u_x \end{aligned} \qquad \Leftrightarrow \qquad \begin{aligned} \min\ & \langle Q_0, X\rangle + b_0^\top x \\ \text{s.t. } & \langle Q_k, X\rangle + b_k^\top x \le c_k \\ & Ax \le b \\ & \ell_x \le x \le u_x \\ & X = xx^\top \end{aligned}$$

- relaxing $X - xx^\top = 0$ to $X - xx^\top \succeq 0$, which is equivalent to

$$\tilde{X} := \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0,$$

yields a semidefinite programming relaxation

Semidefinite Programming (SDP) Relaxation

$$\begin{aligned} \min\ & x^\mathsf{T} Q_0 x + b_0^\mathsf{T} x \\ \text{s.t. } & x^\mathsf{T} Q_k x + b_k^\mathsf{T} x \le c_k \\ & Ax \le b \\ & \ell_x \le x \le u_x \end{aligned} \quad \Leftrightarrow \quad \begin{aligned} \min\ & \langle Q_0, X\rangle + b_0^\mathsf{T} x \\ \text{s.t. } & \langle Q_k, X\rangle + b_k^\mathsf{T} x \le c_k \\ & Ax \le b \\ & \ell_x \le x \le u_x \\ & X = xx^\mathsf{T} \end{aligned}$$

- relaxing $X - xx^\mathsf{T} = 0$ to $X - xx^\mathsf{T} \succeq 0$, which is equivalent to

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

  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

$$\langle v, \tilde{X} v\rangle \ge 0$$

  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

SDP vs RLT vs α-BB

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

SDP vs RLT vs α-BB

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 α-BB underestimators

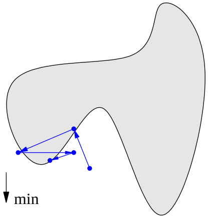
Acceleration – Selected Topics

Primal Heuristics

Sub-NLP Heuristics

Given a solution satisfying all integrality constraints,

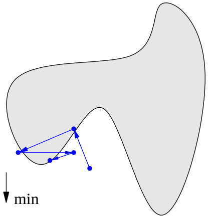
- fix all integer variables in the MINLP
- call an NLP solver to find a local solution to the remaining NLP



Sub-NLP Heuristics

Given a solution satisfying all integrality constraints,

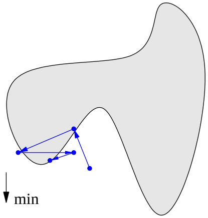
- fix all integer variables in the MINLP
- call an NLP solver to find a local solution to the remaining NLP
- variable fixings given by integer-feasible solution to LP relaxation
- additionally, SCIP runs its MIP heuristics on MIP relaxation (rounding, diving, feas. pump, LNS, . . .)



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, ...)

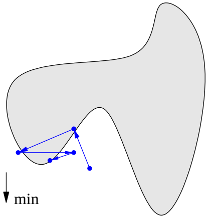


NLP-Diving: solve NLP relaxation, restrict bounds on fractional variable, repeat

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

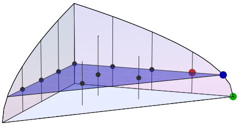
Multistart: run local NLP solver from random starting points to increase likelihood of finding global optimum

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])

Sub-MIP / Sub-MINLP Heuristics

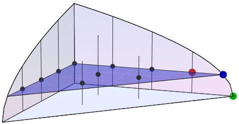
"Undercover" (SCIP) [Berthold and Gleixner, 2014]:

- 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



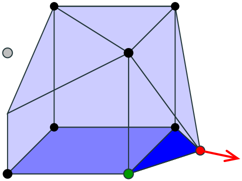
"Undercover" (SCIP) [Berthold and Gleixner, 2014]:

- 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

End.

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

K. Abhishek, S. Leyffer, and J. T. Linderoth. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. *INFORMS Journal On Computing*, 22(4):555–567, 2010. doi:10.1287/ijoc.1090.0373.

Tobias Achterberg. SCIP: Solving Constraint Integer Programs. *Mathematical Programming Computation*, 1(1):1–41, 2009. doi:10.1007/s12532-008-0001-1.

Warren P. Adams and Hanif D. Sherali. A tight linearization and an algorithm for zero-one quadratic programming problems. *Management Science*, 32(10):1274–1290, 1986. doi:10.1287/mnsc.32.10.1274.

Claire S. Adjiman and Christodoulos A. Floudas. Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization*, 9(1):23–40, 1996. doi:10.1007/BF00121749.

Claire S. Adjiman, I. P. Androulakis, and Christodoulos A. Floudas. A global optimization method, αBB, for general twice-differentiable constrained NLPs – II. Implementation and computational results. *Computers & Chemical Engineering*, 22(9):1159–1179, 1998a.

Literature ii

Claire S. Adjiman, S. Dallwig, Christodoulos A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical advances. *Computers & Chemical Engineering*, 22:1137–1158, 1998b. doi:10.1016/S0098-1354(98)00027-1.

Faiz A. Al-Khayyal and James E. Falk. Jointly constrained biconvex programming. *Mathematics of Operations Research*, 8(2):273–286, 1983. doi:10.1287/moor.8.2.273.

Ioannis P. Androulakis, Costas D. Maranas, and Christodoulos A. Floudas. αBB: A global optimization method for general constrained nonconvex problems. *Journal of Global Optimization*, 7(4):337–363, 1995. doi:10.1007/BF01099647.

Kurt Anstreicher. Semidefinite programming versus the reformulation-linearization technique for nonconvex quadratically constrained quadratic programming. *Journal of Global Optimization*, 43(2):471–484, 2009. ISSN 0925-5001. doi:10.1007/s10898-008-9372-0.

Kurt Anstreicher. On convex relaxations for quadratically constrained quadratic programming. *Mathematical Programming*, 136(2):233–251, 2012. doi:10.1007/s10107-012-0602-3.

X. Bao. Automatic convexity detection for global optimization. Master's thesis, University of Illinois at Urbana-Champaign, 2007.

X. Bao, N. V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically-constrained quadratic programs. *Optimization Methods and Software*, 24(4-5): 485–504, 2009. doi:10.1080/10556780902883184.

Pietro Belotti. Disjunctive cuts for non-convex MINLP. In Lee and Leyffer [2012], pages 117–144. doi:10.1007/978-1-4614-1927-3_5.

Pietro Belotti. Bound reduction using pairs of linear inequalities. *Journal of Global Optimization*, 56(3):787–819, 2013. doi:10.1007/s10898-012-9848-9.

Pietro Belotti and Timo Berthold. Three ideas for a feasibility pump for nonconvex minlp. *Optimization Letters*, 11(1):3–15, 2017. doi:10.1007/s11590-016-1046-0.

Pietro Belotti, Jon Lee, Leo Liberti, F. Margot, and Andreas Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5): 597–634, 2009. doi:10.1080/10556780903087124.

Pietro Belotti, Sonia Cafieri, Jon Lee, and Leo Liberti. Feasibility-based bounds tightening via fixed points. In Weili Wu and Ovidiu Daescu, editors, *Combinatorial Optimization and Applications*, volume 6508 of *Lecture Notes in Computer Science*, pages 65–76. Springer, Berlin/Heidelberg, 2010. doi:10.1007/978-3-642-17458-2_7.

Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, Jim Luedtke, and Ashutosh Mahajan. Mixed-integer nonlinear optimization. *Acta Numerica*, 22:1–131, 2013. doi:10.1017/S0962492913000032.

F. Benhamou, C. Bliek, B. Faltings, L. Granvilliers, E. Huens, E. Monfroy, A. Neumaier, D. Sam-Haroud, P. Spellucci, P. Van Hentenryck, and L. Vicente. Algorithms for solving nonlinear constrained and optimization problems: The state of the art. Technical report, Universität Wien, Fakultät für Mathematik, 2001. URL `http://www.mat.univie.ac.at/~neum/glopt/coconut/StArt.html`.

Timo Berthold. *Heuristic algorithms in global MINLP solvers*. PhD thesis, TU Berlin, 2014a.

Timo Berthold. RENS – the optimal rounding. *Mathematical Programming Computation*, 6(1): 33–54, 2014b. doi:10.1007/s12532-013-0060-9.

Timo Berthold and Ambros M. Gleixner. Undercover: a primal MINLP heuristic exploring a largest sub-MIP. *Mathematical Programming*, 144(1-2):315–346, 2014. doi:10.1007/s10107-013-0635-2.

Timo Berthold, Stefan Heinz, Marc E. Pfetsch, and Stefan Vigerske. Large neighborhood search beyond MIP. In Luca Di Gaspero, Andrea Schaerf, and Thomas Stützle, editors, *Proceedings of the 9th Metaheuristics International Conference (MIC 2011)*, pages 51–60, 2011. urn:nbn:de:0297-zib-12989.

Christian Bliek, Pierre Bonami, and Andrea Lodi. Solving mixed-integer quadratic programming problems with IBM-CPLEX: A progress report. In *Proceedings of the Twenty-Sixth RAMP Symposium*, pages 171–190, 2014.

Pierre Bonami and João P. M. Gonçalves. Heuristics for convex mixed integer nonlinear programs. *Computational Optimization and Applications*, 51(2):729–747, 2012. doi:10.1007/s10589-010-9350-6.

Pierre Bonami, Lorenz T. Biegler, Andrew R. Conn, G. Cornuéjols, Ignacio E. Grossmann, Carl D. Laird, Jon Lee, Andrea Lodi, F. Margot, Nicolas Sawaya, and Andreas Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5(2):186–204, 2008. doi:10.1016/j.disopt.2006.10.011.

Pierre Bonami, G. Cornuéjols, Andrea Lodi, and F. Margot. A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming*, 119(2):331–352, 2009. doi:10.1007/s10107-008-0212-2.

Pierre Bonami, Jeff Linderoth, and Andrea Lodi. Disjunctive cuts for mixed integer nonlinear programming problems. In A. Ridha Mahjoub, editor, *Progress in Combinatorial Optimization*, chapter 18, pages 521–541. ISTe-Wiley, 2012.

Fani Boukouvala, Ruth Misener, and Christodoulos A. Floudas. Global optimization advances in mixed-integer nonlinear programming, MINLP, and constrained derivative-free optimization, CDFO. *European Journal of Operational Research*, 252(3):701–727, 2016. doi:10.1016/j.ejor.2015.12.018.

Samuel Burer and Adam N. Letchford. Non-convex mixed-integer nonlinear programming: A survey. *Surveys in Operations Research and Management Science*, 17(2):97–106, 2012. doi:10.1016/j.sorms.2012.08.001.

Michael R. Bussieck and Arne S. Drud. SBB: A new solver for mixed integer nonlinear programming. Talk at OR2001, 2001.

Richard H. Byrd, Jorge Nocedal, and Richard A. Waltz. KNITRO: An integrated package for nonlinear optimization. In G. di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, volume 83 of *Nonconvex Optimization and Its Applications*, pages 35–59. Springer, 2006. doi:10.1007/0-387-30065-1_4.

Sonia Cafieri, Jon Lee, and Leo Liberti. On convex relaxations of quadrilinear terms. *Journal of Global Optimization*, 47(4):661–685, 2010. doi:10.1007/s10898-009-9484-1.

Claudia D'Ambrosio, Antonio Frangioni, Leo Liberti, and Andrea Lodi. Experiments with a feasibility pump approach for non-convex MINLPs. In Paola Festa, editor, *Proceedings of 9th International Symposium on Experimental Algorithms, SEA 2010*, volume 6049 of *Lecture Notes in Computer Science*, pages 350–360. Springer, 2010. doi:10.1007/978-3-642-13193-6_30.

Claudia D'Ambrosio, Antonio Frangioni, Leo Liberti, and Andrea Lodi. A storm of feasibility pumps for nonconvex MINLP. *Mathematical Programming*, 136(2):375–402, 2012. doi:10.1007/s10107-012-0608-x.

Marco A. Duran and Ignacio E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36(3):307–339, 1986. doi:10.1007/BF02592064.

FICO. *Xpress-SLP Program Reference Manual*, 1.41 edition, 2008.

Robert Fourer, Chandrakant Maheshwari, Arnold Neumaier, Dominique Orban, and Hermann Schichl. Convexity and concavity detection in computational graphs: Tree walks for convexity assessment. *INFORMS Journal on Computing*, 22(1):26–43, 2009. doi:10.1287/ijoc.1090.0321.

Ambros M. Gleixner, Timo Berthold, Benjamin Müller, and Stefan Weltge. Three enhancements for optimization-based bound tightening. *Journal of Global Optimization*, 67:731–757, 2017. doi:10.1007/s10898-016-0450-4.

Ralph E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bull. AMS*, 64 (5):275–278, 1958.

J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, 8(4):703–712, 1960. doi:10.1137/0108053.

Mustafa Kilinç, Jeff Linderoth, and Jim Luedtke. Effective separation of disjunctive cuts for convex mixed integer nonlinear programs. Technical Report 1681, University of Wisconsin-Madison, Computer Sciences Department, 2010. URL `http://digital.library.wisc.edu/1793/60720`.

Gary R. Kocis and Ignacio E. Grossmann. Computational experience with DICOPT solving MINLP problems in process systems engineering. *Computers & Chemical Engineering*, 13(3):307–315, 1989. doi:10.1016/0098-1354(89)85008-2.

Jan Kronqvist, Andreas Lundell, and Tapio Westerlund. The extended supporting hyperplane algorithm for convex mixed-integer nonlinear programming. *Journal of Global Optimization*, 64 (2):249–272, 2016. doi:10.1007/s10898-015-0322-3.

Alisa H. Land and Alison G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960. doi:10.2307/1910129.

Toni Lastusilta. *GAMS MINLP Solver Comparison and Some Improvements to the AlphaECP Algorithm*. PhD thesis, Process Design and Systems Engineering Laboratory, Åbo Akademi University, 2011. urn:nbn:fi-fe201311267417.

Jon Lee and Sven Leyffer, editors. *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*. Springer, 2012. doi:10.1007/978-1-4614-1927-3.

S. Leyffer. Integrating SQP and branch and bound for mixed-integer nonlinear programming. Technical Report NA182, Dundee University, 1998.

Sven Leyffer. *Deterministic Methods for Mixed-Integer Nonlinear Programming*. PhD thesis, Department of Mathematics and Computer Science, University of Dundee, 1993.

Leo Liberti. Reformulations in mathematical programming: automatic symmetry detection and exploitation. *Mathematical Programming*, 131(1-2):273–304, 2012. doi:10.1007/s10107-010-0351-0.

Leo Liberti and James Ostrowski. Stabilizer-based symmetry breaking constraints for mathematical programs. *Journal of Global Optimization*, 60(2):183–194, 2014. doi:10.1007/s10898-013-0106-6.

Leo Liberti and Constantinos C. Pantelides. An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *Journal of Global Optimization*, 36:161–189, 2006.

Youdong Lin and Linus Schrage. The global solver in the LINDO API. *Optimization Methods & Software*, 24(4–5):657–668, 2009. doi:10.1080/10556780902753221.

Marco Locatelli and Fabio Schoen. *Global Optimization: Theory, Algorithms, and Applications*. Number 15 in MOS-SIAM Series on Optimization. SIAM, 2013.

A. Mahajan, S. Leyffer, and T. Munson. MINOTAUR: A new framework for solving nonlinear programs. Talk at INFORMS Annual Conference, 2009.

Ashutosh Mahajan and Todd Munson. Exploiting second-order cone structure for global optimization. Technical Report ANL/MCS-P1801-1010, Argonne National Laboratory, 2010. URL `http://www.optimization-online.org/DB_HTML/2010/10/2780.html`.

Ashutosh Mahajan, Sven Leyffer, and Christian Kirches. Solving mixed-integer nonlinear programs by QP-diving. Preprint ANL/MCS-P2071-0312, Argonne National Laboratory, 2012. URL `http://www.optimization-online.org/DB_HTML/2012/03/3409.html`.

Stephen J. Maher, Tobias Fischer, Tristan Gally, Gerald Gamrath, Ambros Gleixner, Robert Lion Gottwald, Gregor Hendel, Thorsten Koch, Marco E. Lübbecke, Matthias Miltenberger, Benjamin Müller, Marc E. Pfetsch, Christian Puchert, Daniel Rehfeldt, Sebastian Schenker, Robert Schwarz, Felipe Serrano, Yuji Shinano, Dieter Weninger, Jonas T. Witt, and Jakob Witzig. The SCIP Optimization Suite 4.0. ZIB Report 17-12, Zuse Institute Berlin, 2017. urn:nbn:de:0297-zib-62170.

Costas D. Maranas and Christodoulos A. Floudas. Finding all solutions of nonlinearly constrained systems of equations. *Journal of Global Optimization*, 7(2):143–182, 1995. doi:10.1007/BF01097059.

Costas D. Maranas and Christodoulos A. Floudas. Global optimization in generalized geometric programming. *Computers & Chemical Engineering*, 21(4):351–369, 1997. doi:10.1016/S0098-1354(96)00282-7.

Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I – convex underestimating problems. *Mathematical Programming*, 10(1):147–175, 1976. doi:10.1007/BF01580665.

Clifford A. Meyer and Christodoulos A. Floudas. Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *Journal of Global Optimization*, 29(2):125–155, 2004. doi:10.1023/B:JOGO.0000042112.72379.e6.

Clifford A. Meyer and Christodoulos A. Floudas. Convex envelopes for edge-concave functions. *Mathematical Programming*, 103(2):207–224, 2005. doi:10.1007/s10107-005-0580-9.

Literature x

Ruth Misener and Christodoulos A. Floudas. GloMIQO: Global mixed-integer quadratic optimizer. *Journal of Global Optimization*, 57(1):3–50, 2012. doi:10.1007/s10898-012-9874-7.

Ruth Misener and Christodoulos A. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 59(2-3):503–526, 2014. doi:10.1007/s10898-014-0166-2.

Ramon E. Moore. *Interval Analysis*. Englewood Cliffs, NJ: Prentice Hall, 1966.

Benjamin Müller, Renke Kuhlmann, and Stefan Vigerske. On the performance of NLP solvers within global MINLP solvers. ZIB Report 17-35, Zuse Institute Berlin, 2017. urn:nbn:de:0297-zib-64504.

Giacomo Nannicini and Pietro Belotti. Rounding-based heuristics for nonconvex MINLPs. *Mathematical Programming Computation*, 4(1):1–31, 2012. ISSN 1867-2949. doi:10.1007/s12532-011-0032-x.

Ivo P. Nenov, Daniel H. Fylstra, and Lubomir V. Kolev. Convexity determination in the Microsoft Excel solver using automatic differentiation techniques. Extended abstract, Frontline Systems Inc., 2004. URL `http://www.autodiff.org/ad04/abstracts/Nenov.pdf`.

Arnold Neumaier. Constrained global optimization. In *Algorithms for Solving Nonlinear Constrained and Optimization Problems: The State of The Art* Benhamou et al. [2001], chapter 4, pages 55–111. URL `http://www.mat.univie.ac.at/~neum/glopt/coconut/StArt.html`.

Andrea Qualizza, Pietro Belotti, and François Margot. Linear programming relaxations of quadratically constrained quadratic programs. In Lee and Leyffer [2012], pages 407–426. doi:10.1007/978-1-4614-1927-3_14.

Ignacio Quesada and Ignacio E. Grossmann. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Computers & Chemical Engineering*, 16(10-11):937–947, 1992. doi:10.1016/0098-1354(92)80028-8.

Ignacio Quesada and Ignacio E. Grossmann. Global optimization algorithm for heat exchanger networks. *Industrial & Engineering Chemistry Research*, 32(3):487–499, 1993. doi:10.1021/ie00015a012.

Anatoliy D. Rikun. A convex envelope formula for multilinear functions. *Journal of Global Optimization*, 10(4):425–437, 1997. doi:10.1023/A:1008217604285.

M. Roelofs and J. Bisschop. Aimms outer approximation algorithm for MINLP. In *AIMMS - The Language Reference*, chapter 18. 2017.

Nikolaos V. Sahinidis. BARON: A general purpose global optimization software package. *Journal of Global Optimization*, 8(2):201–205, 1996. doi:10.1007/BF00138693.

Hanif D. Sherali and W. P. Adams. *A Reformulation-Linearization Technique for Solving Discrete and Continuous Nonconvex Problems*, volume 31 of *Nonconvex Optimization and Its Applications*. Kluwer Academic Publishers, 1999. ISBN 978-0-7923-5487-1.

Hanif D. Sherali and Amine Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization*, 2(4):379–410, 1992. doi:10.1007/BF00122429.

Hanif D. Sherali and Barbara M. P. Fraticelli. Enhancing RLT relaxations via a new class of semidefinite cuts. *Journal of Global Optimization*, 22(1):233–261, 2002. ISSN 0925-5001. doi:10.1023/A:1013819515732.

Edward M. B. Smith and Constantinos C. Pantelides. Global optimization of general process models. In I. E. Grossmann, editor, *Global Optimization in Engineering Design*, volume 9 of *Nonconvex Optimization and Its Applications*, pages 355–386. Kluwer Academic Publishers, 1996. doi:10.1007/978-1-4757-5331-8_12.

Edward M. B. Smith and Constantinos C. Pantelides. Global optimisation of nonconvex MINLPs. *Computers & Chemical Engineering*, 21(suppl.):S791–S796, 1997. doi:10.1016/S0098-1354(97)87599-0.

Edward M. B. Smith and Constantinos C. Pantelides. A symbolic reformulation/spatial branch-and-bound algorithm for the global optimization of nonconvex MINLPs. *Computers & Chemical Engineering*, 23(4-5):457–478, 1999. doi:10.1016/S0098-1354(98)00286-5.

Laurence Smith, John Chinneck, and Victor Aitken. Improved constraint consensus methods for seeking feasibility in nonlinear programs. *Computational Optimization and Applications*, 54(3): 555–578, 2013. doi:10.1007/s10589-012-9473-z.

Fabio Tardella. On a class of functions attaining their maximum at the vertices of a polyhedron. *Discrete Applied Mathematics*, 22(2):191–195, 1988/89. doi:10.1016/0166-218X(88)90093-5.

Mohit Tawarmalani and Nikolaos V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103(2):225–249, 2005. doi:10.1007/s10107-005-0581-8.

Zsolt Ugray, Leon Lasdon, John Plummer, Fred Glover, Jim Kelly, and Rafael Martí. Scatter search and local NLP solvers: A multistart framework for global optimization. *INFORMS Journal on Computing*, 19(3):328–340, 2007. doi:10.1287/ijoc.1060.0175.

Arthur F. Veinott. The supporting hyperplane method for unimodal programming. *Operations Research*, 15(1):147–152, feb 1967. doi:10.1287/opre.15.1.147.

Stefan Vigerske. *Decomposition of Multistage Stochastic Programs and a Constraint Integer Programming Approach to Mixed-Integer Nonlinear Programming*. PhD thesis, Humboldt-Universität zu Berlin, 2013. urn:nbn:de:kobv:11-100208240.

Stefan Vigerske and Ambros Gleixner. SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, to appear, 2017. doi:10.1080/10556788.2017.1335312.

Tapio Westerlund and K. Lundquist. Alpha-ECP, version 5.101. an interactive MINLP-solver based on the extended cutting plane method. Technical Report 01-178-A, Process Design Laboratory, Åbo Akademi University, 2005. URL `http://www.abo.fi/~twesterl/A-ECPManual.pdf`.

Tapio Westerlund and Frank Petterson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 19(suppl.):131–136, 1995. doi:10.1016/0098-1354(95)87027-X.