Global Optimization of Mixed-Integer Nonlinear Programs

Stefan Vigerske

Kyushu University · January 20, 2023

Outline

Introduction

Fundamental Methods

- Mixed-Integer Linear Programming
- Convex MINLP
- Nonconvex MINLP

Example

Further Techniques

- Dual Side (Tighter Relaxations)
- Primal Side (Find Feasible Solutions)

Solver Software

- Solvers for Mixed-Integer Quadratic Programs
- Solvers for Convex MINLP
- Solvers for General MINLP

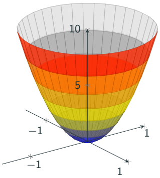
Introduction

Mixed-Integer Nonlinear Programs (MINLPs)

We consider

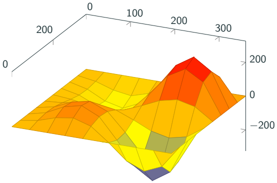
$$\begin{aligned} \min\ & c^{\mathsf{T}} x \\ \text{s.t. } & g_k(x) \le 0 && \forall k \in [m] \\ & x_i \in \mathbb{Z} && \forall i \in \mathcal{I} \subseteq [n] \\ & x_i \in [\ell_i, u_i] && \forall i \in [n] \end{aligned}$$

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



convex

or



nonconvex

Examples of Mixed-Integer Nonlinearities

- Water treatment unit - variable fraction $p \in [0,1]$ of variable quantity q: qp, and valve on/off state $z \in \{0,1\}$

- AC power flow - nonlinear function of voltage magnitudes and angles and binary decisions on switching status of power lines

$$p_{ij} = g_{ij} v_i^2 - g_{ij} v_i v_j \cos(\theta_{ij}) + b_{ij} v_i v_j \sin(\theta_{ij})$$

- Circle packing - non-overlap constraints

$$\|x - y\|_2 \geq r_x + r_y$$

- etc.

Solving a Mixed-Integer Nonlinear Optimization Problem

Two major tasks:

1. Finding and improving feasible solutions (primal side)
   - Ensure feasibility, sacrifice optimality
   - Important for practical applications
2. Proving optimality (dual side)
   - Ensure optimality, sacrifice feasibility
   - Necessary in order to actually solve the problem

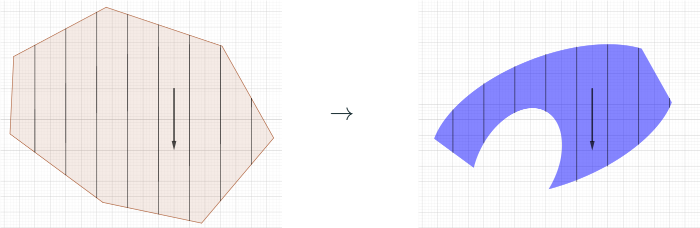
Connected by:

3. Strategy
   - Ensure convergence
   - Divide: branching, decompositions, ...
   - Put together all components

Adding Nonlinearity to a MIP Brings New Challenges

- More numerical issues
- NLP solvers are less efficient and reliable than LP solvers

1. Finding feasible solutions
   - Feasible solutions must also satisfy nonlinear constraints
   - If nonconvex: fixing integer variables and solving the NLP can produce local optima
2. Proving optimality
   - NLP or LP relaxations?
   - If nonconvex: continuous relaxation no longer provides a lower bound
   - "Convenient" descriptions of the feasible set are important
3. Strategy
   - Need to account for all of the above
   - Warmstart for NLP is much less efficient than for LP



Convex MINLP:

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

Fundamental Methods

Fundamental Methods

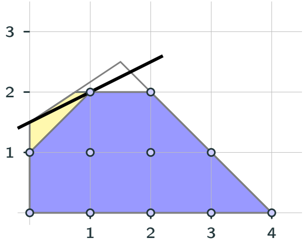
Mixed-Integer Linear Programming

MIP Branch & Cut

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

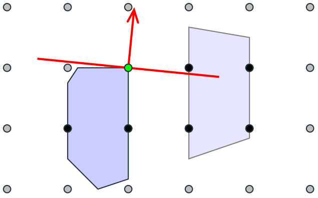
$$\min c^{\mathsf{T}} x,$$
$$\text{s.t. } Ax \leq b,$$
$$x_i \in \mathbb{Z}, \quad i \in \mathcal{I},$$

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



cutting planes
[Gomory, 1958]

&



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

Fundamental Methods

Convex MINLP

Relaxations

Key task: describe the feasible set in a convenient way.

Requirement: the relaxed problem should be efficiently solvable to global optimality.

It is preferable to have relaxations that are:

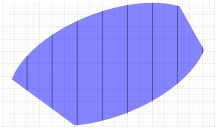
- Convex: NLP solutions are globally optimal, infeasibility detection is reliable
- Linear: solving is more efficient, good for warmstarting

and to avoid:

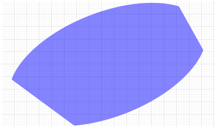
- Very large numbers of constraints and variables
- Bad numerics

Relaxations for Convex MINLPs

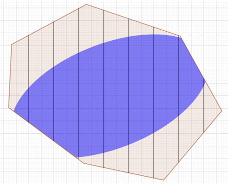
- Relax integrality $\rightarrow$ NLP relaxation



$\longrightarrow$

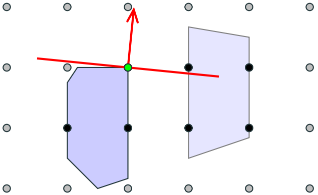


- Replace nonlinear set with linear outer approximation $\rightarrow$ MIP relaxation



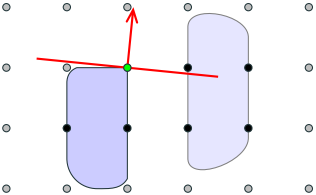
- Linear outer approximation + relax integrality $\rightarrow$ LP relaxation

NLP-based Branch & Bound (NLP-BB)



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

$\Rightarrow$

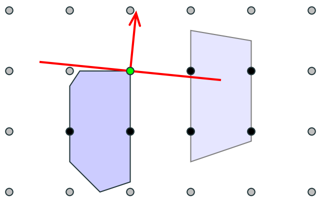


MINLP branch-and-bound
[Leyffer, 1993]

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

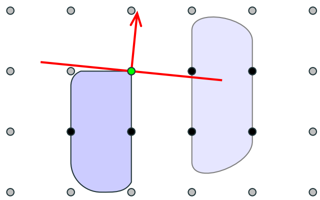
Branching: Subdivide problem along variables x_i, $i \in \mathcal{I}$, that take fractional value in NLP solution.

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.)

$\Rightarrow$ Mahajan, Leyffer, and Kirches [2012]: approximate NLP solves by QPs (hot-start possible)

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

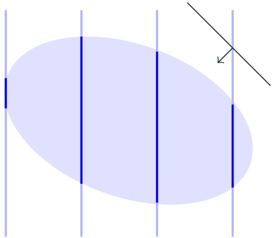
$$\begin{aligned} \min\ & c^\top x, \\ \text{s.t. } & g_k(\hat{x}) + \nabla g_k(\hat{x})^\top (x - \hat{x}) \le 0, \\ & \quad k \in [m], \quad \hat{x} \in R, \\ & x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ & x \in [\ell, u], \end{aligned}$$

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:

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



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

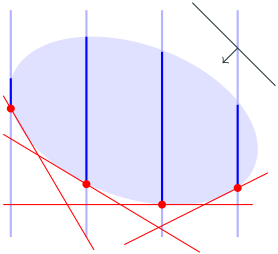
$$\begin{aligned} &\min c^{\mathsf{T}} x, \\ &\text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^{\mathsf{T}}(x - \hat{x}) \le 0, \\ &\qquad k \in [m], \quad \hat{x} \in R, \\ &\quad x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ &\quad x \in [\ell, u], \end{aligned}$$

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

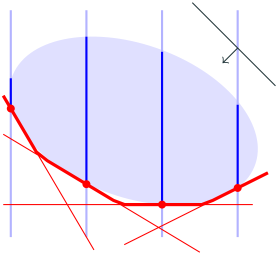
$$\begin{aligned} & \min\ c^{\mathsf{T}} x, \\ & \text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^{\mathsf{T}}(x - \hat{x}) \le 0, \\ & \qquad k \in [m], \quad \hat{x} \in R, \\ & \quad x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ & \quad x \in [\ell, u], \end{aligned}$$

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

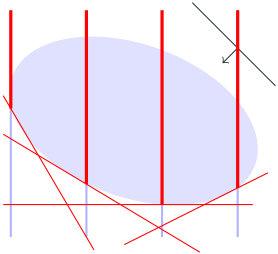
$$\begin{aligned} &\min\ c^{\mathsf{T}} x, \\ &\text{s.t. } g_k(\hat{x}) + \nabla g_k(\hat{x})^{\mathsf{T}}(x - \hat{x}) \leq 0, \\ &\qquad k \in [m], \quad \hat{x} \in R, \\ &\quad x_i \in \mathbb{Z}, \quad i \in \mathcal{I}, \\ &\quad x \in [\ell, u], \end{aligned}$$

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.,

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

Example:

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



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$\equiv$

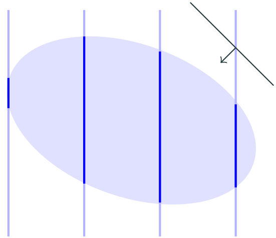
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$\equiv$

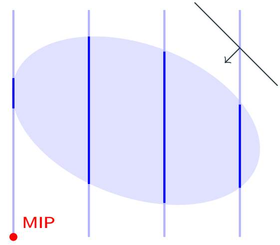
MIP

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

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

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

$$\equiv$$

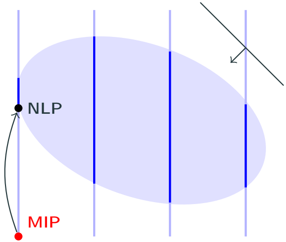
MIP

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

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

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

$\equiv$

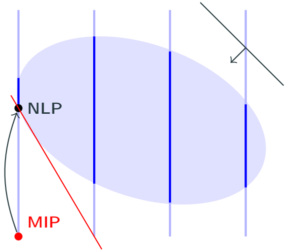
MIP

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

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

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

$\equiv$

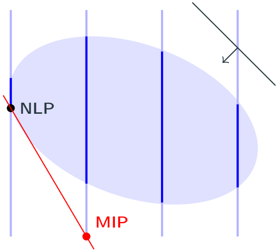
MIP

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

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

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

$\equiv$

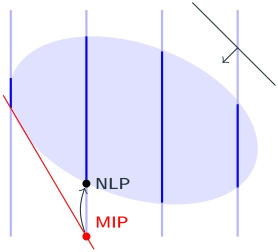
MIP

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

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

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

$\equiv$

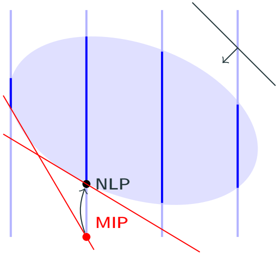
MIP

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

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

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

$$\equiv$$

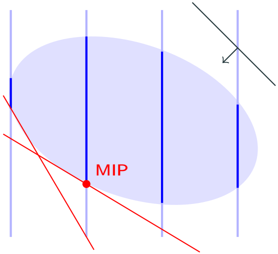
MIP

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

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

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

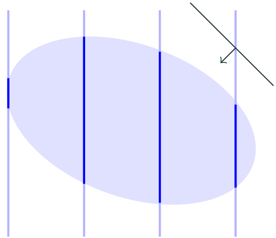
$$\equiv$$

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 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

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

$\equiv$

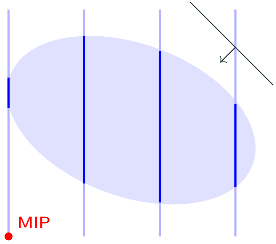
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$\equiv$

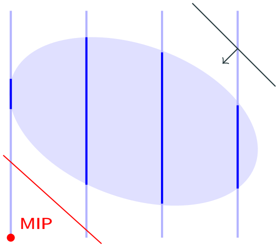
MIP

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

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

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

$\equiv$

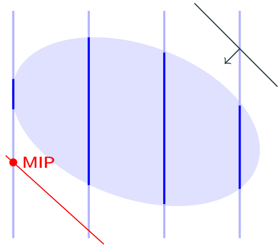
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$\equiv$

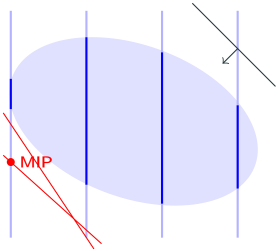
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$$\equiv$$

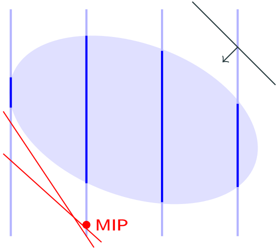
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$\equiv$

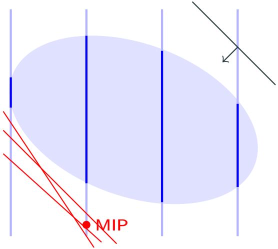
MIP

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

Extended 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

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

$$\equiv$$

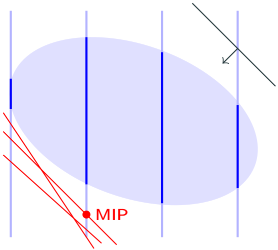
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$$\equiv$$

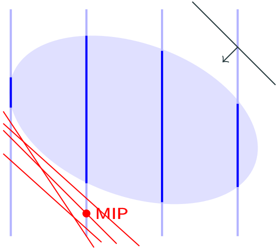
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$$\equiv$$

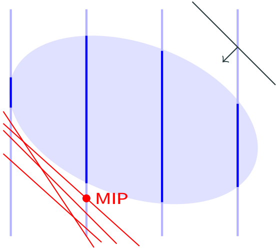
MIP

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

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.



Outer Approximation Method (OA), ECP, EHP

Convex MINLP

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

$\equiv$

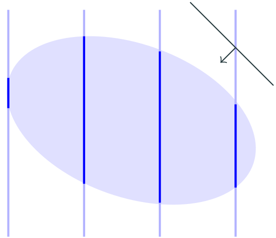
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

- 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

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

$$\equiv$$

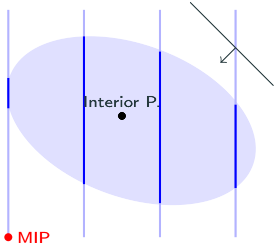
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

- 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

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

$$\equiv$$

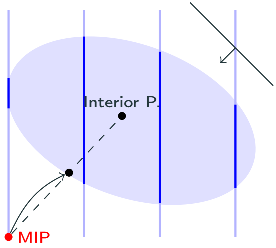
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

- 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

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

$\equiv$

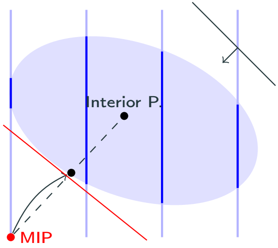
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

- 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

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

$$\equiv$$

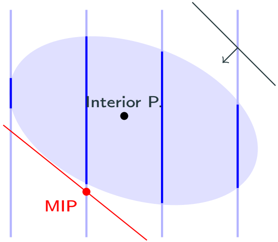
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

- 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

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

$$\equiv$$

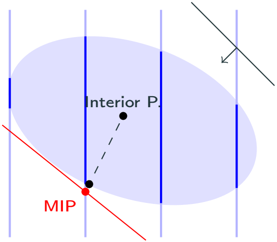
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

- 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

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

$$\equiv$$

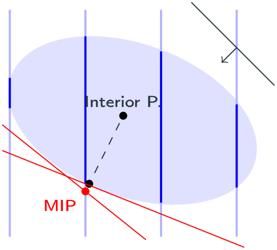
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

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



Convex MINLP

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

$\equiv$

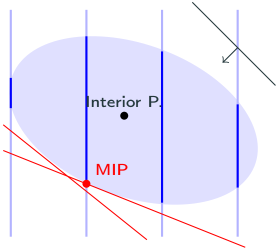
MIP

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

Extended Hyperplane Method (EHP)

[Veinott, 1967, Kronqvist, Lundell, and Westerlund, 2016, Lundell, Kronqvist, and Westerlund, 2022, Serrano, Schwarz, and Gleixner, 2020]:

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



LP/NLP- or LP-based Branch & Bound

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

LP/NLP- or LP-based Branch & Bound

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

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

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

LP/NLP- or LP-based Branch & Bound

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

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

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

LP-based Branch & Bound:

- Integrate Kelley' Cutting Plane method into MIP Branch & Bound.
- Add linearization in LP solution to LP relaxation (as in ECP).
- Optional: Move LP solution onto NLP-feasible set $\{x \in [\ell, u] : g_k(x) \leq 0\}$ via linesearch (as in EHP) [Lundell, Kronqvist, and Westerlund, 2022].

Fundamental Methods

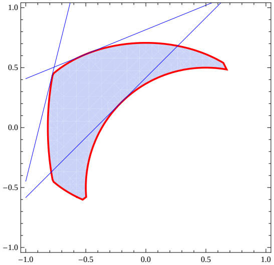
Nonconvex MINLP

Nonconvex MINLP

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

Outer-Approximation:

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



NLP-based Branch & Bound:

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

Nonconvex MINLP

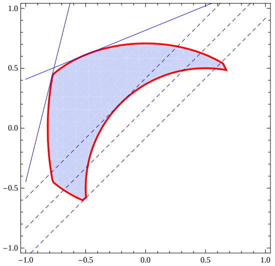
Now: Let $g_k(\cdot)$ be nonconvex for some $k \in [m]$.

Outer-Approximation:

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

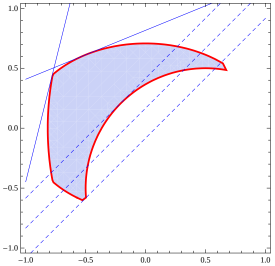
Now: Let $g_k(\cdot)$ be nonconvex for some $k \in [m]$.

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.

Convex Relaxation

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

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

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

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

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

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

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

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

- In practice, conv(X) is impossible to construct explicitly.

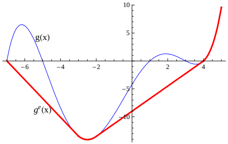
Relax I: Convexify the feasible sets that are defined by each constraint individually, i.e.,

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

- 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.,

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



Convex Relaxation

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

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

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

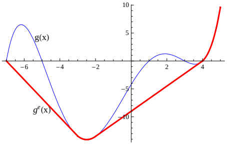
Relax I: Convexify the feasible sets that are defined by each constraint individually, i.e.,

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

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

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

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



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

Convex Relaxation

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

Seek: conv(X) – convex hull of X

- In practice, conv(X) is impossible to construct explicitly.

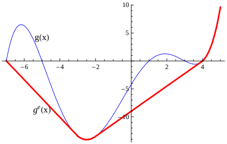
Relax I: Convexify the feasible sets that are defined by each constraint individually, i.e.,

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

- In practice, 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.,

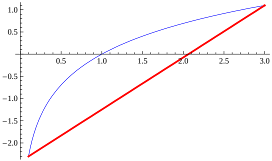
$$\{x \in [\ell, u] \,:\, \text{``conv}(g_k)\text{''}(x) \leq 0\}$$



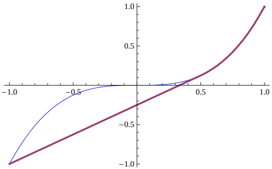
- In practice, convex envelope is not known explicitly in general – **except for many "simple functions"**

Convex Envelopes for “simple” functions

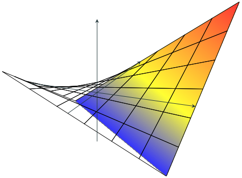
concave functions



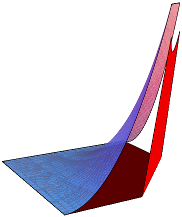
$x^k \quad (k \in 2\mathbb{Z}+1)$



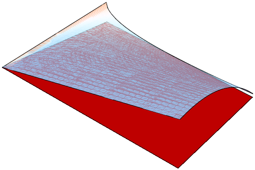
$x \cdot y$



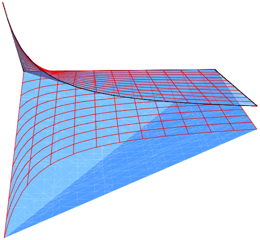
$x^2 \cdot y^2$



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



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



Application to Factorable Functions

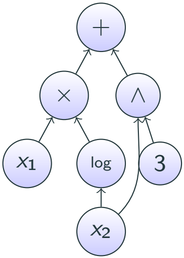
Factorable Functions [McCormick, 1976]

$g(x)$ 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:

$$x_1 \log(x_2) + x_2^3$$



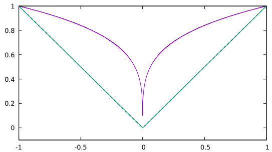
McCormick Underestimator for Factorable Functions

McCormick [1976] has shown a possibility to compose known envelopes.

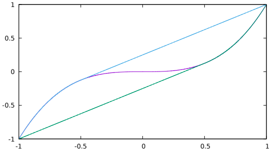
For example, consider $f(g(x))$ with $x \in [\ell_x, u_x]$, $f(\cdot)$ univariate.

1. Let $g(x) \in [\ell_g, u_g]$ for $x \in [\ell_x, u_x]$.
2. Let $\breve{f}(\cdot) \le f(\cdot)$ be convex envelope of $f(\cdot)$ on $[\ell_g, u_g]$.
3. Let $\breve{g}(\cdot) \le g(\cdot) \le \hat{g}(\cdot)$ be convex and concave envelopes of $g(\cdot)$ on $[\ell_x, u_x]$.
4. Let $z^{\min} \in \operatorname{argmin}_{z \in [\ell_g, u_g]} \breve{f}(z)$.

$f(z) = \sqrt{|z|}$, $\breve{f}(z)$, $z^{\min} = 0$:



$g(x) = x^3$, $x \in [-1, 1]$:



McCormick Underestimator for Factorable Functions

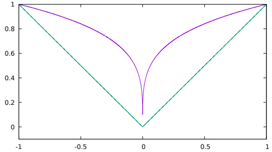
McCormick [1976] has shown a possibility to compose known envelopes.

For example, consider $f(g(x))$ with $x \in [\ell_x, u_x]$, $f(\cdot)$ univariate.

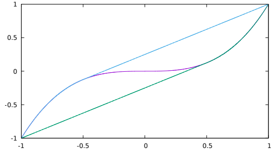
1. Let $g(x) \in [\ell_g, u_g]$ for $x \in [\ell_x, u_x]$.
2. Let $\breve{f}(\cdot) \leq f(\cdot)$ be convex envelope of $f(\cdot)$ on $[\ell_g, u_g]$.
3. Let $\breve{g}(\cdot) \leq g(\cdot) \leq \hat{g}(\cdot)$ be convex and concave envelopes of $g(\cdot)$ on $[\ell_x, u_x]$.
4. Let $z^{\min} \in \operatorname{argmin}_{z \in [\ell_g, u_g]} \breve{f}(z)$.
5. An obvious convex underestimator of $f(g(x))$ is given by

$$x \mapsto \breve{f}(z^{\min}).$$

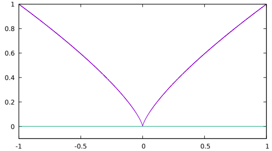
$f(z) = \sqrt{|z|}$, $\breve{f}(z)$, $z^{\min} = 0$:



$g(x) = x^3$, $x \in [-1, 1]$:



$f(g(x))$, McCormick$(f \circ g)$:



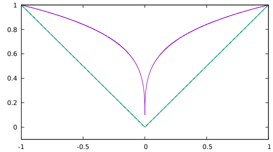
McCormick Underestimator for Factorable Functions

McCormick [1976] has shown a possibility to compose known envelopes.

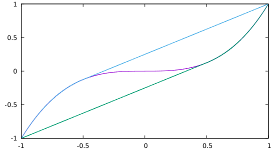
For example, consider $f(g(x))$ with $x \in [\ell_x, u_x]$, $f(\cdot)$ univariate.

1. Let $g(x) \in [\ell_g, u_g]$ for $x \in [\ell_x, u_x]$.
2. Let $\breve{f}(\cdot) \le f(\cdot)$ be convex envelope of $f(\cdot)$ on $[\ell_g, u_g]$.
3. Let $\breve{g}(\cdot) \le g(\cdot) \le \hat{g}(\cdot)$ be convex and concave envelopes of $g(\cdot)$ on $[\ell_x, u_x]$.
4. Let $z^{\min} \in \operatorname{argmin}_{z \in [\ell_g, u_g]} \breve{f}(z)$.
5. An obvious convex underestimator of $f(g(x))$ is given by
$$x \mapsto \breve{f}(z^{\min}).$$
6. The McCormick underestimator is
$$x \mapsto \breve{f}\left(\text{project } z^{\min} \text{ onto } [\breve{g}(x), \hat{g}(x)]\right)$$
(tighter for $z^{\min} \notin [\breve{g}(x), \hat{g}(x)]$).

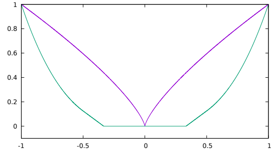
$f(z) = \sqrt{|z|}$, $\breve{f}(z)$, $z^{\min} = 0$:



$g(x) = x^3$, $x \in [-1, 1]$:



$f(g(x))$, McCormick$(f \circ g)$:

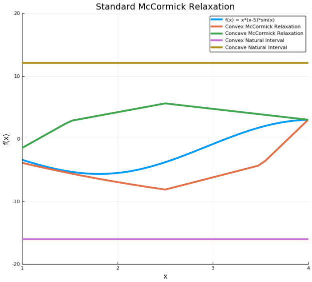


McCormick Underestimators

McCormick [1976]: A convex underestimator of $f(g(\cdot))$ on $[\ell_x, u_x]$ is

$$x \mapsto \begin{cases} \breve{f}(\breve{g}(x)), & \text{if } z^{\min} < \breve{g}(x), \\ \breve{f}(\hat{g}(x)), & \text{if } z^{\min} > \hat{g}(x), \\ \breve{f}(z^{\min}), & \text{else.} \end{cases} \qquad \text{where } z^{\min} = \operatorname*{argmin}_{z \in [\ell_g, u_g]} f(z).$$

- additional formulas for $f(x) \cdot g(x)$
- in general nonsmooth (nondifferentiable)
- implementations for evaluation and computation of subgradients exist, e.g., MC++ [Mitsos, Chachuat, and Barton, 2009]



Source: https://psorlab.github.io/EAGO.jl/stable/McCormick/Usage.html

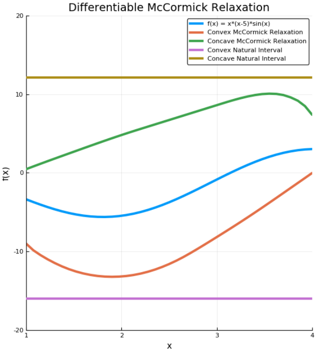
McCormick Underestimators

McCormick [1976]: A convex underestimator of $f(g(\cdot))$ on $[\ell_x, u_x]$ is

$$x \mapsto \begin{cases} \breve{f}(\breve{g}(x)), & \text{if } z^{\min} < \breve{g}(x), \\ \breve{f}(\hat{g}(x)), & \text{if } z^{\min} > \hat{g}(x), \\ \breve{f}(z^{\min}), & \text{else.} \end{cases} \qquad \text{where } z^{\min} = \operatorname*{argmin}_{z \in [\ell_g, u_g]} f(z).$$

- additional formulas for $f(x) \cdot g(x)$
- in general nonsmooth (nondifferentiable)
- implementations for evaluation and computation of subgradients exist, e.g., MC++ [Mitsos, Chachuat, and Barton, 2009]
- differentiable relaxation by Khan, Watson, and Barton [2017]

$\Rightarrow$ usable for convex NLP relaxations ($\rightarrow$ solvers EAGO and MAiNGO)



Source: https://psorlab.github.io/EAGO.jl/stable/McCormick/Usage.html

Reformulation of Factorable MINLP

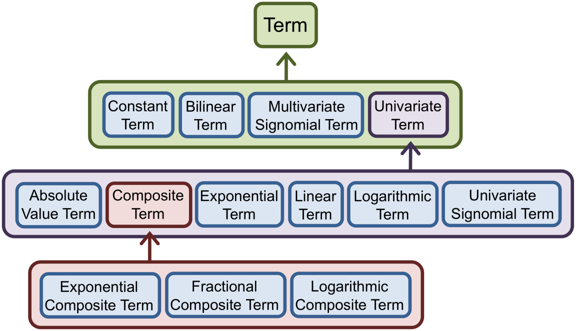
However, most global solvers reformulate factorable MINLPs by introducing new variables and equations [Smith and Pantelides, 1996, 1997]:

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

- 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, e.g., for ANTIGONE [Misener and Floudas, 2014]:



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

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

Spatial Branching

Recall **Spatial Branch & Bound**:

- ✓ **Relax nonconvexity** to obtain a tractable relaxation.
- **Branch on “nonconvexities”** to enforce original constraints.

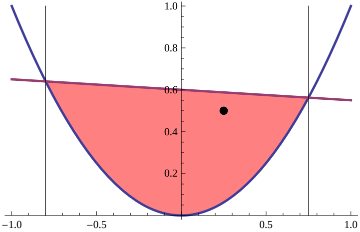
Spatial Branching

Recall **Spatial Branch & Bound**:

- ✓ **Relax nonconvexity** to obtain a tractable relaxation.
- **Branch on "nonconvexities"** to enforce original constraints.

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

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



Spatial Branching

Recall **Spatial Branch & Bound**:

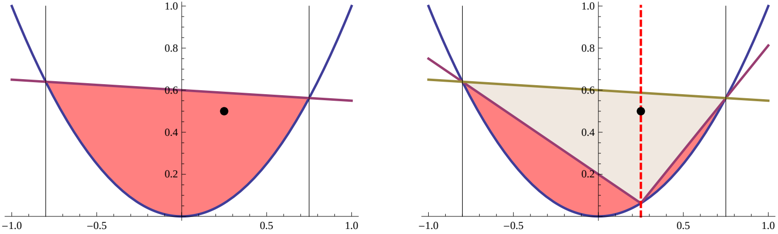
✓ **Relax nonconvexity** to obtain a tractable relaxation.

- **Branch on "nonconvexities"** to enforce original constraints.

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

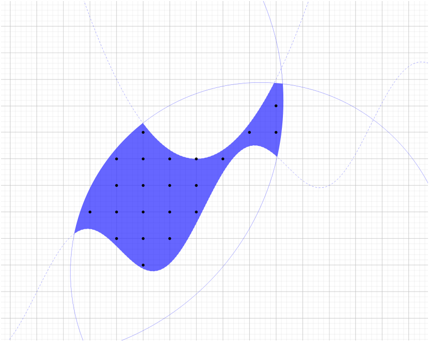
$$x^2 \leq \ell^2 + \frac{u^2 - \ell^2}{u - \ell}(x - \ell) \quad \forall x \in [\ell, u].$$

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



Spatial Branch and Bound

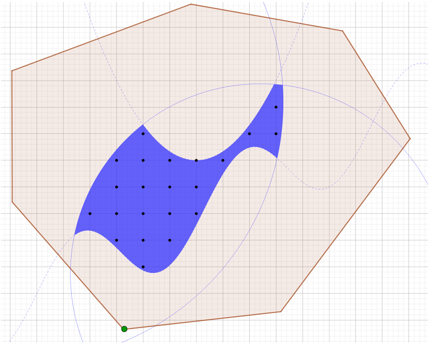
- Solve a relaxation → lower bound
- Run heuristics to look for feasible solutions → upper bound
- Branch on a suitable variable
- Discard parts of the tree that are infeasible or where lower bound > best known upper bound
- Repeat until gap is below given tolerance



Tighter variable bounds → improved relaxations → improved bounds on optimal value.

Spatial Branch and Bound

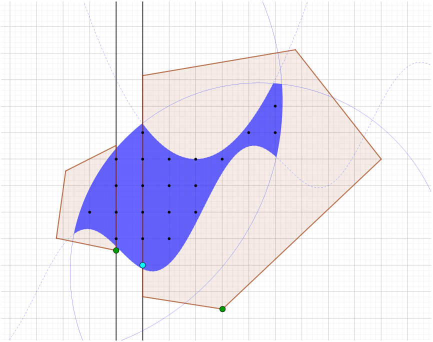
- Solve a relaxation → lower bound
- Run heuristics to look for feasible solutions → upper bound
- Branch on a suitable variable
- Discard parts of the tree that are infeasible or where lower bound > best known upper bound
- Repeat until gap is below given tolerance



Tighter variable bounds → improved relaxations → improved bounds on optimal value.

Spatial Branch and Bound

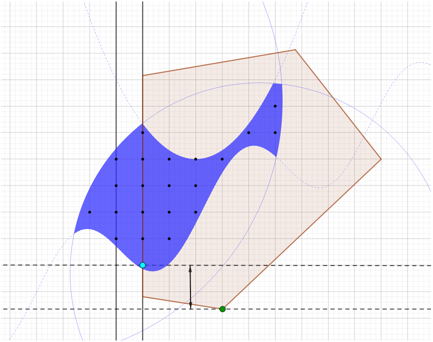
- Solve a relaxation → lower bound
- Run heuristics to look for feasible solutions → upper bound
- Branch on a suitable variable
- Discard parts of the tree that are infeasible or where lower bound > best known upper bound
- Repeat until gap is below given tolerance



Tighter variable bounds → improved relaxations → improved bounds on optimal value.

Spatial Branch and Bound

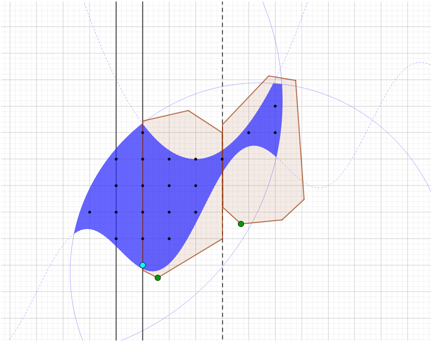
- Solve a relaxation → lower bound
- Run heuristics to look for feasible solutions → upper bound
- Branch on a suitable variable
- Discard parts of the tree that are infeasible or where lower bound > best known upper bound
- Repeat until gap is below given tolerance



Tighter variable bounds → improved relaxations → improved bounds on optimal value.

Spatial Branch and Bound

- Solve a relaxation → lower bound
- Run heuristics to look for feasible solutions → upper bound
- Branch on a suitable variable
- Discard parts of the tree that are infeasible or where lower bound > best known upper bound
- Repeat until gap is below given tolerance



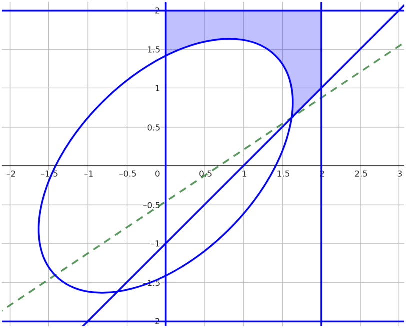
Tighter variable bounds → improved relaxations → improved bounds on optimal value.

Example

Example

Consider

$$\begin{aligned} \text{minimize} \quad & -2x + 3y \\ \text{such that } & x^2 - xy + y^2 \geq 2 \\ & x - y \leq 1 \\ & x \in [0, 2], \\ & y \in [-2, 2] \end{aligned}$$



Optimal solution:

- from the picture, both inequalities are active $\Rightarrow y = x - 1$

$\Rightarrow 2 = x^2 - x(x-1) + (x-1)^2 = x^2 - x + 1 \Rightarrow (x - \frac{1}{2})^2 = \frac{5}{4}$

- $x \geq 0 \Rightarrow x = \frac{1+\sqrt{5}}{2}$, $y = \frac{\sqrt{5}-1}{2}$, objective $= \frac{\sqrt{5}-5}{2} \approx -1.38$

Example: Solvers

Solve with GAMS (AMPL works too):

```
Variables x, y, z;
Equations e1, e2, e3;

e1.. -2*x + 3*y =E= z;
e2.. sqr(x)+sqr(y)-x*y =G= 2;
e3.. x - y =L= 1;

x.lo =  0;   x.up =  2;
y.lo = -1;   y.up =  2;

Model m /all/;
Solve m min z using qcp;
```

solver	optimum	time	B&B tree
ANTIGONE	-1.381966	0.00s	1 node
BARON	-1.381966	0.03s	1 node
CONOPT	infeasible	0.00s	–
Gurobi	-1.381966	0.02s	13 nodes
Ipopt	-1.381966	0.00s	–
Knitro	-1.381966	0.01s	–
Lindo API	-1.381968	0.22s	3 nodes
Minos	infeasible	0.01s	–
SCIP	-1.381966	0.05s	1 node
SNOPT	infeasible	0.00s	–
Octeract	-1.381966	0.01s	4 nodes

Initial LP Relaxation: X enters the stage

Constraint:

$$x^2 - xy + y^2 \geq 2, \qquad x \in [0, 2], \qquad y \in [-2, 2]$$

Introduce $X_{xx} = x^2$, $X_{xy} = xy$, $X_{yy} = y^2$.

Initial LP Relaxation: X enters the stage

Constraint:

$$x^2 - xy + y^2 \geq 2, \qquad x \in [0, 2], \qquad y \in [-2, 2]$$

Introduce $X_{xx} = x^2$, $X_{xy} = xy$, $X_{yy} = y^2$.

Since x^2 and y^2 are convex, we can use a tangent and secant on its graph, e.g.,

$$\underbrace{4 + 4(x-2)}_{\text{tangent at } x=2} \leq x^2 \leq \underbrace{0 + \frac{4-0}{2-0}(x-0)}_{\text{secant from } x=0 \text{ to } x=2} \quad \Rightarrow \quad 4x - 4 \leq X_{xx} \leq 2x$$

Initial LP Relaxation: X enters the stage

Constraint:

$$x^2 - xy + y^2 \geq 2, \qquad x \in [0,2], \qquad y \in [-2,2]$$

Introduce $X_{xx} = x^2$, $X_{xy} = xy$, $X_{yy} = y^2$.

Since x^2 and y^2 are convex, we can use a tangent and secant on its graph, e.g.,

$$\underbrace{4 + 4(x-2)}_{\text{tangent at } x=2} \leq x^2 \leq \underbrace{0 + \frac{4-0}{2-0}(x-0)}_{\text{secant from } x=0 \text{ to } x=2} \quad \Rightarrow \quad 4x - 4 \leq X_{xx} \leq 2x$$

Or derive inequalities by multiplying variable bound constraints:

$$0 \leq (x-0)^2 \qquad = x^2 \qquad = X_{xx} \qquad \rightarrow X_{xx} \geq 0$$

Initial LP Relaxation: X enters the stage

Constraint:

$$x^2 - xy + y^2 \geq 2, \qquad x \in [0,2], \qquad y \in [-2,2]$$

Introduce $X_{xx} = x^2$, $X_{xy} = xy$, $X_{yy} = y^2$.

Since x^2 and y^2 are convex, we can use a tangent and secant on its graph, e.g.,

$$\underbrace{4 + 4(x-2)}_{\text{tangent at } x=2} \leq x^2 \leq \underbrace{0 + \frac{4-0}{2-0}(x-0)}_{\text{secant from } x=0 \text{ to } x=2} \quad \Rightarrow \quad 4x - 4 \leq X_{xx} \leq 2x$$

Or derive inequalities by multiplying variable bound constraints:

$$\begin{array}{llll} 0 \leq (x-0)^2 & = x^2 & = X_{xx} & \to X_{xx} \geq 0 \\ 0 \leq (2-x)^2 & = x^2 - 4x + 4 & = X_{xx} - 4x + 4 & \to X_{xx} \geq 4x - 4 \end{array}$$

Initial LP Relaxation: X enters the stage

Constraint:

$$x^2 - xy + y^2 \geq 2, \qquad x \in [0,2], \qquad y \in [-2,2]$$

Introduce $X_{xx} = x^2$, $X_{xy} = xy$, $X_{yy} = y^2$.

Since x^2 and y^2 are convex, we can use a tangent and secant on its graph, e.g.,

$$\underbrace{4 + 4(x-2)}_{\text{tangent at } x=2} \leq x^2 \leq \underbrace{0 + \frac{4-0}{2-0}(x-0)}_{\text{secant from } x=0 \text{ to } x=2} \quad \Rightarrow \quad 4x - 4 \leq X_{xx} \leq 2x$$

Or derive inequalities by multiplying variable bound constraints:

$$\begin{array}{llll}
0 \leq (x-0)^2 & = x^2 & = X_{xx} & \to X_{xx} \geq 0 \\
0 \leq (2-x)^2 & = x^2 - 4x + 4 & = X_{xx} - 4x + 4 & \to X_{xx} \geq 4x - 4 \\
0 \leq (2-x)(x-0) & = -x^2 + 2x & = -X_{xx} + 2x & \to X_{xx} \leq 2x
\end{array}$$

Initial LP Relaxation: X enters the stage

Constraint:

$$x^2 - xy + y^2 \geq 2, \qquad x \in [0,2], \qquad y \in [-2,2]$$

Introduce $X_{xx} = x^2$, $X_{xy} = xy$, $X_{yy} = y^2$.

Since x^2 and y^2 are convex, we can use a tangent and secant on its graph, e.g.,

$$\underbrace{4 + 4(x-2)}_{\text{tangent at } x=2} \leq x^2 \leq \underbrace{0 + \frac{4-0}{2-0}(x-0)}_{\text{secant from } x=0 \text{ to } x=2} \quad \Rightarrow \quad 4x - 4 \leq X_{xx} \leq 2x$$

Or derive inequalities by multiplying variable bound constraints:

$$\begin{aligned}
0 &\leq (x-0)^2 &&= x^2 &&= X_{xx} &&\to X_{xx} \geq 0 \\
0 &\leq (2-x)^2 &&= x^2 - 4x + 4 &&= X_{xx} - 4x + 4 &&\to X_{xx} \geq 4x - 4 \\
0 &\leq (2-x)(x-0) &&= -x^2 + 2x &&= -X_{xx} + 2x &&\to X_{xx} \leq 2x \\
0 &\leq (y-(-2))^2 &&= y^2 + 4y + 4 &&= X_{yy} + 4y + 4 &&\to X_{yy} \geq -4y - 4 \\
0 &\leq (y-(-2))(2-y) &&= -y^2 + 4 &&= -X_{yy} + 4 &&\to X_{yy} \leq 4 \\
0 &\leq (2-y)^2 &&= y^2 - 4y + 4 &&= X_{yy} - 4y + 4 &&\to X_{yy} \geq 4y - 4 \\
0 &\leq (x-0)(y-(-2)) &&= xy + 2x &&= X_{xy} + 2x &&\to X_{xy} \geq -2x \\
0 &\leq (x-0)(2-y) &&= -xy + 2x &&= -X_{xy} + 2x &&\to X_{xy} \leq 2x \\
0 &\leq (2-x)(y-(-2)) &&= -xy - 2x + 2y + 4 &&= -X_{xy} - 2x + 2y + 4 &&\to X_{xy} \leq -2x + 2y + 4 \\
0 &\leq (2-x)(2-y) &&= xy - 2x - 2y + 4 &&= X_{xy} - 2x - 2y + 4 &&\to X_{xy} \geq 2x + 2y - 4
\end{aligned}$$

Initial LP Relaxation

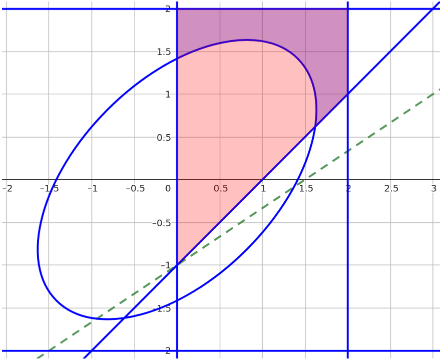
Replace $(x^2,\ xy,\ y^2)$ by (X_{xx}, X_{xy}, X_{yy})
and add derived inequalities:

$$\min\ -2x + 3y$$

s.t. ~~$x^2 - xy + y^2 \geq 2$~~

$$X_{xx} - X_{xy} + X_{yy} \geq 2$$
$$x - y \leq 1$$
$$X_{xx} \geq 4x - 4$$
$$X_{xx} \leq 2x$$
$$X_{yy} \geq -4y - 4$$
$$X_{yy} \geq 4y - 4$$
$$X_{xy} \leq 2x$$
$$X_{xy} \leq -2x + 2y + 4$$
$$X_{xy} \geq 2x + 2y + 4$$
$$x \in [0, 2], y \in [-2, 2]$$
$$X_{xx} \in [0, \infty], X_{yy} \in [-\infty, 4]$$

Projected on (x, y):

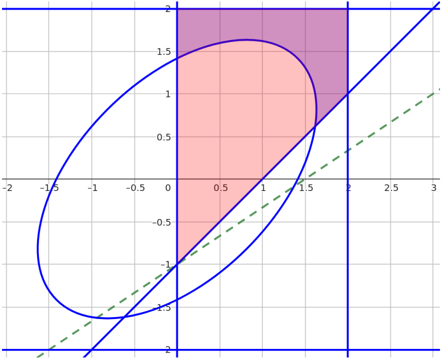


- Lower Bound = -3

$\Rightarrow$ none of the inequalities in (X_{xx}, X_{xy}, X_{yy}) are active :-(

Tighten variable bounds

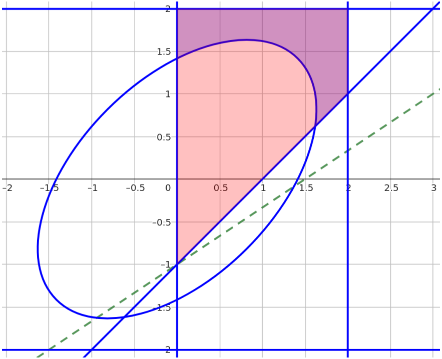
- inequalities for relaxation were derived using bounds on x and y
- tighter bounds could mean a tighter relaxation



Tighten variable bounds

- inequalities for relaxation were derived using bounds on x and y
- tighter bounds could mean a tighter relaxation

$$x - y \leq 1, x \in [0, 2] \quad \Rightarrow y \geq x - 1 \geq -1$$
$$x - y \leq 1, y \in [-2, 2] \quad \Rightarrow x \leq y + 1 \leq 3$$



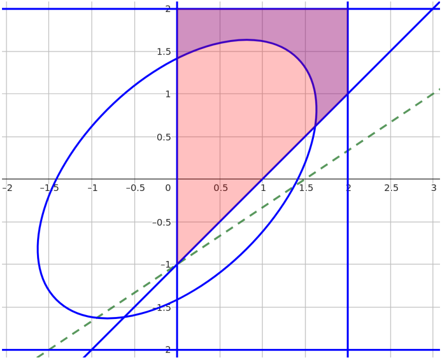
- updated bounds:

$$x \in [0, 2], \quad y \in [-1, 2]$$

Tighten variable bounds

- inequalities for relaxation were derived using bounds on x and y
- tighter bounds could mean a tighter relaxation

$$x - y \leq 1, x \in [0, 2] \quad \Rightarrow y \geq x - 1 \geq -1$$
$$x - y \leq 1, y \in [-2, 2] \quad \Rightarrow x \leq y + 1 \leq 3$$

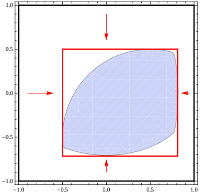


- updated bounds:

$$x \in [0, 2], \quad y \in [-1, 2]$$

- from $x^2 - xy + y^2 \geq 2$, no bound tightening can be derived

In General: 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

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"

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:

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

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

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

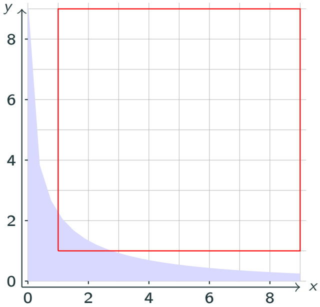
- Belotti, Cafieri, Lee, and Liberti [2010]: fixed point of iterating FBBT on set of linear constraints can be computed by solving one LP

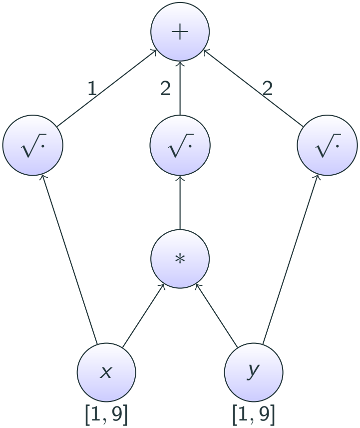
Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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



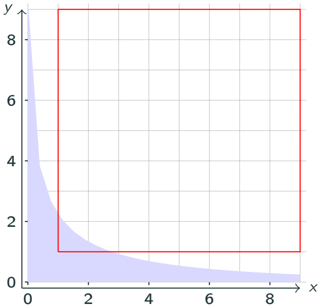


Feasibility-Based Bound Tightening on Expression Tree

Example:

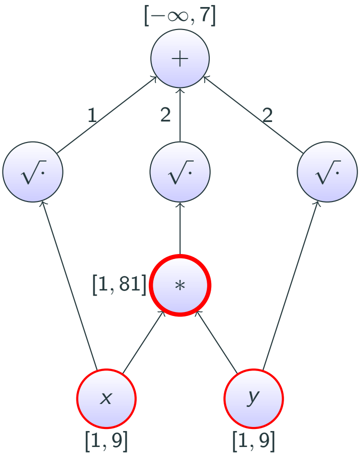
$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$

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



Forward propagation:

- compute bounds on intermediate nodes (bottom-up)



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

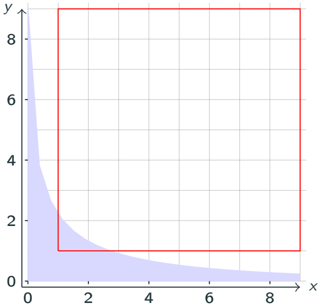
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

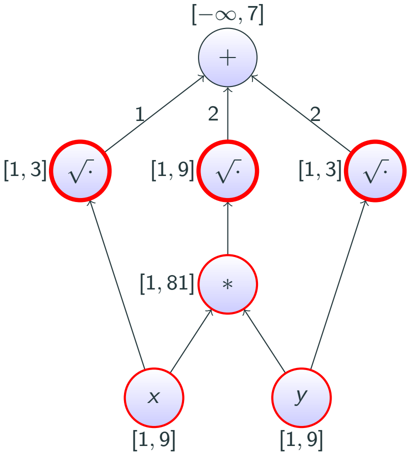
$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$

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



Forward propagation:

- compute bounds on intermediate nodes (bottom-up)



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

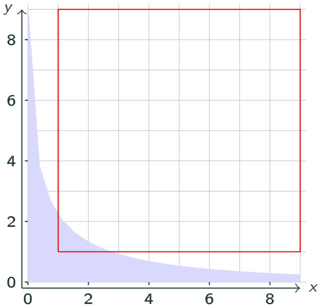
Application of **Interval Arithmetics** [Moore, 1966]

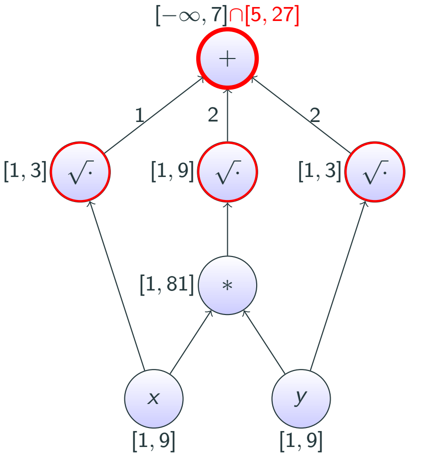
Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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





$$[1, 3] + 2\,[1, 9] + 2\,[1, 3] = [5, 27]$$

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

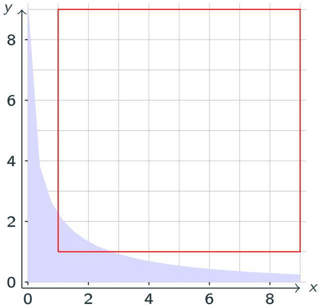
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

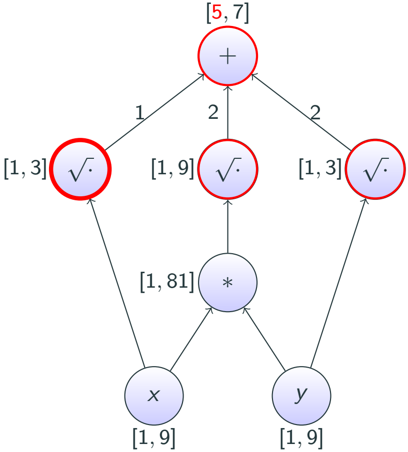


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

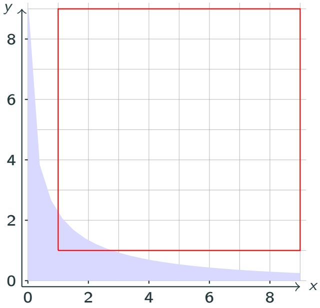
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

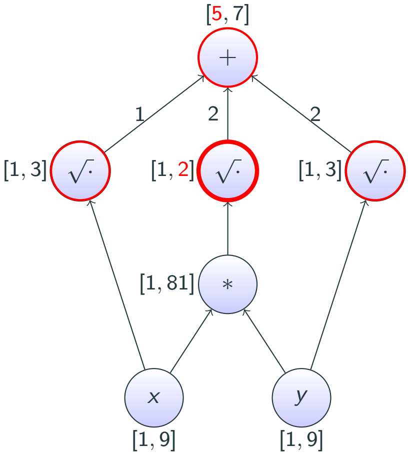


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

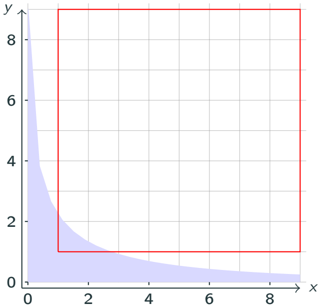
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

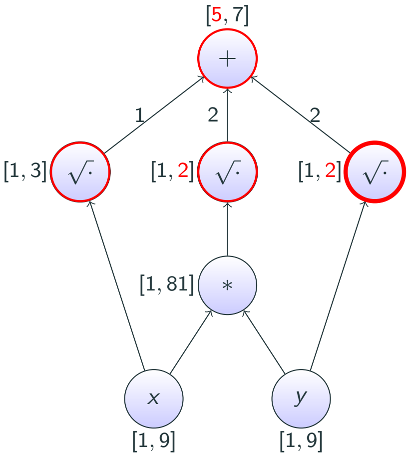


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

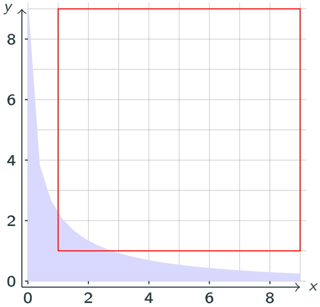
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

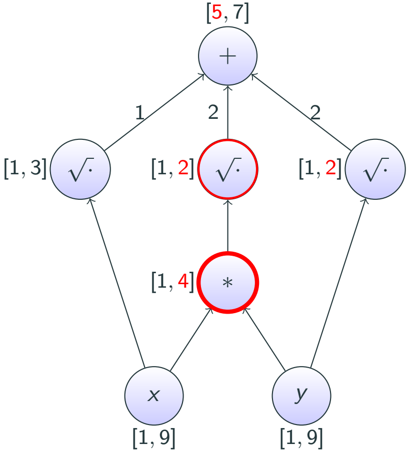


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

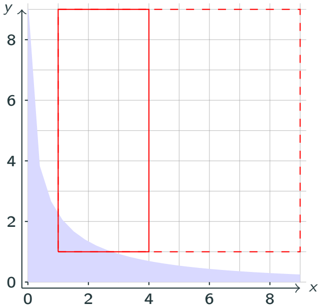
Application of **Interval Arithmetics** [Moore, 1966]

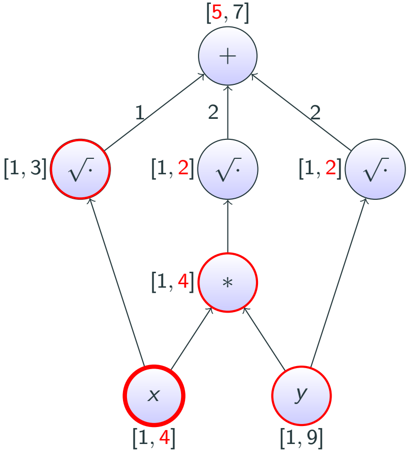
Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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





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

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

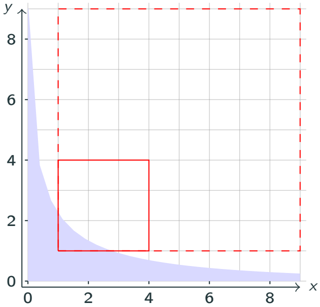
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

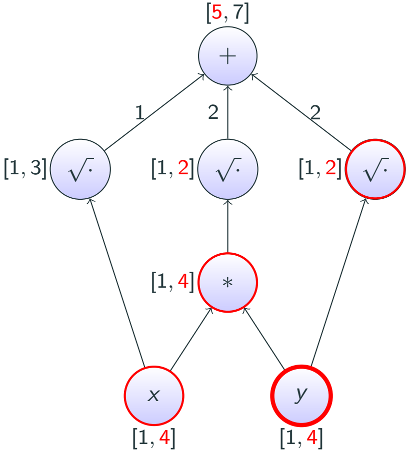


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

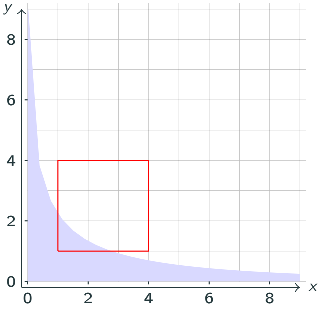
Application of **Interval Arithmetics** [Moore, 1966]

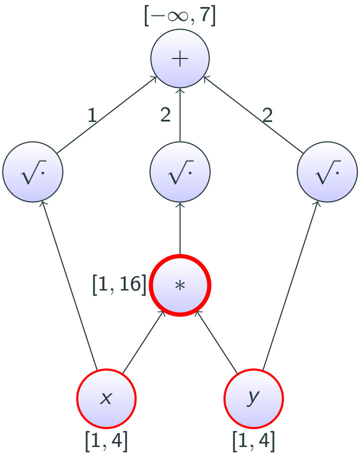
Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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





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

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

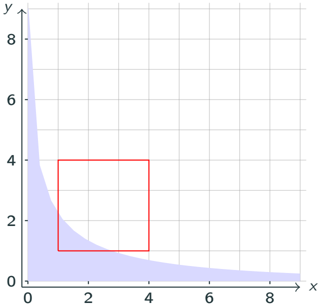
- reduce bounds using reverse operations (top-down)

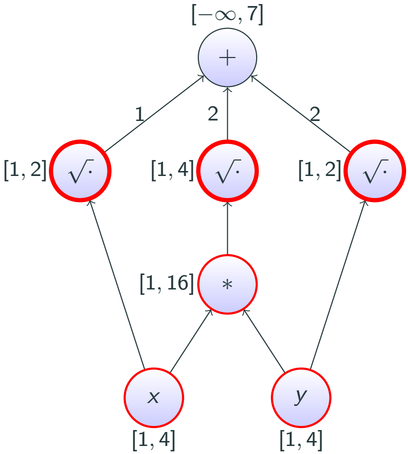
Application of **Interval Arithmetics** [Moore, 1966]

Example:

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

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





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

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

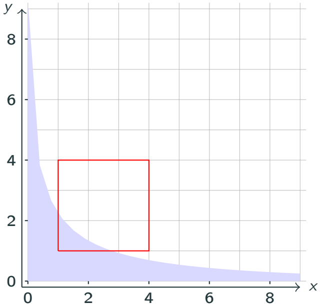
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

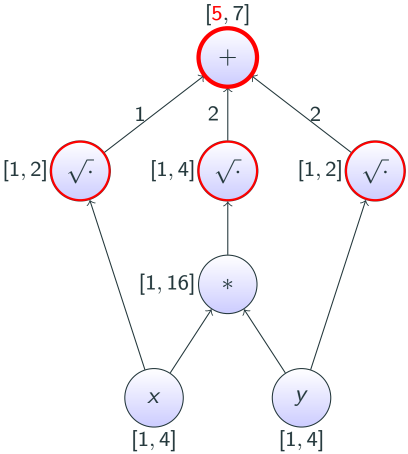


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

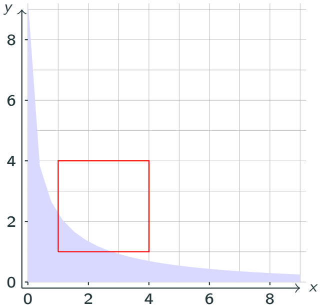
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

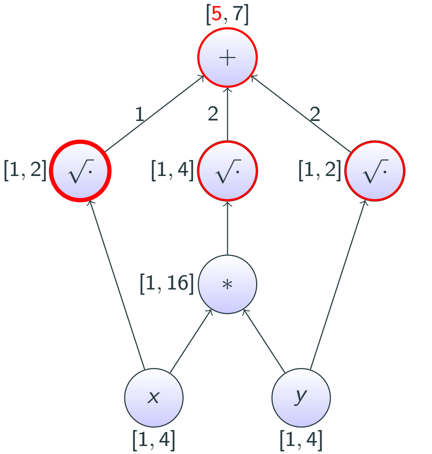


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



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

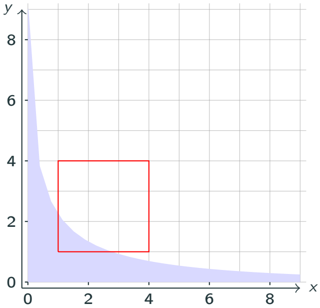
Application of **Interval Arithmetics** [Moore, 1966]

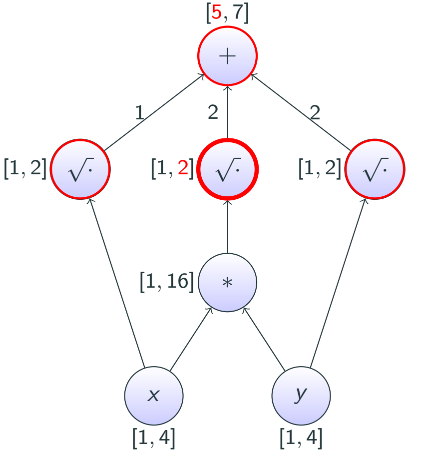
Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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





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

Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

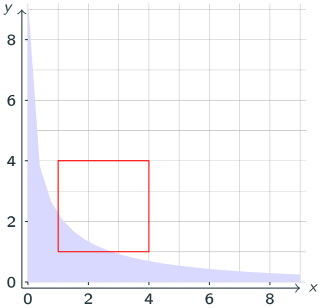
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

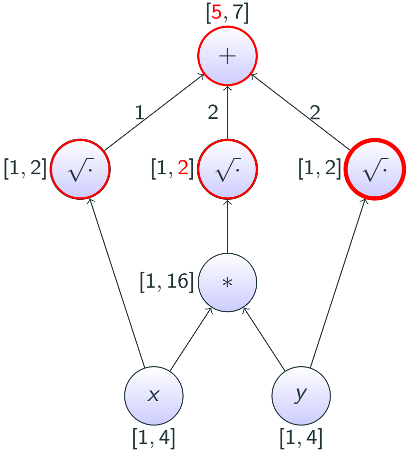


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



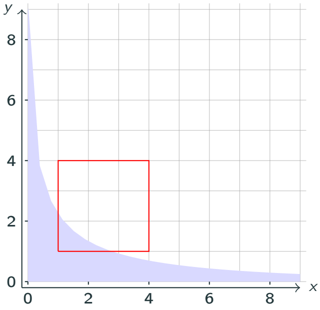
$$([5, 7] - [1, 2] - 2\,[1, 4])/2 = [-2.5, 2]$$

Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

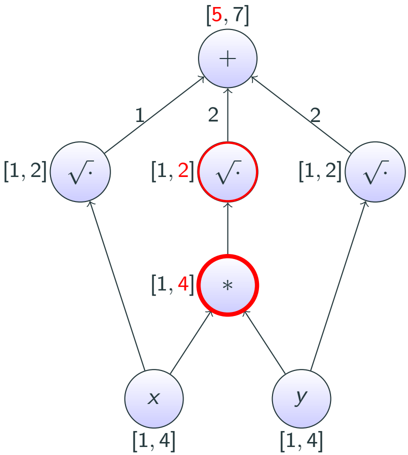


Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)



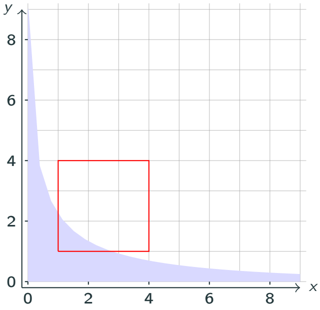
$[1, 2]^2 = [1, 4]$

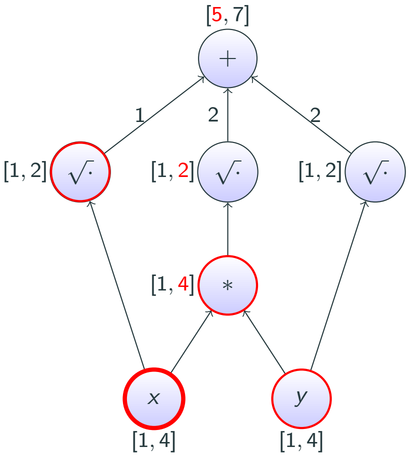
Application of **Interval Arithmetics** [Moore, 1966]

Example:

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

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





Forward propagation:

- compute bounds on intermediate nodes (bottom-up)

Backward propagation:

- reduce bounds using reverse operations (top-down)

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

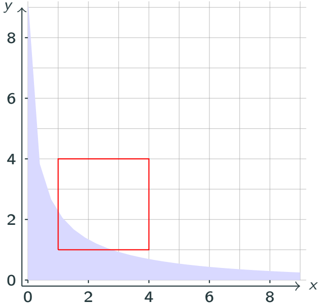
Application of **Interval Arithmetics** [Moore, 1966]

Feasibility-Based Bound Tightening on Expression Tree

Example:

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

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

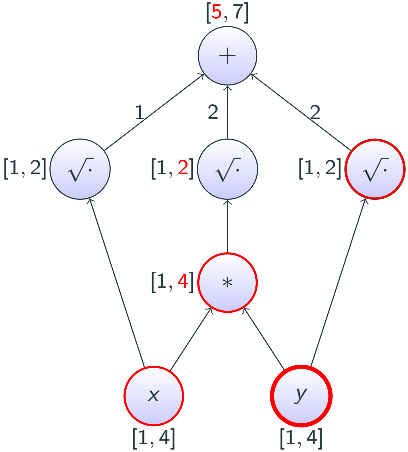


Forward propagation:

- 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

Back to Example: Relaxation after bound update

Problem: $\min\{-2x + 3y \;:\; x^2 - xy + y^2 \geq 2,\, x - y \leq 1,\, x \in [0,2], y \in [-1,2]\}$

Linearization: $x^2 \to X_{xx},\, xy \to X_{xy},\, y^2 \to X_{yy}$

Recompute initial relaxation with lower bound on y updated to -1:

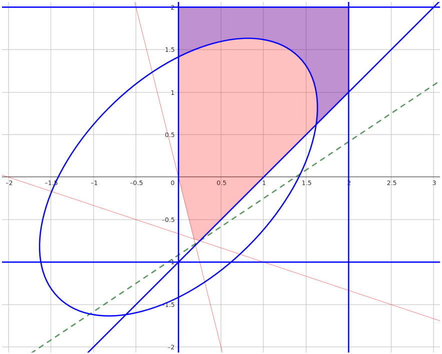
$$
\begin{array}{llll}
0 \leq (x-0)^2 & = x^2 & = X_{xx} & \to X_{xx} \geq 0 \\
0 \leq (2-x)^2 & = x^2 - 4x + 4 & = X_{xx} - 4x + 4 & \to X_{xx} \geq 4x - 4 \\
0 \leq (2-x)(x-0) & = -x^2 + 2x & = -X_{xx} + 2x & \to X_{xx} \leq 2x \\
0 \leq (y-(-1))^2 & = y^2 + y + 1 & = X_{yy} + y + 1 & \to X_{yy} \geq -y - 1 \\
0 \leq (y-(-1))(2-y) & = -y^2 + y + 2 & = -X_{yy} + y + 2 & \to X_{yy} \leq y + 2 \\
0 \leq (2-y)^2 & = y^2 - 4y + 4 & = X_{yy} - 4y + 4 & \to X_{yy} \geq 4y - 4 \\
0 \leq (x-0)(y-(-1)) & = xy + x & = X_{xy} + x & \to X_{xy} \geq -x \\
0 \leq (x-0)(2-y) & = -xy + 2x & = -X_{xy} + 2x & \to X_{xy} \leq 2x \\
0 \leq (2-x)(y-(-1)) & = -xy - x + 2y + 2 & = -X_{xy} - x + 2y + 2 & \to X_{xy} \leq -x + 2y + 2 \\
0 \leq (2-x)(2-y) & = xy - 2x - 2y + 4 & = X_{xy} - 2x - 2y + 4 & \to X_{xy} \geq 2x + 2y - 4
\end{array}
$$

LP Relaxation after Bound Tightening

With $y \geq -1$:

$$
\begin{aligned}
\min \quad & -2x + 3y \\
\text{s.t.} \quad & X_{xx} - X_{xy} + X_{yy} \geq 2 \\
& x - y \leq 1 \\
& X_{xx} \geq 0 \\
& X_{xx} \geq 4x - 4 \\
& X_{xx} \leq 2x \\
& X_{yy} \geq -y - 1 \\
& X_{yy} \leq y + 2 \\
& X_{yy} \geq 4y - 4 \\
& X_{xy} \geq -x \\
& X_{xy} \leq 2x \\
& X_{xy} \leq -x + 2y + 2 \\
& X_{xy} \geq 2x + 2y + 4 \\
& x \in [0, 2], y \in [-1, 2]
\end{aligned}
$$

Projected on (x, y):



- Lower Bound = -2.75 (improvement from -3)

Can we get more cuts?

- we should make use of the inequality $x - y \leq 1$
- Idea: multiply bounds with linear inequality

Can we get more cuts?

- we should make use of the inequality $x - y \leq 1$
- Idea: multiply bounds with linear inequality

$$0 \leq (1 - x + y)(x - 0) \qquad = x - x^2 + xy \qquad\qquad = x - X_{xx} + X_{xy}$$

Can we get more cuts?

- we should make use of the inequality $x - y \leq 1$
- Idea: multiply bounds with linear inequality

$$
\begin{aligned}
0 &\leq (1 - x + y)(x - 0) &&= x - x^2 + xy &&= x - X_{xx} + X_{xy} \\
0 &\leq (1 - x + y)(2 - x) &&= 2 - x - 2x + x^2 + 2y - xy &&= 2 - 3x + X_{xx} + 2y - X_{xy}
\end{aligned}
$$

Can we get more cuts?

- we should make use of the inequality $x - y \leq 1$
- Idea: multiply bounds with linear inequality

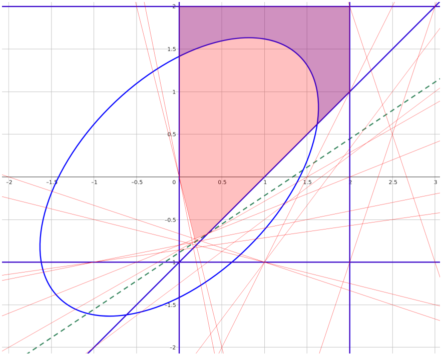
$$
\begin{aligned}
0 &\leq (1 - x + y)(x - 0) &&= x - x^2 + xy &&= x - X_{xx} + X_{xy} \\
0 &\leq (1 - x + y)(2 - x) &&= 2 - x - 2x + x^2 + 2y - xy &&= 2 - 3x + X_{xx} + 2y - X_{xy} \\
0 &\leq (1 - x + y)(y - (-1)) &&= y + 1 - xy - x + y^2 + y &&= 2y + 1 - X_{xy} - x + X_{yy} \\
0 &\leq (1 - x + y)(2 - y) &&= 2 - y - 2x + xy + 2y - y^2 &&= 2 + y - 2x + X_{xy} - X_{yy}
\end{aligned}
$$

Inequalities that couple several X $\rightarrow$ looks promising

LP Relaxation with additional cuts

$$
\begin{aligned}
\min\ & -2x + 3y \\
\text{s.t. } & X_{xx} - X_{xy} + X_{yy} \geq 2 \\
& x - y \leq 1 \\
& X_{xx} \geq 0 \\
& X_{xx} \geq 4x - 4 \\
& X_{xx} \leq 2x \\
& X_{yy} \geq -y - 1 \\
& X_{yy} \leq y + 2 \\
& X_{yy} \geq 4y - 4 \\
& X_{xy} \geq -x \\
& X_{xy} \leq 2x \\
& X_{xy} \leq -x + 2y + 2 \\
& X_{xy} \geq 2x + 2y + 4 \\
& X_{xx} - X_{xy} \leq x \\
& X_{xx} - X_{xy} \geq 3x - 2y - 2 \\
& X_{xy} - X_{yy} \leq 2y - x + 1 \\
& X_{xy} - X_{yy} \geq 2x - y - 2 \\
& x \in [0, 2], y \in [-1, 2]
\end{aligned}
$$

Projected on (x, y):

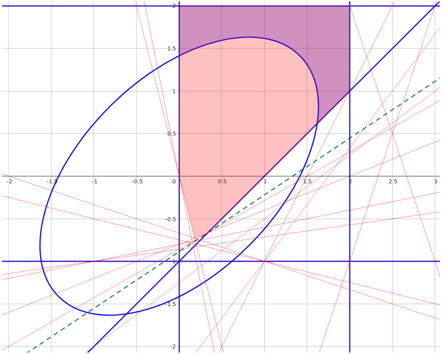


- Lower Bound = -2.66 (improvement from -2.75)

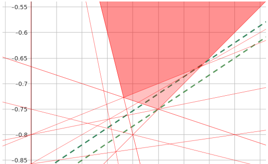
LP Relaxation with additional cuts

$$
\begin{aligned}
\min\ & -2x + 3y \\
\text{s.t. } & X_{xx} - X_{xy} + X_{yy} \geq 2 \\
& x - y \leq 1 \\
& X_{xx} \geq 0 \\
& X_{xx} \geq 4x - 4 \\
& X_{xx} \leq 2x \\
& X_{yy} \geq -y - 1 \\
& X_{yy} \leq y + 2 \\
& X_{yy} \geq 4y - 4 \\
& X_{xy} \geq -x \\
& X_{xy} \leq 2x \\
& X_{xy} \leq -x + 2y + 2 \\
& X_{xy} \geq 2x + 2y + 4 \\
& X_{xx} - X_{xy} \leq x \\
& X_{xx} - X_{xy} \geq 3x - 2y - 2 \\
& X_{xy} - X_{yy} \leq 2y - x + 1 \\
& X_{xy} - X_{yy} \geq 2x - y - 2 \\
& x \in [0, 2],\ y \in [-1, 2]
\end{aligned}
$$

Projected on (x, y):



- Lower Bound = -2.66 (improvement from -2.75)



In General: Reformulation Linearization Technique (RLT)

Consider the QCQP

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

In General: Reformulation Linearization Technique (RLT)

Consider the QCQP

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

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

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

In General: Reformulation Linearization Technique (RLT)

Consider the QCQP

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

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

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

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

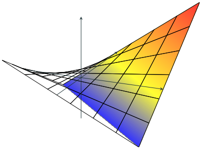
Multiplying bounds $\ell_i \leq x_i \leq u_i$ and $\ell_j \leq x_j \leq u_j$ and using $X_{i,j} = x_i x_j$ yields

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

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

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

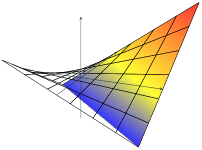


- these are more widely known as McCormick inequalities [McCormick, 1976]

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

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



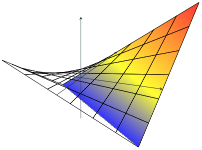
- these are more widely known as McCormick inequalities [McCormick, 1976]
- the resulting linear relaxation is

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

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

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



- these are more widely known as McCormick inequalities [McCormick, 1976]
- the resulting linear relaxation is

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

- 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:

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

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:

$$
\begin{aligned}
(A_k^{\mathsf{T}} x - b_k)(x_j - \ell_j) \geq 0 \quad &\Rightarrow \quad \sum_{i=1}^{n} A_{k,i}(X_{i,j} - x_i \ell_j) - b_k(x_j - \ell_j) \geq 0 \\
(A_k^{\mathsf{T}} x - b_k)(A_{k'}^{\mathsf{T}} x - b_{k'}) \geq 0 \quad &\Rightarrow \quad A_k^{\mathsf{T}} X A_{k'}^{\mathsf{T}} - (b_k A_{k'} + b_{k'} A_k^{\mathsf{T}})x + b_k b_{k'} \geq 0
\end{aligned}
$$

RLT: Multiplying Bounds and Inequalities

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

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

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

RLT is also used for polynomial programs [Sherali and Tuncbilek, 1992]:

- any monomial $\prod_i x_i^{\alpha_i}$ is replaced by a new variable
- more than two bounds or (in)equalities are multiplied
- solver: RAPOSa [González-Rodríguez et al., 2022]

Back to Example: Objective Cutoff

$\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\ x - y \leq 1,\ x \in [0,2], y \in [-1,2]\}$

Assume the optimal solution with objective $= \frac{\sqrt{5}-5}{2}$ has been found, e.g., by a NLP solver, but proof of optimality is still missing.

Objective cutoff: Look only for improving solutions: $-2x + 3y \leq \frac{\sqrt{5}-5}{2}$

Back to Example: Objective Cutoff

$\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\ x - y \leq 1,\ x \in [0,2], y \in [-1,2]\}$

Assume the optimal solution with objective $= \frac{\sqrt{5}-5}{2}$ has been found, e.g., by a NLP solver, but proof of optimality is still missing.

Objective cutoff: Look only for improving solutions: $-2x + 3y \leq \frac{\sqrt{5}-5}{2}$

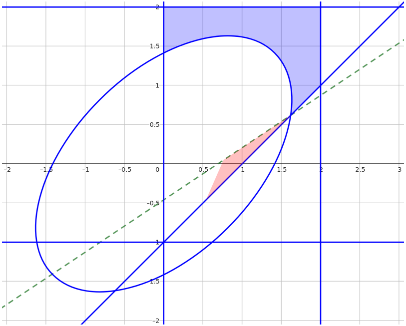
RLT with this inequality:

$$0 \leq 2X_{xx} - 3X_{xy} + \frac{\sqrt{5}}{2}x - \frac{5}{2}x$$

$$0 \leq -2X_{xx} + 3X_{xy} - \frac{\sqrt{5}}{2}x + \frac{13}{2}x - 6y + \sqrt{5} - 5$$

$$0 \leq 2X_{xy} - 3X_{yy} + \frac{\sqrt{5}}{2}y + 2x - \frac{11}{2}y + \frac{\sqrt{5}}{2} - \frac{5}{2}$$

$$0 \leq -2X_{xy} + 3X_{yy} - \frac{\sqrt{5}}{2}y + 4x - \frac{7}{2}y + \sqrt{5} - 5$$



- Lower bound = -2.46 (improvement from -2.66)

Back to Example: Objective Cutoff

$\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\ x - y \leq 1,\ x \in [0,2], y \in [-1,2]\}$

Assume the optimal solution with objective $= \frac{\sqrt{5}-5}{2}$ has been found, e.g., by a NLP solver, but proof of optimality is still missing.

Objective cutoff: Look only for improving solutions: $-2x + 3y \leq \frac{\sqrt{5}-5}{2}$

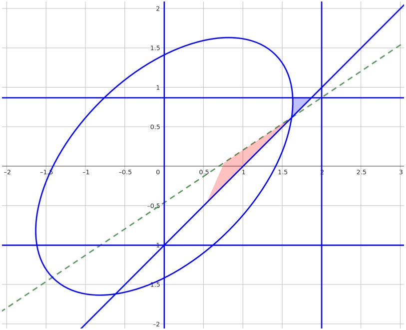
RLT with this inequality:

$$0 \leq 2X_{xx} - 3X_{xy} + \frac{\sqrt{5}}{2}x - \frac{5}{2}x$$

$$0 \leq -2X_{xx} + 3X_{xy} - \frac{\sqrt{5}}{2}x + \frac{13}{2}x - 6y + \sqrt{5} - 5$$

$$0 \leq 2X_{xy} - 3X_{yy} + \frac{\sqrt{5}}{2}y + 2x - \frac{11}{2}y + \frac{\sqrt{5}}{2} - \frac{5}{2}$$

$$0 \leq -2X_{xy} + 3X_{yy} - \frac{\sqrt{5}}{2}y + 4x - \frac{7}{2}y + \sqrt{5} - 5$$



- Lower bound = -2.46 (improvement from -2.66)

Use objective cutoff for bound tightening: $y \leq \frac{1}{3}\left(\frac{\sqrt{5}-5}{2} + 2x\right) \leq \frac{\sqrt{5}+3}{6} \approx 0.87$

More Bound Tightening

Looking at the LP relaxation including objective cutoff only, it seems that variable bounds could be improved further:

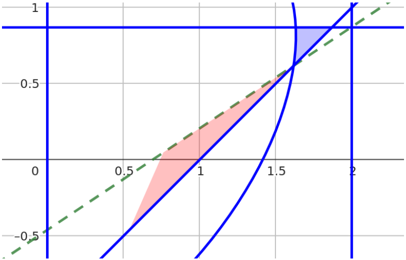
$$x - y \leq 1$$

$$-2x + 3y \leq \frac{\sqrt{5} - 5}{2}$$

$$\dots$$

$$x \in [0, 2],\ y \in [-1, 0.87]$$

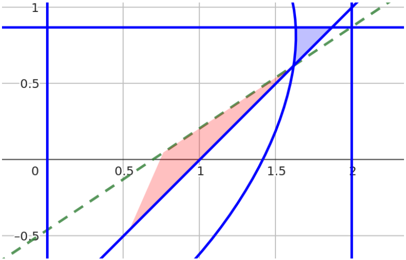
Apparently, $x \ll 2$.



More Bound Tightening

Looking at the LP relaxation including objective cutoff only, it seems that variable bounds could be improved further:

$$x - y \leq 1$$
$$-2x + 3y \leq \frac{\sqrt{5} - 5}{2}$$
$$\dots$$
$$x \in [0, 2],\ y \in [-1, 0.87]$$



Apparently, $x \ll 2$.
Propagating each inequality individually works:

$$x - y \leq 1 \Rightarrow x \leq 1.87$$
$$-2x + 3y \leq -1.38 \Rightarrow y \leq 0.79$$
$$x - y \leq 1 \Rightarrow x \leq 1.79$$
$$-2x + 3y \leq -1.38 \Rightarrow y \leq 0.73$$
$$\vdots$$

Belotti [2013]: FBBT on two linear constraints simultaneously

More Bound Tightening

Looking at the LP relaxation including objective cutoff only, it seems that variable bounds could be improved further:

$$x - y \leq 1$$

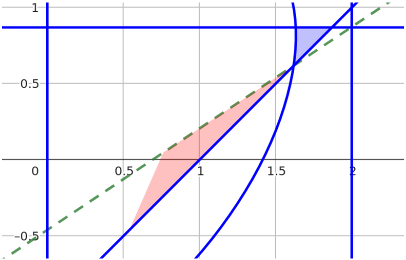
$$-2x + 3y \leq \frac{\sqrt{5} - 5}{2}$$

$$\dots$$

$$x \in [0, 2],\ y \in [-1, 0.87]$$

Apparently, $x \ll 2$.
Propagating each inequality individually works:

$$x - y \leq 1 \Rightarrow x \leq 1.87$$
$$-2x + 3y \leq -1.38 \Rightarrow y \leq 0.79$$
$$x - y \leq 1 \Rightarrow x \leq 1.79$$
$$-2x + 3y \leq -1.38 \Rightarrow y \leq 0.73$$
$$\vdots$$



Eventually, this terminates with upper bounds equal to

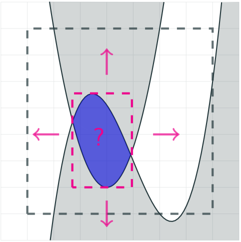
$$\max\{x : x - y \leq 1, -2x + 3y \leq -1.38\}$$
$$\max\{y : x - y \leq 1, -2x + 3y \leq -1.38\}$$

Idea: Just solve this LP!

Belotti [2013]: FBBT on two linear constraints simultaneously

In General: Optimization-based bound tightening

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

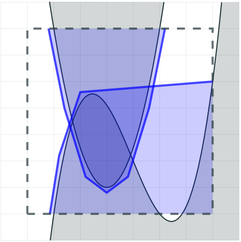


In General: Optimization-based bound tightening

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)



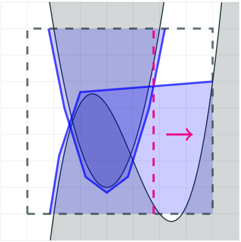
In General: Optimization-based bound tightening

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)

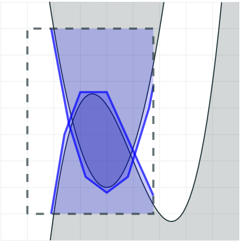


In General: Optimization-based bound tightening

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

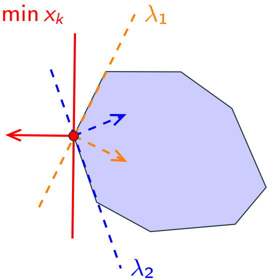


In General: Optimization-based bound tightening

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

Back to Example: Bound Tightening by OBBT

We tightened upper bounds via

$$\max\left\{x : x - y \leq 1, -2x + 3y \leq \frac{\sqrt{5} - 5}{2}\right\} = \frac{1 + \sqrt{5}}{2} \approx 1.62$$

$$\max\left\{y : x - y \leq 1, -2x + 3y \leq \frac{\sqrt{5} - 5}{2}\right\} = \frac{\sqrt{5} - 1}{2} \approx 0.62$$

Back to Example: Bound Tightening by OBBT

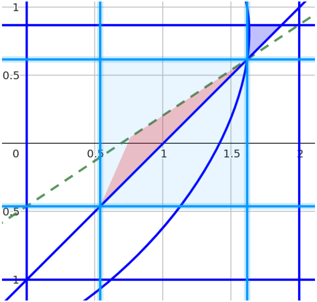
We tightened upper bounds via

$$\max\left\{x : x - y \le 1, -2x + 3y \le \frac{\sqrt{5}-5}{2}\right\} = \frac{1+\sqrt{5}}{2} \approx 1.62$$

$$\max\left\{y : x - y \le 1, -2x + 3y \le \frac{\sqrt{5}-5}{2}\right\} = \frac{\sqrt{5}-1}{2} \approx 0.62$$

To tighten also lower bounds, consider the complete relaxation:

$$\begin{aligned}
&\min x \text{ or } y\\
&\text{s.t. } x - y \le 1\\
&\quad -2x + 3y \le \frac{\sqrt{5}-5}{2}\\
&\quad X_{xx} - X_{xy} + X_{yy} \ge 2\\
&\quad \mathrm{RLT}(X, x, y),\\
&\quad x \in \left[0, \frac{1+\sqrt{5}}{2}\right], y \in \left[-1, \frac{\sqrt{5}-1}{2}\right]
\end{aligned}$$

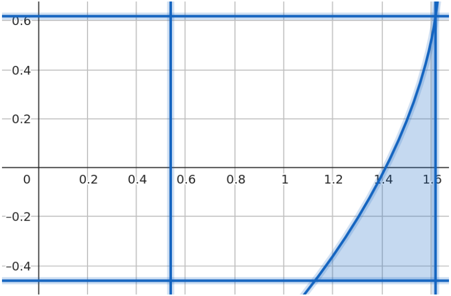


$\Rightarrow x \ge 0.54,\ y \ge -0.46$

FBBT on quadratic constraint

With the tighter bounds from OBBT, let us try to derive further boundtightening from the quadratic constraint, that is

$$\min / \max\{x \text{ or } y \;:\; x^2 - xy + y^2 \geq 2, x \in [0.54, 1.62], y \in [-0.46, 0.62]\}$$



For y we cannot expect any tightening, but what about the lower bound for x?

FBBT on quadratic constraint – do the math

$x^2 - xy + y^2 = (y - \frac{1}{2}x)^2 + \frac{3}{4}x^2$ is supposed to be ≥ 2

FBBT on quadratic constraint – do the math

$x^2 - xy + y^2 = (y - \frac{1}{2}x)^2 + \frac{3}{4}x^2$ is supposed to be ≥ 2

$\Rightarrow (x - \frac{1}{2}y)^2 \geq 2 - \frac{3}{4}y^2 \Rightarrow |x - \frac{1}{2}y| \geq \sqrt{2 - \frac{3}{4}y^2}$

$\Rightarrow x - \frac{1}{2}y \geq \sqrt{2 - \frac{3}{4}y^2}$ or $x - \frac{1}{2}y \leq -\sqrt{2 - \frac{3}{4}y^2}$

FBBT on quadratic constraint – do the math

$x^2 - xy + y^2 = (y - \frac{1}{2}x)^2 + \frac{3}{4}x^2$ is supposed to be ≥ 2

$\Rightarrow (x - \frac{1}{2}y)^2 \geq 2 - \frac{3}{4}y^2 \Rightarrow |x - \frac{1}{2}y| \geq \sqrt{2 - \frac{3}{4}y^2}$

$\Rightarrow x - \frac{1}{2}y \geq \sqrt{2 - \frac{3}{4}y^2}$ or $x - \frac{1}{2}y \leq -\sqrt{2 - \frac{3}{4}y^2}$

$$\Rightarrow x \in \left(\left[-\infty, \frac{1}{2}y - \sqrt{2 - \frac{3}{4}y^2}\right] \cup \left[\frac{1}{2}y + \sqrt{2 - \frac{3}{4}y^2}, \infty\right]\right) \cap [0.54, 1.62]$$

The right-hand side now depends on y only.

FBBT on quadratic constraint – do the math

$x^2 - xy + y^2 = (y - \frac{1}{2}x)^2 + \frac{3}{4}x^2$ is supposed to be ≥ 2

$\Rightarrow (x - \frac{1}{2}y)^2 \geq 2 - \frac{3}{4}y^2 \Rightarrow |x - \frac{1}{2}y| \geq \sqrt{2 - \frac{3}{4}y^2}$

$\Rightarrow x - \frac{1}{2}y \geq \sqrt{2 - \frac{3}{4}y^2}$ or $x - \frac{1}{2}y \leq -\sqrt{2 - \frac{3}{4}y^2}$

$$\Rightarrow x \in \left(\left[-\infty, \frac{1}{2}y - \sqrt{2 - \frac{3}{4}y^2}\right] \cup \left[\frac{1}{2}y + \sqrt{2 - \frac{3}{4}y^2}, \infty\right]\right) \cap [0.54, 1.62]$$

The right-hand side now depends on y only.

We now need to find

$$\max_{y \in [-0.46, 0.62]} \frac{1}{2}y - \sqrt{2 - \frac{3}{4}y^2} \qquad \min_{y \in [-0.46, 0.62]} \frac{1}{2}y + \sqrt{2 - \frac{3}{4}y^2}$$

FBBT on quadratic constraint – do the math

$x^2 - xy + y^2 = (y - \frac{1}{2}x)^2 + \frac{3}{4}x^2$ is supposed to be ≥ 2

$\Rightarrow (x - \frac{1}{2}y)^2 \geq 2 - \frac{3}{4}y^2 \Rightarrow |x - \frac{1}{2}y| \geq \sqrt{2 - \frac{3}{4}y^2}$

$\Rightarrow x - \frac{1}{2}y \geq \sqrt{2 - \frac{3}{4}y^2}$ or $x - \frac{1}{2}y \leq -\sqrt{2 - \frac{3}{4}y^2}$

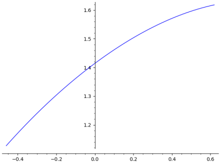
$$\Rightarrow x \in \left(\left[-\infty, \frac{1}{2}y - \sqrt{2 - \frac{3}{4}y^2}\right] \cup \left[\frac{1}{2}y + \sqrt{2 - \frac{3}{4}y^2}, \infty\right]\right) \cap [0.54, 1.62]$$

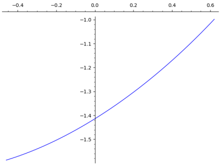
The right-hand side now depends on y only.

We now need to find

$$\max_{y \in [-0.46, 0.62]} \frac{1}{2}y - \sqrt{2 - \frac{3}{4}y^2} \qquad \min_{y \in [-0.46, 0.62]} \frac{1}{2}y + \sqrt{2 - \frac{3}{4}y^2}$$

These are univariate bound-constrained optimization problems.





FBBT on quadratic constraint – do the math (cont.)

$$\max_{y\in[-0.46,0.62]} \frac{1}{2}y - \sqrt{2-\frac{3}{4}y^2} \underbrace{=}_{y=0.62} \frac{0.62}{2} - \sqrt{2-\frac{3}{4}0.62^2} \qquad \approx -1$$

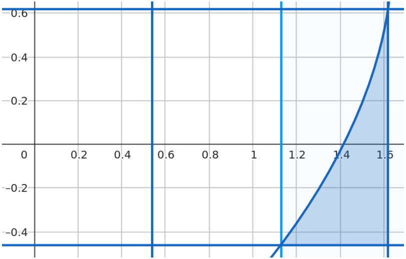
$$\min_{y\in[-0.46,0.62]} \frac{1}{2}y + \sqrt{2-\frac{3}{4}y^2} \underbrace{=}_{y=-0.46} -\frac{0.46}{2} + \sqrt{2-\frac{3}{4}(-0.46)^2} \qquad \approx 1.13$$

FBBT on quadratic constraint – do the math (cont.)

$$\max_{y\in[-0.46,0.62]} \frac{1}{2}y - \sqrt{2-\frac{3}{4}y^2} \underbrace{=}_{y=0.62} \frac{0.62}{2} - \sqrt{2-\frac{3}{4}0.62^2} \qquad \approx -1$$

$$\min_{y\in[-0.46,0.62]} \frac{1}{2}y + \sqrt{2-\frac{3}{4}y^2} \underbrace{=}_{y=-0.46} -\frac{0.46}{2} + \sqrt{2-\frac{3}{4}(-0.46)^2} \qquad \approx 1.13$$

$$\Rightarrow x \in \left(\left[-\infty, \underbrace{\frac{1}{2}y - \sqrt{2-\frac{3}{4}y^2}}_{\approx -1}\right] \cup \left[\underbrace{\frac{1}{2}y + \sqrt{2-\frac{3}{4}y^2}}_{\approx 1.13}, \infty\right]\right) \cap [0.54, 1.62] = [1.13, 1.62]$$



Note: feasible range on x is disconnected (2 intervals); we used $x \geq 0.54$ to exclude the left interval and derive $x \geq 1.13$

Vigerske and Gleixner [2017]: general formulas

Updated Relaxation after FBBT and OBBT

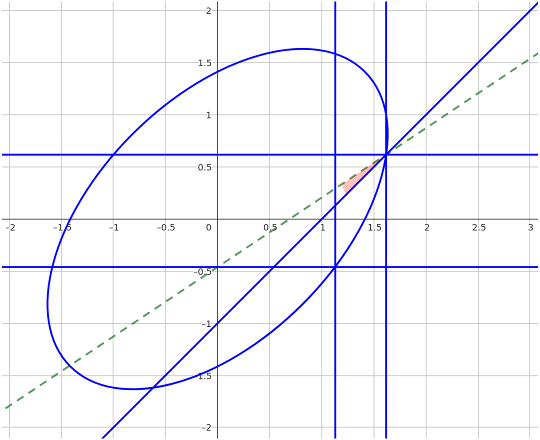
We derived

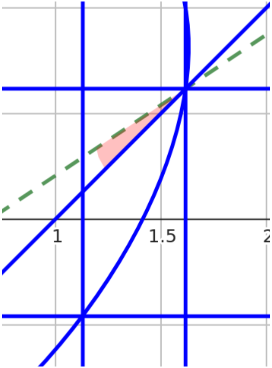
- $x \leq 1.62$, $y \leq 0.62$ via OBBT or alternating FBBT on $x - y \leq 1$ and $-2x + 3y \leq -1.38$
- $y \geq -0.46$ via OBBT on LP relaxation (incl. RLT cuts)
- $x \geq 1.13$ via careful (avoid overestimation of interval arith.) FBBT on $x^2 - xy + y^2 \geq 2$

Update RLT:

$$
\begin{aligned}
&0 \leq (x-1.13)^2 && 0 \leq (x-1.13)(1-x+y) \\
&0 \leq (1.62-x)^2 && 0 \leq (1.62-x)(1-x+y) \\
&0 \leq (x-1.13)(1.62-x) && 0 \leq (y+0.46)(1-x+y) \\
& && 0 \leq (0.62-y)(1-x+y) \\
&0 \leq (y+0.46)^2 && \\
&0 \leq (0.62-y)^2 && 0 \leq (x-1.13)(-1.38+2x-3y) \\
&0 \leq (0.62-y)(y+0.46) && 0 \leq (1.62-x)(-1.38+2x-3y) \\
& && 0 \leq (y+0.46)(-1.38+2x-3y) \\
&0 \leq (x-1.13)(y+0.46) && 0 \leq (0.62-y)(-1.38+2x-3y) \\
&0 \leq (x-1.13)(0.62-y) && \\
&0 \leq (1.62-x)(y+0.46) && xx \to X_{xx},\ xy \to X_{xy},\ yy \to X_{yy} \\
&0 \leq (1.62-x)(0.62-y) &&
\end{aligned}
$$

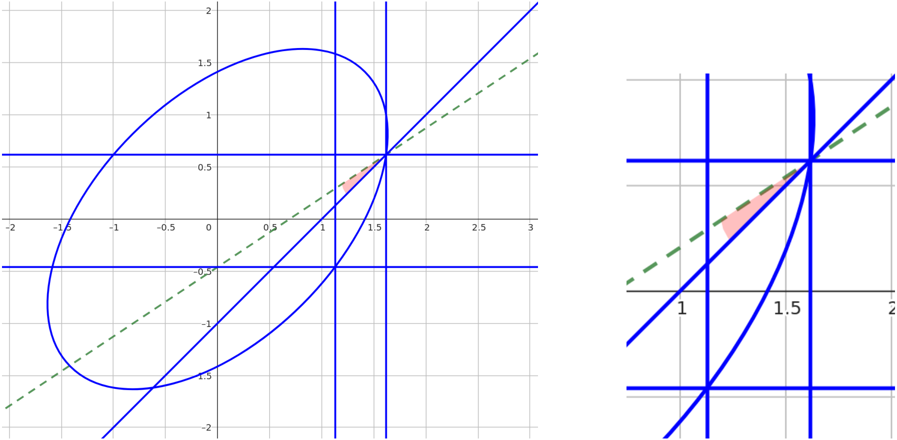
Updated Relaxation (cont.)





Lower bound = -1.76 (improvement from -2.46, optimal value = -1.38)

Updated Relaxation (cont.)



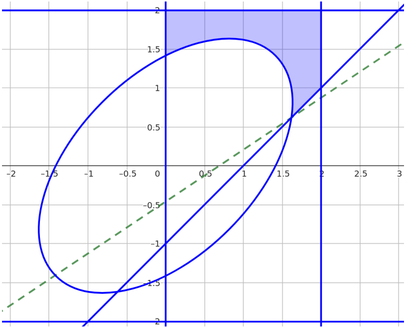
Lower bound = -1.76 (improvement from -2.46, optimal value = -1.38)

Next steps:

- OBBT improves lower bound on y due to tighter RLT cuts
- FBBT on quad. cons. improves lower bound on x due to better bound on y
- RLT cuts tighten due to better lower bounds on x and y

Recap

Problem: $\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2, \, x - y \leq 1, \, x \in [0,2], y \in [-2,2]\}$



Recap

Problem: $\min\{-2x + 3y \;:\; x^2 - xy + y^2 \geq 2,\, x - y \leq 1,\, x \in [0,2], y \in [-2,2]\}$

Initial Relaxation:

- replace any square and bilinear term by new variable (X)
- derive cuts for X by multiplying variable bounds, e.g., $(2-x)(2-y) \geq 0$ (also known as McCormick cuts)

LP Relaxation:

$$\min \; -2x + 3y$$

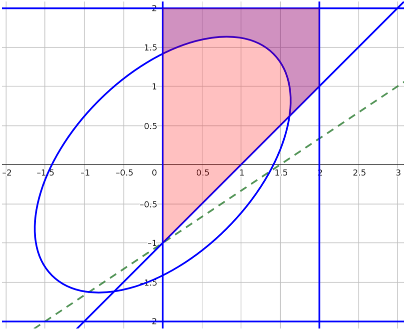
$$\text{s.t. } X_{xx} - X_{xy} + X_{yy} \geq 2$$

$$x - y \leq 1$$

RLT(multiply bounds)

$x \in [0,2]$

$y \in [-2,2]$



Lower bound = -3

Recap

Problem: $\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\ x - y \leq 1,\ x \in [0, 2], y \in [-2, 2]\}$

Bound Tightening:

- FBBT on linear constraint: $x - y \leq 1 \Rightarrow y \geq -1$

LP Relaxation:

$$\min \ -2x + 3y$$

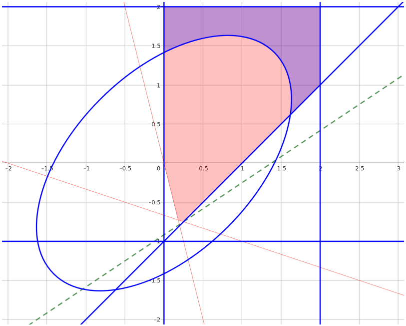
$$\text{s.t. } X_{xx} - X_{xy} + X_{yy} \geq 2$$

$$x - y \leq 1$$

RLT(multiply bounds)

$x \in [0, 2]$

$y \in [-1, 2]$



Lower bound = -2.75

Recap

Problem: $\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\ x - y \leq 1,\ x \in [0,2], y \in [-2,2]\}$

RLT with Linear Inequality:

- multiply $x - y \leq 1$ with variable bound, e.g., $(2 - x)(1 - x + y) \geq 0$

LP Relaxation:

$$\min \ -2x + 3y$$

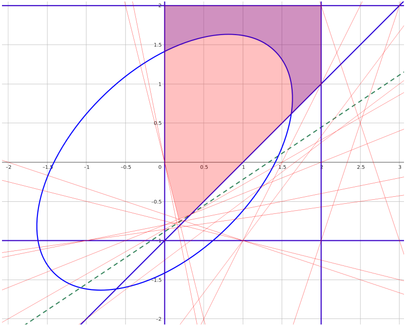
$$\text{s.t. } X_{xx} - X_{xy} + X_{yy} \geq 2$$

$$x - y \leq 1$$

RLT(bounds & $x - y \leq 1$)

$x \in [0, 2]$

$y \in [-1, 2]$



Lower Bound = -2.66

Recap

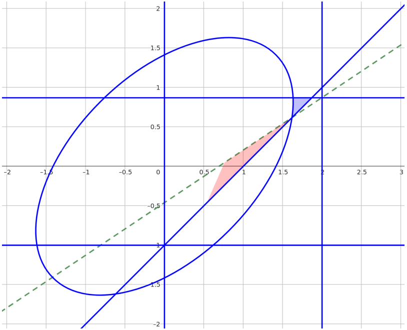
Problem: $\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\, x - y \leq 1,\, x \in [0,2], y \in [-2,2]\}$

Objective Cutoff:

- look only for improving solutions: $-2x + 3y \leq -1.36$
- use for FBBT and RLT (improving upper bound can improve lower bound!)

LP Relaxation:

$$\begin{aligned} \min\ & -2x + 3y \\ \text{s.t. } & X_{xx} - X_{xy} + X_{yy} \geq 2 \\ & x - y \leq 1 \\ & -2x + 3y \leq 1.38 \\ & \text{RLT(bounds \& linear inequ.)} \\ & x \in [0,2] \\ & y \in [-1, 0.87] \end{aligned}$$



Lower Bound = -2.46

Recap

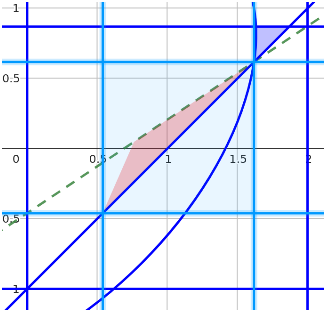
Problem: $\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\ x - y \leq 1,\ x \in [0,2], y \in [-2,2]\}$

Bound Tightening:

- OBBT on relaxation: min / max x or y w.r.t. LP relaxation
- expensive, best when objective cutoff included

LP Relaxation:

$$
\begin{aligned}
\min\ & -2x + 3y \\
\text{s.t. } & X_{xx} - X_{xy} + X_{yy} \geq 2 \\
& x - y \leq 1 \\
& -2x + 3y \leq 1.38 \\
& \text{RLT(bounds \& linear inequ.)} \\
& x \in [0.54, 1.62] \\
& y \in [-0.46, 0.62]
\end{aligned}
$$



Recap

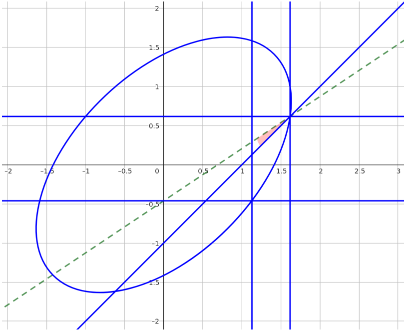
Problem: $\min\{-2x + 3y \ : \ x^2 - xy + y^2 \geq 2,\ x - y \leq 1,\ x \in [0, 2], y \in [-2, 2]\}$

Bound Tightening:

- FBBT on $x^2 - xy + y^2 \geq 2 \Rightarrow x \geq 1.13$

LP Relaxation:

$$
\begin{aligned}
\min\ & -2x + 3y \\
\text{s.t. } & X_{xx} - X_{xy} + X_{yy} \geq 2 \\
& x - y \leq 1 \\
& -2x + 3y \leq 1.38 \\
& \text{RLT(bounds \& linear inequ.)} \\
& x \in [1.13, 1.62] \\
& y \in [-0.46, 0.62]
\end{aligned}
$$



Lower bound = -1.76

Further Techniques

Further Techniques

Dual Side (Tighter Relaxations)

Semidefinite Programming (SDP) Relaxation

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

- 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
- Anstreicher [2009]: the SDP and RLT relaxations do not dominate each other; combining both can produce substantially better bounds

SDP Cuts

SDP is computationally demanding, so approximate by linear inequalities:

- for $\tilde{X}^* \not\succeq 0$ compute eigenvector v with eigenvalue $\lambda < 0$, then

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

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

- these cuts can be very dense (involve many variables), which slows down the LP solver

SDP Cuts

SDP is computationally demanding, so approximate by linear inequalities:

- for $\tilde{X}^* \not\succeq 0$ compute eigenvector v with eigenvalue $\lambda < 0$, then

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

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

- these cuts can be very dense (involve many variables), which slows down the LP solver

Approaches for sparser cuts:

- Qualizza et al. [2009]: relax cut by setting entries of v to 0
- Saxena et al. [2011]: project into x-variables space (no X variables in cut)
- Sherali et al. [2012]: consider only a subset of variables and corresponding submatrix of X
  - Baltean-Lugojan et al. [2018]: pick submatrix via neural network
  - SCIP [Bestuzheva et al., 2021]: consider only two variables and corresponding 2×2 submatrix of X

Second Order Cones (SOC)

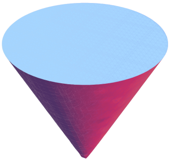
Consider a constraint $x^\mathsf{T} A x + b^\mathsf{T} x \leq c.$

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

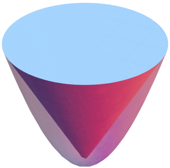
$$\sum_{k=1}^{N} x_k^2 - x_{N+1}^2 \leq 0, x_{N+1} \geq 0 \quad \Leftrightarrow \quad \sqrt{\sum_{k=1}^{N} x_k^2} \leq x_{N+1}$$

- $\sqrt{\sum_{k=1}^{N} x_k^2}$ is a convex term that can easily be linearized

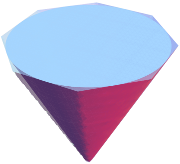
Example: $x^2 + y^2 - z^2 \leq 0$ in $[-1, 1] \times [-1, 1] \times [0, 1]$



feasible region



not recognizing SOC



recognizing SOC

(initial relaxation)

Cone Disaggregation

For high-dimensional cones (large N), linearizations of $\sqrt{\sum_{k=1}^{N} x_k^2}$ generate dense cuts $\Rightarrow$ slow LP solves.

Cone Disaggregation

For high-dimensional cones (large N), linearizations of $\sqrt{\sum_{k=1}^N x_k^2}$ generate dense cuts $\Rightarrow$ slow LP solves.

Vielma et al. [2016]: consider disaggregated formulation in extended space:

- introduce new variables z_k, $k = 1, \dots, N$ and add constraints

$$z_k \geq \frac{x_k^2}{x_{N+1}}, \qquad \sum_{k=1}^{N} z_k \leq x_{N+1}$$

- then SOC $\sum_k x_k^2 \leq x_{N+1}^2$ is satisfied because

$$\frac{1}{x_{N+1}} \sum_{k=1}^{N} x_k^2 \leq \sum_{k=1}^{N} z_k \leq x_{N+1}$$

- new cons. ${x_k^2}/{x_{N+1}} \leq z_k$ are 3-dimensional SOC:

$$x_k^2 \leq z_k x_{N+1} = {1}/{4}\left((z_k + x_{N+1})^2 - (z_k - x_{N+1})^2\right)$$

$$\Leftrightarrow \sqrt{4x_k^2 + (z_k - x_{N+1})^2} \leq z_k + x_{N+1}$$

$\Downarrow$

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 et al., 2009, Vigerske, 2013]

Analyze Expression for Hessian: Klaus, Merk, Wiedom, Laue, and Giesen [2022]

Stronger relaxations with semi-continuous variables

Consider

$$x^2 \leq w, \qquad \ell y \leq x \leq uy, \quad y \in \{0, 1\}, \qquad \text{(with } \ell > 0\text{)}.$$

That is, $x \in \{0\} \cup [\ell, u]$.

Stronger relaxations with semi-continuous variables

Consider

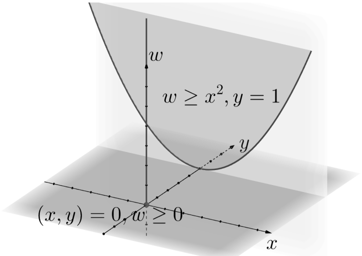
$$x^2 \leq w, \qquad \ell y \leq x \leq uy, \quad y \in \{0,1\}, \qquad (\text{with } \ell > 0).$$

That is, $x \in \{0\} \cup [\ell, u]$.

A tight relaxation would be the convex hull of relaxations for $y = 0$ and $y = 1$:

$$\operatorname{conv}\left(\underbrace{\{(0,w,0) \;:\; w \geq 0\}}_{y=0} \;\cup\; \underbrace{\{(x,w,1) \;:\; x^2 \leq w, x \in [\ell,u]\}}_{y=1} \right)$$

By just relaxing $y \in \{0,1\}$ to $y \in [0,1]$, one does not get this set.



Stronger relaxations with semi-continuous variables

Consider

$$x^2 \leq w, \qquad \ell y \leq x \leq uy, \quad y \in \{0,1\}, \qquad (\text{with } \ell > 0).$$

That is, $x \in \{0\} \cup [\ell, u]$.

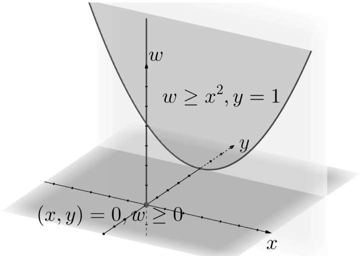
A tight relaxation would be the convex hull of relaxations for $y = 0$ and $y = 1$:

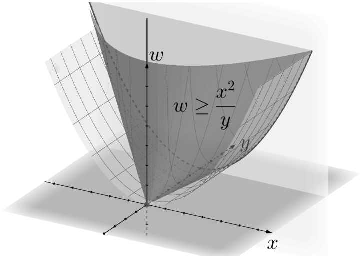
$$\text{conv}\left(\underbrace{\{(0, w, 0) \ : \ w \geq 0\}}_{y=0} \ \cup \ \underbrace{\{(x, w, 1) \ : \ x^2 \leq w, x \in [\ell, u]\}}_{y=1} \right)$$

By just relaxing $y \in \{0,1\}$ to $y \in [0,1]$, one does not get this set.

However, replacing $x^2 \leq w$ by the SOC $x^2 \leq wy$ and $w \geq 0$ is sufficient.

[Günlük and Linderoth, 2012]





Why $x^2 \leq wy$?

$$\text{conv}(\{(0, w, 0) \, : \, w \geq 0\} \, \cup \, \{(x, w, 1) \, : \, x^2 \leq w, x \in [\ell, u]\})$$

Why $x^2 \leq wy$?

$$\text{conv}(\underbrace{\{(0,w,0) : w \geq 0\}}_{\ni(x_0,w_0,y_0)} \cup \underbrace{\{(x,w,1) : x^2 \leq w,\ x \in [\ell,u]\}}_{\ni(x_1,w_1,y_1)})$$

$$= \left\{ (x,w,y) : \begin{array}{l} x = \lambda x_1 + (1-\lambda)x_0, \\ w = \lambda w_1 + (1-\lambda)w_0, \\ y = \lambda y_1 + (1-\lambda)y_0, \\ x_0 = 0, y_0 = 0, w_0 \geq 0, \\ x_1^2 \leq w_1,\ x_1 \in [\ell,u],\ y_1 = 1 \\ \lambda \in [0,1] \end{array} \right\}$$

Why $x^2 \leq wy$?

$$\text{conv}(\underbrace{\{(0,w,0) \,:\, w \geq 0\}}_{\ni(x_0,w_0,y_0)} \cup \underbrace{\{(x,w,1) \,:\, x^2 \leq w,\, x \in [\ell,u]\}}_{\ni(x_1,w_1,y_1)})$$

$$= \left\{ (x,w,y) \,:\, \begin{array}{l} x = \lambda x_1 \\ w = \lambda w_1 + (1-\lambda) w_0, \\ y = \lambda \\ \qquad\qquad\qquad w_0 \geq 0, \\ x_1^2 \leq w_1,\; x_1 \in [\ell,u], \\ \lambda \in [0,1] \end{array} \right\}$$

Why $x^2 \le wy$?

$$\operatorname{conv}\Big(\underbrace{\{(0,w,0)\,:\,w\ge 0\}}_{\ni(x_0,w_0,y_0)} \cup \underbrace{\{(x,w,1)\,:\,x^2\le w,\ x\in[\ell,u]\}}_{\ni(x_1,w_1,y_1)}\Big)$$

$$= \left\{ (x,w,y)\ :\ \begin{array}{l} x = \lambda x_1 \\ w = \lambda w_1 + (1-\lambda) w_0, \\ y = \lambda \\ \qquad\qquad\qquad w_0 \ge 0, \\ x_1^2 \le w_1,\ x_1\in[\ell,u], \\ \lambda\in[0,1] \end{array} \right\}$$

$$= \underbrace{\left\{ (x,w,y)\ :\ \begin{array}{l} \left(\frac{x}{y}\right)^2 \le \frac{w}{y},\ \frac{x}{y}\in[\ell,u], \\ y\in(0,1] \end{array} \right\}}_{\text{for } w_0=0,\lambda>0,\quad \text{using } x_1=x/\lambda, w_1=w/\lambda, \lambda=y} \cup \underbrace{\{(0,w,0)\,:\,w\ge 0\}}_{\text{for } w_0\ge 0,\lambda=0}$$

Why $x^2 \leq wy$?

$$\operatorname{conv}\Big(\underbrace{\{(0,w,0) \,:\, w \geq 0\}}_{\ni(x_{\mathbf{0}},w_{\mathbf{0}},y_{\mathbf{0}})} \cup \underbrace{\{(x,w,1) \,:\, x^2 \leq w,\ x \in [\ell,u]\}}_{\ni(x_{\mathbf{1}},w_{\mathbf{1}},y_{\mathbf{1}})}\Big)$$

$$= \left\{ (x,w,y) \,:\, \begin{array}{l} x = \lambda x_1 \\ w = \lambda w_1 + (1-\lambda) w_0, \\ y = \lambda \\ \qquad\qquad\qquad w_0 \geq 0, \\ x_1^2 \leq w_1,\ x_1 \in [\ell,u], \\ \lambda \in [0,1] \end{array} \right\}$$

$$= \underbrace{\left\{ (x,w,y) \,:\, \begin{array}{l} \left(\frac{x}{y}\right)^2 \leq \frac{w}{y},\ \frac{x}{y} \in [\ell,u], \\ y \in (0,1] \end{array} \right\}}_{\text{for } w_{\mathbf{0}}=0,\lambda>0, \quad \text{using } x_{\mathbf{1}}=x/\lambda, w_{\mathbf{1}}=w/\lambda, \lambda=y} \cup \underbrace{\{(0,w,0) \,:\, w \geq 0\}}_{\text{for } w_{\mathbf{0}}\geq 0,\lambda=0}$$

$$= \left\{ (x,w,y) \,:\, x^2 \leq wy,\ \ell y \leq x \leq uy,\ w \geq 0,\ y \in [0,1] \right\}$$

Convex Hull of Point and Convex Set

More general, consider

$$\{(0,0)\} \quad \cup \quad \{(x,1) \; : \; f(x) \leq 0, \, \ell \leq x \leq u\} \qquad (f \text{ convex})$$

As before, build the convex combination of both sets and eliminate variables:

$$\{(x,y) \; : \; f(x/y) \leq 0, \, \ell y \leq x \leq uy, \; y \in (0,1]\} \cup \{(0,0)\}$$

[Günlük and Linderoth, 2012]

Convex Hull of Point and Convex Set

More general, consider

$$\{(0,0)\} \quad \cup \quad \{(x,1) \,:\, f(x) \leq 0,\, \ell \leq x \leq u\} \qquad (f \text{ convex})$$

As before, build the convex combination of both sets and eliminate variables:

$$\begin{aligned} &\{(x,y) \,:\, f(x/y) \leq 0,\, \ell y \leq x \leq uy,\, y \in (0,1]\} \cup \{(0,0)\} \\ =\; &\{(x,y) \,:\, \tilde{f}(x,y) \leq 0,\, \ell y \leq x \leq uy,\, y \in [0,1]\}, \end{aligned}$$

where $\tilde{f}(x,y) = \begin{cases} y\, f(x/y), & \text{if } y > 0, \\ 0, & \text{if } y = 0, \\ \infty, & \text{otherwise,} \end{cases}$ is the perspective function of $f(x)$.

Important property: If f is convex, then $\tilde{f}$ is convex.

[Günlük and Linderoth, 2012]

Perspective Cuts

Applying the perspective reformulation (replacing $f(x)$ by $\tilde{f}(x, y)$) in a problem can be problematic, because $\tilde{f}(x, y)$ is not differentiable at $y = 0$.

Frangioni and Gentile [2006]: effect of perspective reformulation can be captured in LP relaxation by supporting hyperplanes on the epigraph of $\tilde{f}(x, y)$:

Perspective Cuts

Applying the perspective reformulation (replacing $f(x)$ by $\tilde{f}(x,y)$) in a problem can be problematic, because $\tilde{f}(x,y)$ is not differentiable at $y=0$.

Frangioni and Gentile [2006]: effect of perspective reformulation can be captured in LP relaxation by supporting hyperplanes on the epigraph of $\tilde{f}(x,y)$:

- linearization of $f(x) \le 0$ at $x = \hat{x}$:

$$f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \le 0$$

- perspective cut tilts cut to be tight at $(x,y) = (0,0)$ by adding $(f(0) - f(\hat{x}) + \nabla f(\hat{x})\hat{x})(1-y)$:

$$f(\hat{x})y + \nabla f(\hat{x})(x - \hat{x}y) + f(0)(1-y) \le 0$$

Check: $y = 0 \Rightarrow x = 0 \Rightarrow$ left-hand-side $= f(0)$
$y = 1 \Rightarrow$ left-hand-side $= f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x})$

Perspective Cuts

Applying the perspective reformulation (replacing $f(x)$ by $\tilde{f}(x,y)$) in a problem can be problematic, because $\tilde{f}(x,y)$ is not differentiable at $y=0$.

Frangioni and Gentile [2006]: effect of perspective reformulation can be captured in LP relaxation by supporting hyperplanes on the epigraph of $\tilde{f}(x,y)$:

- linearization of $f(x) \leq 0$ at $x = \hat{x}$:

$$f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq 0$$

- perspective cut tilts cut to be tight at $(x,y) = (0,0)$ by adding $(f(0) - f(\hat{x}) + \nabla f(\hat{x})\hat{x})(1-y)$:

$$f(\hat{x})y + \nabla f(\hat{x})(x - \hat{x}y) + f(0)(1-y) \leq 0$$

  Check: $y = 0 \Rightarrow x = 0 \Rightarrow$ left-hand-side $= f(0)$
  $y = 1 \Rightarrow$ left-hand-side $= f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x})$

- example: $f(x) = x^2$, $\hat{x} = 1$
  - linearization cut: $1 + 2(x-1) \leq 0$; at $x = 0$: $-1 \leq 0 \Rightarrow$ not active
  - perspective cut: $y + 2(x-y) \leq 0$; at $(x,y) = (0,0)$: $0 \leq 0 \Rightarrow$ active, thus tighter

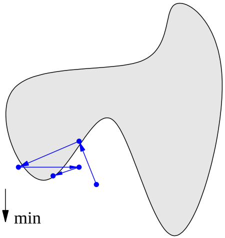
Further Techniques

Primal Side (Find Feasible Solutions)

Sub-NLP Heuristics

Given a solution satisfying all integrality constraints,

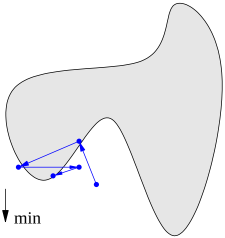
- fix all integer variables in the MINLP
- call an NLP solver to find a local solution to the remaining NLP



Sub-NLP Heuristics

Given a solution satisfying all integrality constraints,

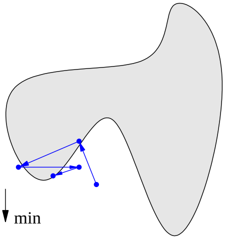
- fix all integer variables in the MINLP
- call an NLP solver to find a local solution to the remaining NLP
- variable fixings given by integer-feasible solution to LP relaxation (maybe from running MIP heuristics on MIP relaxation)



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 (maybe from running MIP heuristics on MIP relaxation)



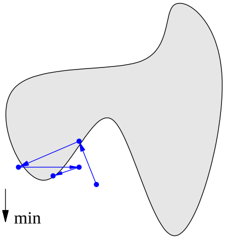
Multistart: run local NLP solver from random starting points to increase likelihood of finding global optimum

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

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 (maybe from running MIP heuristics on MIP relaxation)



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

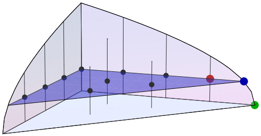
Smith, Chinneck, and Aitken [2013]: sample many random starting points, move them cheaply towards feasible region (average gradients of violated constraints), cluster, run NLP solvers from (few) center of cluster

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

Sub-MIP / Sub-MINLP Heuristics

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

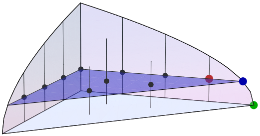
- Fix nonlinear variables, so problem becomes MIP
- not always necessary to fix all nonlinear variables, e.g., consider $x \cdot y$
- find a minimal set of variables to fix by solving a Set Covering Problem



Sub-MIP / Sub-MINLP Heuristics

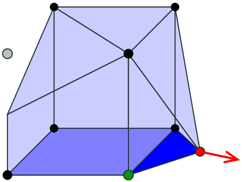
"Undercover" (SCIP) [Berthold and Gleixner, 2014]:

- Fix nonlinear variables, so problem becomes MIP
- 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 et al., 2011]:

- RENS [Berthold, 2014]: fix integer variables with integral value in LP relaxation
- RINS, DINS, Crossover, Local Branching



Alternating Direction

Feasibility Pump [D'Ambrosio, Frangioni, Liberti, and Lodi, 2010, 2012, Belotti and Berthold, 2017]:

- originally for MIP [Fischetti, Glover, and Lodi, 2005]
- MINLP: 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)
- Geißler, Morsi, Schewe, and Schmidt [2017]: modifications for convergent algorithm (avoid cycling)

Solver Software

Solvers

The following gives a list of MINLP solvers.

- it is incomplete
- omitted solvers that do not seem to be maintained anymore
- omitted continuous-only (NLP) solvers, e.g., COCONUT [Neumaier, 2001], Ibex (`http://www.ibex-lib.org`), RAPOSa [González-Rodríguez et al., 2022], ...
- omitted solvers without guarantee for global optimality, e.g., LocalSolver
- solver surveys:
  - Kronqvist, Bernal, Lundell, and Grossmann [2019]
  - Bussieck and Vigerske [2010]

Solver Software

Solvers for Mixed-Integer Quadratic Programs

Solvers for Mixed-Integer Quadratic Programs

CPLEX: https://www.ibm.com/products/ilog-cplex-optimization-studio

- commercial solver by IBM, unclear future
- available for all modeling languages and APIs to many languages
- convex quadratic objective and constraints
- second-order cone (SOC) constraints
- nonconvex quadratic objective (spatial branch-and-bound)
- branch-and-bound with LP and SOCP (SOC program) relaxation

Solvers for Mixed-Integer Quadratic Programs

CPLEX: https://www.ibm.com/products/ilog-cplex-optimization-studio

- commercial solver by IBM, unclear future
- available for all modeling languages and APIs to many languages
- convex quadratic objective and constraints
- second-order cone (SOC) constraints
- nonconvex quadratic objective (spatial branch-and-bound)
- branch-and-bound with LP and SOCP (SOC program) relaxation

GUROBI: https://www.gurobi.com

- commercial solver by GUROBI
- available for many modeling languages and APIs to many languages
- convex and nonconvex quadratic objective and constraints
- SOC constraints
- branch-and-bound with LP and SOCP (SOC program) relaxation

Solvers for Mixed-Integer Quadratic Programs (cont.)

MINOTAUR: [Mahajan, Leyffer, Linderoth, Luedtke, and Munson, 2021]

https://github.com/coin-or/minotaur

- open-source solver by IIT Bombay, Argonne Lab, and UW Madison
- available for AMPL and C++ API
- convex and nonconvex quadratic objective and constraints
- spatial branch-and-bound with LP relaxation

Solvers for Mixed-Integer Quadratic Programs (cont.)

MINOTAUR: [Mahajan, Leyffer, Linderoth, Luedtke, and Munson, 2021]

https://github.com/coin-or/minotaur

- open-source solver by IIT Bombay, Argonne Lab, and UW Madison
- available for AMPL and C++ API
- convex and nonconvex quadratic objective and constraints
- spatial branch-and-bound with LP relaxation

MOSEK: https://www.mosek.com

- commercial solver by MOSEK ApS
- available for many modeling languages and APIs to many languages
- convex quadratic objectives and constraints
- SOC constraints
- branch-and-bound with LP and SOCP (SOC program) relaxation
- also SDP and some other cones

Solvers for Mixed-Integer Quadratic Programs (cont.)

Pajarito: [Coey, Lubin, and Vielma, 2020] https://github.com/jump-dev/Pajarito.jl

- open-source solver by Chris Coey, Miles Lubin, and Juan Pablo Vielma
- available for JuMP, implemented in Julia
- SOC constraints, and other cones
- outer-approximation algorithm

Solvers for Mixed-Integer Quadratic Programs (cont.)

Pajarito: [Coey, Lubin, and Vielma, 2020] https://github.com/jump-dev/Pajarito.jl

- open-source solver by Chris Coey, Miles Lubin, and Juan Pablo Vielma
- available for JuMP, implemented in Julia
- SOC constraints, and other cones
- outer-approximation algorithm

SMIQP: [Elloumi and Lambert, 2019] https://github.com/amelie-lambert/SMIQP

- open-source solver by Amélie Lambert (CNAM CEDRIC, Paris)
- spatial branch-and-bound with quadratic convex relaxation (constructed via QCR method)

Solvers for Mixed-Integer Quadratic Programs (cont.)

Pajarito: [Coey, Lubin, and Vielma, 2020] https://github.com/jump-dev/Pajarito.jl

- open-source solver by Chris Coey, Miles Lubin, and Juan Pablo Vielma
- available for JuMP, implemented in Julia
- SOC constraints, and other cones
- outer-approximation algorithm

SMIQP: [Elloumi and Lambert, 2019] https://github.com/amelie-lambert/SMIQP

- open-source solver by Amélie Lambert (CNAM CEDRIC, Paris)
- spatial branch-and-bound with quadratic convex relaxation (constructed via QCR method)

XPRESS: https://www.fico.com/en/products/fico-xpress-optimization

- commercial solver by FICO
- available for many modeling languages and APIs to many languages
- convex quadratic objective and constraints
- second-order cone (SOC) constraints
- global MINLP solver announced

Solver Software

Solvers for Convex MINLP

Solvers for Convex MINLP

AOA: https://documentation.aimms.com/platform/solvers/aoa.html

- integrated in AIMMS modeling system
- outer-approximation algorithm

DICOPT: [Kocis and Grossmann, 1989]

https://distdocs.gams.com/41/docs/S_DICOPT.html

- integrated in GAMS modeling system
- outer-approximation algorithm

Solvers for Convex MINLP

AOA: `https://documentation.aimms.com/platform/solvers/aoa.html`

- integrated in AIMMS modeling system
- outer-approximation algorithm

DICOPT: [Kocis and Grossmann, 1989]

`https://distdocs.gams.com/41/docs/S_DICOPT.html`

- integrated in GAMS modeling system
- outer-approximation algorithm

Juniper: [Kröger, Coffrin, Hijazi, and Nagarajan, 2018]

`https://github.com/lanl-ansi/juniper.jl`

- open-source solver by Los Alamos Lab
- available for JuMP, implemented in Julia
- NLP-based branch-and-bound

Solvers for Convex MINLP (cont.)

Knitro: https://www.artelys.com/solvers/knitro

- commercial solver by Artelys
- available for several modeling systems and many APIs
- LP/NLP-based branch-and-bound, mixed-integer sequential quadratic programming

Solvers for Convex MINLP (cont.)

Knitro: https://www.artelys.com/solvers/knitro

- commercial solver by Artelys
- available for several modeling systems and many APIs
- LP/NLP-based branch-and-bound, mixed-integer sequential quadratic programming

MINOTAUR: [Mahajan, Leyffer, Linderoth, Luedtke, and Munson, 2021]
https://github.com/coin-or/minotaur

- open-source solver by IIT Bombay, Argonne Lab, and UW Madison
- available for AMPL and C++ API
- LP-, QP-, and NLP-based branch-and-bound with fast warmstarts, outer-approximation

Solvers for Convex MINLP (cont.)

Knitro: https://www.artelys.com/solvers/knitro

- commercial solver by Artelys
- available for several modeling systems and many APIs
- LP/NLP-based branch-and-bound, mixed-integer sequential quadratic programming

MINOTAUR: [Mahajan, Leyffer, Linderoth, Luedtke, and Munson, 2021]
https://github.com/coin-or/minotaur

- open-source solver by IIT Bombay, Argonne Lab, and UW Madison
- available for AMPL and C++ API
- LP-, QP-, and NLP-based branch-and-bound with fast warmstarts, outer-approximation

Muriqui: [Melo, Fampa, and Raupp, 2020] https://wendelmelo.net/software

- open-source solver by Wendel Melo, Marcia Fampa, and Fernanda Raupp
- available for AMPL and GAMS and C++ API
- LP/NLP-based branch-and-bound, outer-approximation, various hybrids

Solvers for Convex MINLP (cont.)

Pavito: https://github.com/jump-dev/Pavito.jl

- open-source solver by Chris Coey, Miles Lubin, and Juan P. Vielma
- available for JuMP, implemented in Julia
- LP/NLP-based branch-and-bound, outer-approximation
- sibling of Pajarito [Coey et al., 2020]

Solvers for Convex MINLP (cont.)

Pavito: https://github.com/jump-dev/Pavito.jl

- open-source solver by Chris Coey, Miles Lubin, and Juan P. Vielma
- available for JuMP, implemented in Julia
- LP/NLP-based branch-and-bound, outer-approximation
- sibling of Pajarito [Coey et al., 2020]

SHOT: [Lundell, Kronqvist, and Westerlund, 2022, Lundell and Kronqvist, 2022]
https://shotsolver.dev

- open-source solver by Andreas Lundell and Jan Kronqvist
- available for AMPL and GAMS, Mathematica, C++ API
- LP-based branch-and-bound and outer-approximation with supporting hyperplanes (EHP algorithm)
- can utilize GUROBI for nonconvex quadratics

Solvers for Convex MINLP (cont.)

Pavito: https://github.com/jump-dev/Pavito.jl

- open-source solver by Chris Coey, Miles Lubin, and Juan P. Vielma
- available for JuMP, implemented in Julia
- LP/NLP-based branch-and-bound, outer-approximation
- sibling of Pajarito [Coey et al., 2020]

SHOT: [Lundell, Kronqvist, and Westerlund, 2022, Lundell and Kronqvist, 2022]
https://shotsolver.dev

- open-source solver by Andreas Lundell and Jan Kronqvist
- available for AMPL and GAMS, Mathematica, C++ API
- LP-based branch-and-bound and outer-approximation with supporting hyperplanes (EHP algorithm)
- can utilize GUROBI for nonconvex quadratics

XPRESS-SLP: https://www.fico.com/en/products/fico-xpress-optimization

- commercial solver by FICO
- available for several modeling systems, several APIs
- mixed-integer sequential linear programming (NLP-based branch-and-bound or sequence of MIP approximations)

Solver Software

Solvers for General MINLP

Solvers for General MINLP

Alpine: [Nagarajan, Lu, Yamangil, and Bent, 2016, Nagarajan, Lu, Wang, Bent, and Sundar, 2019] `https://github.com/lanl-ansi/Alpine.jl`

- open-source solver by LANL-ANSI (Los Alamos)
- available for JuMP, implemented in Julia
- at most polynomials
- adaptive, piecewise-linear McCormick convexification scheme

Solvers for General MINLP

Alpine: [Nagarajan, Lu, Yamangil, and Bent, 2016, Nagarajan, Lu, Wang, Bent, and Sundar, 2019] `https://github.com/lanl-ansi/Alpine.jl`

- open-source solver by LANL-ANSI (Los Alamos)
- available for JuMP, implemented in Julia
- at most polynomials
- adaptive, piecewise-linear McCormick convexification scheme

BARON: [Sahinidis, 1996, Tawarmalani and Sahinidis, 2005, Khajavirad and Sahinidis, 2018] `https://minlp.com`

- commercial solver by The Optimization Firm
- available for AIMMS, AMPL, GAMS, JuMP, and more
- spatial branch-and-bound with LP (sometimes also MIP, NLP) relaxation

Solvers for General MINLP

Alpine: [Nagarajan, Lu, Yamangil, and Bent, 2016, Nagarajan, Lu, Wang, Bent, and Sundar, 2019] `https://github.com/lanl-ansi/Alpine.jl`

- open-source solver by LANL-ANSI (Los Alamos)
- available for JuMP, implemented in Julia
- at most polynomials
- adaptive, piecewise-linear McCormick convexification scheme

BARON: [Sahinidis, 1996, Tawarmalani and Sahinidis, 2005, Khajavirad and Sahinidis, 2018] `https://minlp.com`

- commercial solver by The Optimization Firm
- available for AIMMS, AMPL, GAMS, JuMP, and more
- spatial branch-and-bound with LP (sometimes also MIP, NLP) relaxation

EAGO: [Wilhelm and Stuber, 2020] `https://github.com/PSORLab/EAGO.jl`

- open-source solver by Matthew Wilhelm, PSOR Lab at Uni. of Connecticut
- available for JuMP, implemented in Julia
- propagating McCormick relaxations along the factorable structure of each expression (spatial branch-and-bound without auxiliary variables)

Solvers for General MINLP (cont.)

Lindo API: [Lin and Schrage, 2009] `https://www.lindo.com`

- commercial solver by Lindo Systems, Inc.
- available for LINGO and GAMS; APIs for MATLAB, C++, and other
- spatial branch-and-bound with nonlinear relaxations

Solvers for General MINLP (cont.)

Lindo API: [Lin and Schrage, 2009] `https://www.lindo.com`

- commercial solver by Lindo Systems, Inc.
- available for LINGO and GAMS; APIs for MATLAB, C++, and other
- spatial branch-and-bound with nonlinear relaxations

MAiNGO: [Bongartz, Najman, Sass, and Mitsos, 2018]
`https://git.rwth-aachen.de/avt-svt/public/maingo`

- open-source solver by RWTH Aachen, Germany
- C++ and Python APIs
- propagating McCormick relaxations along the factorable structure of each expression (spatial branch-and-bound without auxiliary variables)

Solvers for General MINLP (cont.)

Lindo API: [Lin and Schrage, 2009] `https://www.lindo.com`

- commercial solver by Lindo Systems, Inc.
- available for LINGO and GAMS; APIs for MATLAB, C++, and other
- spatial branch-and-bound with nonlinear relaxations

MAiNGO: [Bongartz, Najman, Sass, and Mitsos, 2018]
`https://git.rwth-aachen.de/avt-svt/public/maingo`

- open-source solver by RWTH Aachen, Germany
- C++ and Python APIs
- propagating McCormick relaxations along the factorable structure of each expression (spatial branch-and-bound without auxiliary variables)

Octeract: `https://octeract.gg`

- commercial solver by Octeract Limited
- available for AIMMS, AMPL, GAMS, JuMP and C++ API
- spatial branch-and-bound with linear relaxation

Solvers for General MINLP (cont.)

SCIP: [Achterberg, 2009, Bestuzheva, Besançon, Chen, Chmiela, Donkiewicz, van Doornmalen, Eifler, Gaul, Gamrath, Gleixner, Gottwald, Graczyk, Halbig, Hoen, Hojny, van der Hulst, Koch, Lübbecke, Maher, Matter, Mühmer, Müller, Pfetsch, Rehfeldt, Schlein, Schlösser, Serrano, Shinano, Sofranac, Turner, Vigerske, Wegscheider, Wellner, Weninger, and Witzig, 2021, Bestuzheva, Chmiela, Müller, Serrano, Vigerske, and Wegscheider, 2023] `https://www.scipopt.org/`

- open-source solver by Zuse Institute Berlin, TU Darmstadt, RWTH Aachen, TU Eindhoven, FAU Erlangen, GAMS, etc
- available for AMPL, GAMS, JuMP, . . . ; APIs for C, Matlab, Python, . . .
- part of a solver for constraint integer programs
- spatial branch-and-bound with linear relaxation

End.

Thank you for your attention!

Some MINLP reviews:

- Burer and Letchford [2012]
- Belotti, Kirches, Leyffer, Linderoth, Luedtke, and Mahajan [2013]
- Boukouvala, Misener, and Floudas [2016]
- Kılınç and Sahinidis [2017]
- Kronqvist, Bernal, Lundell, and Grossmann [2019]

Some books:

- Lee and Leyffer [2012]
- Locatelli and Schoen [2013]

References

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.

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.

Radu Baltean-Lugojan, Pierre Bonami, Ruth Misener, and Andrea Tramontani. Scoring positive semidefinite cutting planes for quadratic optimization via trained neural networks. Technical Report 6943, Optimization Online, 2018. URL `https://optimization-online.org/2018/11/6943/`.

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

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.

Literature ii

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. RENS – the optimal rounding. Mathematical Programming Computation, 6(1):33–54, 2014. 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.

Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, Leona Gottwald, Christoph Graczyk, Katrin Halbig, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Thorsten Koch, Marco Lübbecke, Stephen J. Maher, Frederic Matter, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Daniel Rehfeldt, Steffan Schlein, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Boro Sofranac, Mark Turner, Stefan Vigerske, Fabian Wegscheider, Philipp Wellner, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 8.0. ZIB Report 21-41, Zuse Institute Berlin, 2021. urn:nbn:de:0297-zib-85309.

Ksenia Bestuzheva, Antonia Chmiela, Benjamin Müller, Felipe Serrano, Stefan Vigerske, and Fabian Wegscheider. Global optimization of mixed-integer nonlinear programs with SCIP 8. Technical Report 2301.00587, arXiv, 2023.

D. Bongartz, J. Najman, S. Sass, and A. Mitsos. MAiNGO – McCormick-based algorithm for mixed-integer nonlinear global optimization. Technical report, Process Systems Engineering (AVT.SVT), RWTH Aachen University, 2018. URL `http://permalink.avt.rwth-aachen.de/?id=729717`.

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 S. Vigerske. MINLP solver software. In J. J. Cochran et.al., editor, Wiley Encyclopedia of Operations Research and Management Science. Wiley & Sons, Inc., 2010. doi:10.1002/9780470400531.eorms0527.

Chris Coey, Miles Lubin, and Juan Pablo Vielma. Outer approximation with conic certificates for mixed-integer convex problems. Mathematical Programming Computation, 12(2):249–293, 2020. doi:10.1007/s12532-020-00178-3.

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.

Sourour Elloumi and Amélie Lambert. Global solution of non-convex quadratically constrained quadratic programs. Optimization Methods and Software, 34(1):98–114, 2019. doi:10.1080/10556788.2017.1350675.

Matteo Fischetti, Fred Glover, and Andrea Lodi. The feasibility pump. Mathematical Programming, 104(1):91–104, 2005. doi:10.1007/s10107-004-0570-3.

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.

Antonio Frangioni and Claudio Gentile. Perspective cuts for a class of convex 0–1 mixed integer programs. Mathematical Programming, 106(2):225–236, 2006. doi:10.1007/s10107-005-0594-3.

Björn Geißler, Antonio Morsi, Lars Schewe, and Martin Schmidt. Penalty alternating direction methods for mixed-integer optimization: A new view on feasibility pumps. SIAM Journal on Optimization, 27 (3):1611–1636, 2017. doi:10.1137/16M1069687.

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.

Brais González-Rodríguez, Joaquín Ossorio-Castillo, Julio González-Díaz, Ángel M. González-Rueda, David R. Penas, and Diego Rodríguez-Martínez. Computational advances in polynomial optimization: RAPOSa, a freely available global solver. Journal of Global Optimization, 2022. ISSN 1573-2916. doi:10.1007/s10898-022-01229-w.

Oktay Günlük and Jeff T. Linderoth. Perspective reformulation and applications. In Lee and Leyffer [2012], pages 61–89. doi:10.1007/978-1-4614-1927-3_3.

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.

Aida Khajavirad and Nikolaos V. Sahinidis. A hybrid LP/NLP paradigm for global optimization relaxations. Mathematical Programming Computation, 10(3):383–421, 2018. doi:10.1007/s12532-018-0138-5.

Kamil A. Khan, Harry A. J. Watson, and Paul I. Barton. Differentiable McCormick relaxations. Journal of Global Optimization, 67(4):687–729, 2017. doi:10.1007/s10898-016-0440-6.

Julien Klaus, Niklas Merk, Konstantin Wiedom, Sören Laue, and Joachim Giesen. Convexity certificates from hessians. Technical Report 2210.10430, arXiv, 2022. URL https://arxiv.org/abs/2210.10430.

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.

Ole Kröger, Carleton Coffrin, Hassan Hijazi, and Harsha Nagarajan. Juniper: An open-source nonlinear branch-and-bound solver in Julia. In Integration of Constraint Programming, Artificial Intelligence, and Operations Research, pages 377–386. Springer, 2018. doi:10.1007/978-3-319-93031-2_27.

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.

Jan Kronqvist, David E. Bernal, Andreas Lundell, and Ignacio E. Grossmann. A review and comparison of solvers for convex MINLP. Optimization and Engineering, 20(2):397–455, 2019. doi:10.1007/s11081-018-9411-8.

Mustafa R. Kılınç and Nikolaos V. Sahinidis. Advances and Trends in Optimization with Engineering Applications, chapter State of the Art in Mixed-Integer Nonlinear Optimization, pages 273–292. MOS-SIAM Series on Optimization. 2017. doi:10.1137/1.9781611974683.ch21.

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.

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.

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

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.

Andreas Lundell and Jan Kronqvist. Polyhedral approximation strategies for nonconvex mixed-integer nonlinear programming in SHOT. Journal of Global Optimization, 82(4):863–896, 2022. doi:10.1007/s10898-021-01006-1.

Andreas Lundell, Jan Kronqvist, and Tapio Westerlund. The supporting hyperplane optimization toolkit for convex MINLP. Journal of Global Optimization, 84(1):1–41, 2022. doi:10.1007/s10898-022-01128-0.

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`.

Literature viii

Ashutosh Mahajan, Sven Leyffer, Jeff Linderoth, James Luedtke, and Todd Munson. Minotaur: a mixed-integer nonlinear optimization toolkit. Mathematical Programming Computation, 13(2): 301–338, 2021. doi:10.1007/s12532-020-00196-1.

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.

Wendel Melo, Marcia Fampa, and Fernanda Raupp. An overview of MINLP algorithms and their implementation in Muriqui Optimizer. Annals of Operations Research, 286(1):217–241, 2020. doi:10.1007/s10479-018-2872-5.

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.

Alexander Mitsos, Benoit Chachuat, and Paul I. Barton. McCormick-based relaxations of algorithms. SIAM Journal on Optimization, 20(2):573–601, 2009. doi:10.1137/080717341.

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

Harsha Nagarajan, Mowen Lu, Emre Yamangil, and Russell Bent. Tightening McCormick relaxations for nonlinear programs via dynamic multivariate partitioning. In International Conference on Principles and Practice of Constraint Programming, pages 369–387. Springer, 2016. doi:10.1007/978-3-319-44953-1_24.

Harsha Nagarajan, Mowen Lu, Site Wang, Russell Bent, and Kaarthik Sundar. An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs. Journal of Global Optimization, 2019. doi:10.1007/s10898-018-00734-1.

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.

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

Anureet Saxena, Pierre Bonami, and Jon Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. Mathematical Programming, 130(2): 359–413, 2011. ISSN 0025-5610. doi:10.1007/s10107-010-0340-3.

Felipe Serrano, Robert Schwarz, and Ambros Gleixner. On the relation between the extended supporting hyperplane algorithm and kelley's cutting plane algorithm. Journal of Global Optimization, 78:161 – 179, 2020. doi:10.1007/s10898-020-00906-y.

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.

Hanif D. Sherali and C. H. Tuncbilek. A global optimization algorithm for polynomial programming problems using a reformulation-linearization technique. Journal of Global Optimization, 2:101–112, 1992. doi:10.1007/BF00121304.

Hanif D. Sherali, Evrim Dalkiran, and Jitamitra Desai. Enhancing RLT-based relaxations for polynomial programming problems via a new class of v-semidefinite cuts. Computational Optimization and Applications, 52(2):483–506, 2012. doi:10.1007/s10589-011-9425-z.

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.

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.

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.

Juan Pablo Vielma, Iain Dunning, Joey Huchette, and Miles Lubin. Extended formulations in mixed integer conic quadratic programming. Mathematical Programming Computation, 9(3):369–418, 2016. doi:10.1007/s12532-016-0113-y.

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 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.

M. E. Wilhelm and M. D. Stuber. EAGO.jl: easy advanced global optimization in Julia. Optimization Methods and Software, pages 1–26, 2020. doi:10.1080/10556788.2020.1786566.