# LP/MIP based techniques for solving MINLPs

Stefan Vigerske

Humboldt-Universität zu Berlin

**DFG Research Center MATHEON**
**Mathematics for key technologies**

17th December 2010, Corporate Technology, Siemens AG

# Overview

1. Convex MINLP
   - Algorithms
   - Extensions
   - Solvers
2. Nonconvex MINLP
   - Spatial Branch-and-Bound Algorithm
   - Solvers
   - Application: Mine Production Scheduling
   - Techniques

## Mixed Integer Nonlinear Program (MINLP)

$$
\begin{aligned}
\text{minimize} \quad & f(x) \\
\text{such that} \quad & g_j(x) \leq 0, \qquad j \in J, \\
& x \in X = \{x \in \mathbb{R}^n : Dx \leq d, \underline{x} \leq x \leq \overline{x}\}, \\
& x_I \in \mathbb{Z}^{|I|},
\end{aligned}
$$

where

- $f : X \to \mathbb{R}$ and $g_j : X \to \mathbb{R}$, $j \in J$, are differentiable functions
- $I \subseteq \{1, \ldots, n\}$ the set of integer variables

## Mixed Integer Nonlinear Program (MINLP)

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{such that} \quad & g_j(x) \leq 0, \qquad j \in J, \\ & x \in X = \{x \in \mathbb{R}^n : Dx \leq d, \underline{x} \leq x \leq \overline{x}\}, \\ & x_I \in \mathbb{Z}^{|I|}, \end{aligned}$$

where

- $f : X \to \mathbb{R}$ and $g_j : X \to \mathbb{R}$, $j \in J$, are differentiable functions
- $I \subseteq \{1, \ldots, n\}$ the set of integer variables

Important Special Cases:

- Convex MINLP: $f$ and $g_j$ are convex functions on $X$
- MIQQP (Mixed Integer Quadratically Constraint Quadratic Program): $f$ and $g_j$ are quadratic functions

# Overview: Convex MINLP

1. Convex MINLP
   - Algorithms
   - Extensions
   - Solvers
2. Nonconvex MINLP
   - Spatial Branch-and-Bound Algorithm
   - Solvers
   - Application: Mine Production Scheduling
   - Techniques

Survey papers:

- I.E. Grossmann, Review of Nonlinear Mixed-Integer and Disjunctive Programming Techniques, 2002
- P. Bonami, M. Kilinç, J. Linderoth, Algorithms and Software for Convex Mixed Integer Nonlinear Programs, 2010

# Outline

Some NLP subproblems that will be useful:

## NLP relaxation

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g_j(x) \leq 0,\ j \in J, \\ & x \in X, \\ & x_I \in [\underline{x}_I, \overline{x}_I] \end{aligned}$$

- $\underline{x}_I$ and $\overline{x}_I$ possible restrictions on integer variables (as in B&B)
- solution yields lower bound for MINLP restricted to $[\underline{x}_I^k, \overline{x}_I^k]$

Some NLP subproblems that will be useful:

## NLP relaxation

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g_j(x) \leq 0,\ j \in J, \\ & x \in X, \\ & x_I \in [\underline{x}_I, \overline{x}_I] \end{aligned}$$

- $\underline{x}_I$ and $\overline{x}_I$ possible restrictions on integer variables (as in B&B)
- solution yields lower bound for MINLP restricted to $[\underline{x}_I^k, \overline{x}_I^k]$

## NLP subproblem

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g_j(x) \leq 0,\ j \in J, \\ & x \in X, \\ & x_I = \hat{x}_I \end{aligned}$$

- $\hat{x}_I \in X_I \cap \mathbb{Z}^{|I|}$ a candidate for integer part of a solution
- feasible solution yields upper bound for MINLP

Some NLP subproblems that will be useful:

### NLP relaxation

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g_j(x) \leq 0,\ j \in J, \\ & x \in X, \\ & x_I \in [\underline{x}_I, \overline{x}_I] \end{aligned}$$

- $\underline{x}_I$ and $\overline{x}_I$ possible restrictions on integer variables (as in B&B)
- solution yields lower bound for MINLP restricted to $[\underline{x}_I^k, \overline{x}_I^k]$

### NLP subproblem

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g_j(x) \leq 0,\ j \in J, \\ & x \in X, \\ & x_I = \hat{x}_I \end{aligned}$$

- $\hat{x}_I \in X_I \cap \mathbb{Z}^{|I|}$ a candidate for integer part of a solution
- feasible solution yields upper bound for MINLP

### Feasibility subproblem

$$\begin{aligned} \min\ & u \\ \text{s.t. } & g_j(x) \leq u,\ j \in J, \\ & x \in X, \quad u \in \mathbb{R}, \\ & x_I = \hat{x}_I \end{aligned}$$

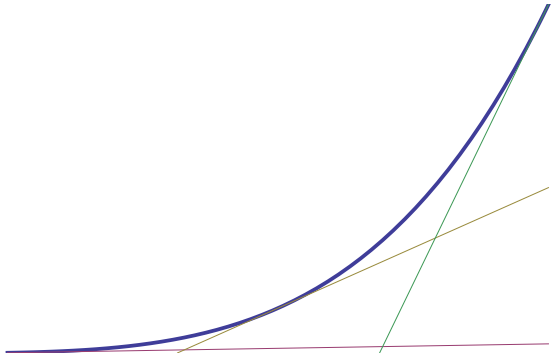
- $\hat{x}_I \in X_I \cap \mathbb{Z}^{|I|}$ a candidate for integer part of a solution
- solution yields minimal infeasible solution w.r.t. $\hat{x}_I$

- replace nonlinearities in MINLP by supporting hyperplanes (use convexity)
- given $K$ points $x^k$, $k = 1, \ldots, K$

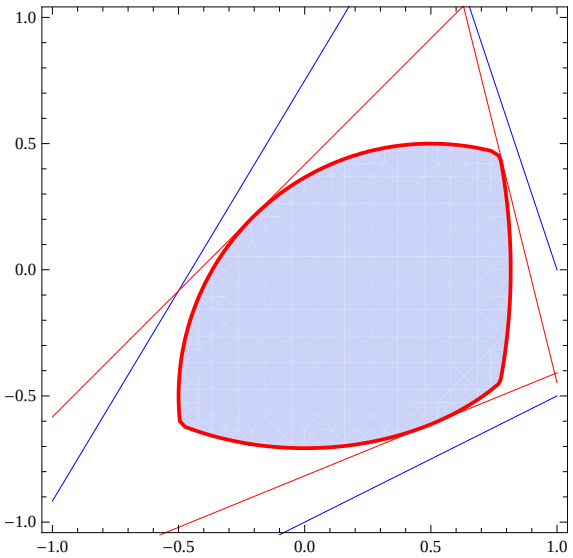
## MIP outer-approximation of MINLP

$$
\begin{aligned}
\min\ & \alpha \\
\text{s.t. } & \alpha \geq f(x^k) + \nabla f(x^k)^T (x - x^k), \qquad k=1,\ldots,K, \\
& g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \leq 0, \qquad j \in J^k,\ k=1,\ldots,K, \\
& x \in X, \quad x_I \in \mathbb{Z}^{|I|}
\end{aligned}
$$

with $J^k \subseteq J$ set of constraints for which to include linearization in $x^k$



underestimate objective




overestimate feasible region

When to solve which NLP subproblem and MIP relaxation?

When to solve which NLP subproblem and MIP relaxation?

Strategies:

- NLP-based Branch-and-Bound [Leyffer, 1993, 2001]
  - solve NLP relaxation to compute lower bound in each node
  - branch on fractional integer variables
  - performs well if NLP relaxation is tight
  - not in the scope of this talk

# Algorithms for Convex MINLP

When to solve which NLP subproblem and MIP relaxation?

Strategies:

- NLP-based Branch-and-Bound [Leyffer, 1993, 2001]
  - solve NLP relaxation to compute lower bound in each node
  - branch on fractional integer variables
  - performs well if NLP relaxation is tight
  - not in the scope of this talk
- Outer-Approximation [Duran and Grossmann, 1986, Yuan et al., 1988, Fletcher and Leyffer, 1994]
- Generalized Benders Decomposition [Geoffrion, 1972]
  - similar to Outer-Approximation
  - good performance only in some special cases (e.g., many discrete variables)
  - not discussed in this talk

# Algorithms for Convex MINLP

When to solve which NLP subproblem and MIP relaxation?

Strategies:

- NLP-based Branch-and-Bound [Leyffer, 1993, 2001]
  - solve NLP relaxation to compute lower bound in each node
  - branch on fractional integer variables
  - performs well if NLP relaxation is tight
  - not in the scope of this talk
- Outer-Approximation [Duran and Grossmann, 1986, Yuan et al., 1988, Fletcher and Leyffer, 1994]
- Generalized Benders Decomposition [Geoffrion, 1972]
  - similar to Outer-Approximation
  - good performance only in some special cases (e.g., many discrete variables)
  - not discussed in this talk
- Extended cutting plane [Westerlund and Petterson, 1995]

# Algorithms for Convex MINLP

When to solve which NLP subproblem and MIP relaxation?

Strategies:

- NLP-based Branch-and-Bound [Leyffer, 1993, 2001]
  - solve NLP relaxation to compute lower bound in each node
  - branch on fractional integer variables
  - performs well if NLP relaxation is tight
  - not in the scope of this talk
- Outer-Approximation [Duran and Grossmann, 1986, Yuan et al., 1988, Fletcher and Leyffer, 1994]
- Generalized Benders Decomposition [Geoffrion, 1972]
  - similar to Outer-Approximation
  - good performance only in some special cases (e.g., many discrete variables)
  - not discussed in this talk
- Extended cutting plane [Westerlund and Petterson, 1995]
- LP/NLP-based Branch-and-Bound [Quesada and Grossmann, 1992, Bonami et al., 2008, Abhishek et al., 2010]

# Outer-Approximation for Convex MINLP

[Duran and Grossmann, 1986, Yuan et al., 1988, Fletcher and Leyffer, 1994]

## Basic Theorem [Duran and Grossmann, 1986]

MINLP and the following problem have the same optimal solution

$$
\begin{aligned}
\min\ & \alpha \\
\text{s.t. } & \alpha \geq f(x^k) + \nabla f(x^k)^T (x - x^k), && \forall\, x_I^k \in X_I \cap \mathbb{Z}^{|I|}, \\
& g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \leq 0, \quad j \in J, && \forall\, x_I^k \in X_I \cap \mathbb{Z}^{|I|}, \\
& x \in X, \quad x_I \in \mathbb{Z}^{|I|}
\end{aligned}
$$

where $x^k$ is the optimal solution for the NLP subproblem in $x_I^k$ or the NLP feasibility problem in $x_I^k$.

# Outer-Approximation for Convex MINLP

[Duran and Grossmann, 1986, Yuan et al., 1988, Fletcher and Leyffer, 1994]

## Basic Theorem [Duran and Grossmann, 1986]

MINLP and the following problem have the same optimal solution

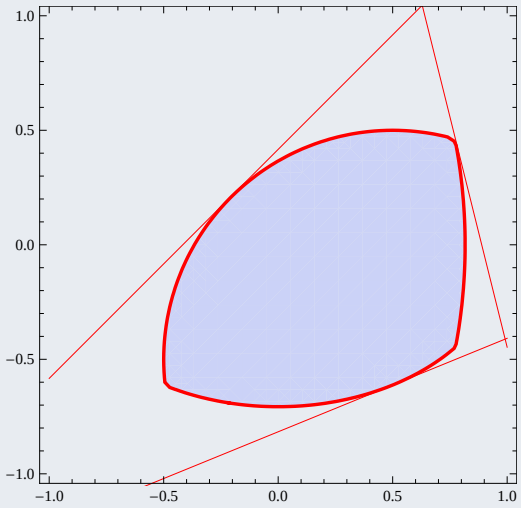
$$
\begin{aligned}
\min\ & \alpha \\
\text{s.t. } & \alpha \geq f(x^k) + \nabla f(x^k)^T (x - x^k), && \forall\, x_I^k \in X_I \cap \mathbb{Z}^{|I|}, \\
& g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \leq 0, \quad j \in J, && \forall\, x_I^k \in X_I \cap \mathbb{Z}^{|I|}, \\
& x \in X, \quad x_I \in \mathbb{Z}^{|I|}
\end{aligned}
$$

where $x^k$ is the optimal solution for the NLP subproblem in $x_I^k$ or the NLP feasibility problem in $x_I^k$.

Idea: relax equivalent MIP to MIP approximation by solving a sequence of NLP sub- and feasibility problems

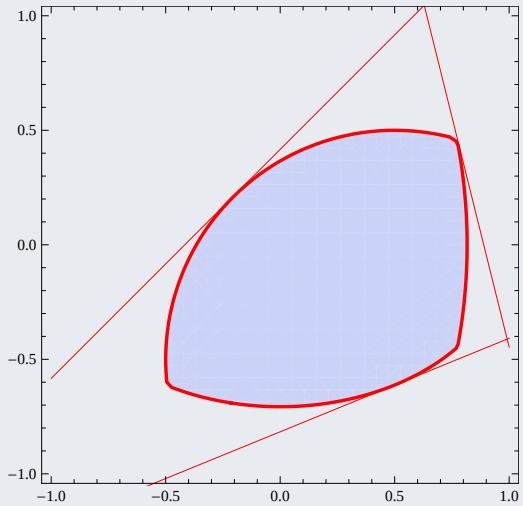
## Outer-Approximation Algorithm [Duran and Grossmann, 1986]

1. solve NLP relaxation
   $x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$
2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$



## Outer-Approximation Algorithm [Duran and Grossmann, 1986]

1. solve NLP relaxation
   $x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$
2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$
3. iterate until lower bound = upper bound or MIP is infeasible:
   3.1 compute lower bound and $x_I^k \in X \cap \mathbb{Z}^{|I|}$ by solving MIP relaxation



## Outer-Approximation Algorithm [Duran and Grossmann, 1986]

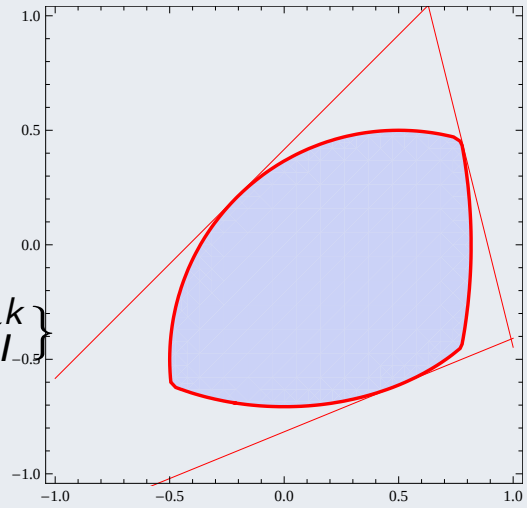
1. solve NLP relaxation
   $x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$
2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$
3. iterate until lower bound = upper bound or MIP is infeasible:
   - 3.1 compute lower bound and $x_I^k \in X \cap \mathbb{Z}^{|I|}$ by solving MIP relaxation
   - 3.2 solve NLP subproblem

     $$x^k := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J,\ \ x \in X,\ \ x_I = x_I^k\}$$

   - 3.3 if feasible, update upper bound
   - 3.4 if infeasible, solve NLP feasibility problem

     $$x^k := \text{argmin}\{u : g_j(x) \leq u,\ j \in J,\ \ x \in X,\ \ u \in \mathbb{R},\ \ x_I = x_I^k\}$$

   - 3.5 add linearizations of $g_j(\cdot)$ in $x^k$ to MIP relaxation



## Outer-Approximation Algorithm [Duran and Grossmann, 1986]

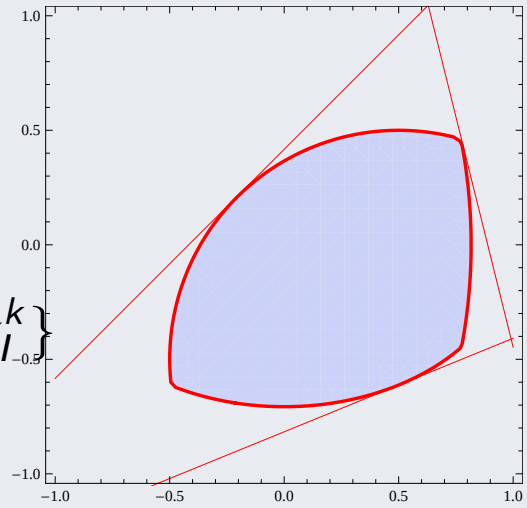
1. solve NLP relaxation
   $x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$
2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$
3. iterate until lower bound = upper bound or MIP is infeasible:
   - 3.1 compute lower bound and $x_I^k \in X \cap \mathbb{Z}^{|I|}$ by solving MIP relaxation (not to optimality in early iterations)
   - 3.2 solve NLP subproblem

     $$x^k := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J,\ \ x \in X,\ \ x_I = x_I^k\}$$

   - 3.3 if feasible, update upper bound
   - 3.4 if infeasible, solve NLP feasibility problem

     $$x^k := \text{argmin}\{u : g_j(x) \leq u,\ j \in J,\ \ x \in X,\ \ u \in \mathbb{R},\ \ x_I = x_I^k\}$$

   - 3.5 add linearizations of $g_j(\cdot)$ in $x^k$ to MIP relaxation



## Outer-Approximation Algorithm [Duran and Grossmann, 1986]

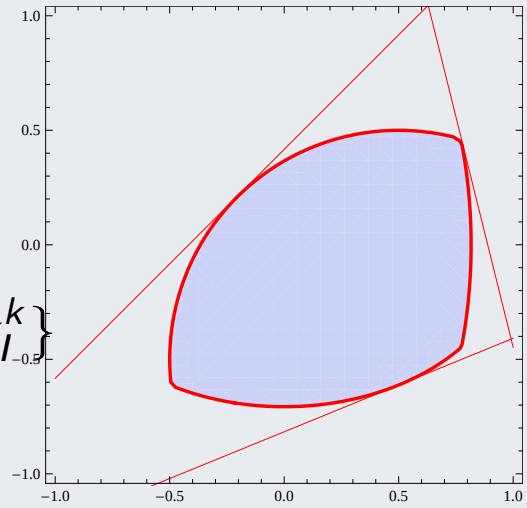
1. solve NLP relaxation
   $x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$
2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$
3. iterate until lower bound $\geq$ upper bound or MIP is infeasible:
   - 3.1 compute lower bound and $x_I^k \in X \cap \mathbb{Z}^{|I|}$ by solving MIP relaxation (not to optimality in early iterations)
   - 3.2 solve NLP subproblem

     $$x^k := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J,\ \ x \in X,\ \ x_I = x_I^k\}$$

   - 3.3 if feasible, update upper bound
   - 3.4 if infeasible, solve NLP feasibility problem

     $$x^k := \text{argmin}\{u : g_j(x) \leq u,\ j \in J,\ \ x \in X,\ \ u \in \mathbb{R},\ \ x_I = x_I^k\}$$

   - 3.5 add linearizations of $g_j(\cdot)$ in $x^k$ to MIP relaxation
   - 3.6 optional: if $X_I \subseteq [0,1]^{|I|}$, add cut to forbid $x_I^k$ in MIP



# Extended Cutting Plane Algorithm

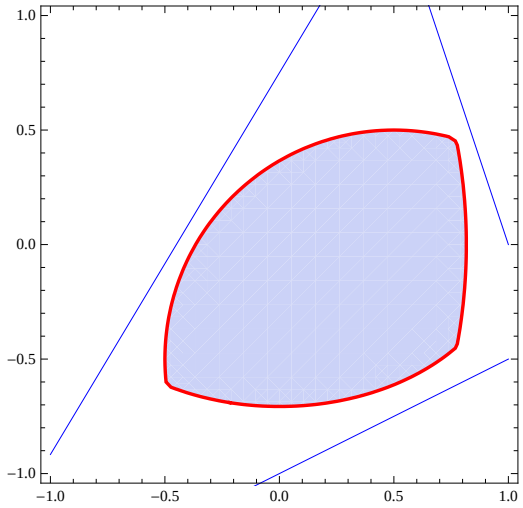
- extension of Kelley's cutting plane method [Kelley, 1961]
- does not rely on solving NLP subproblems

# Extended Cutting Plane Algorithm

- extension of Kelley's cutting plane method [Kelley, 1961]
- does not rely on solving NLP subproblems

## Extended Cutting Plane Algorithm
[Westerlund and Petterson, 1995]

1. initialize MIP outer-approximation by omitting all nonlinear constraints from MINLP



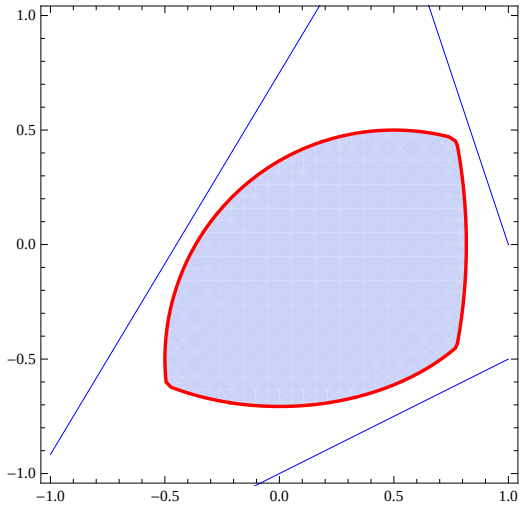
# Extended Cutting Plane Algorithm

- extension of Kelley's cutting plane method [Kelley, 1961]
- does not rely on solving NLP subproblems

## Extended Cutting Plane Algorithm

[Westerlund and Petterson, 1995]

1. initialize MIP outer-approximation by omitting all nonlinear constraints from MINLP
2. iterate:
   - 2.1 solve MIP relaxation to obtain $x^k \in X$ with $x_I^k \in \mathbb{Z}^{|I|}$
   - 2.2 if solution is feasible for MINLP, finish
   - 2.3 add linearization of $g_j(\cdot)$ in $x^k$ to MIP relaxation for most violated nonlinear constraint ($j = \text{argmax}_{j \in J}\{g_j(x^k)\}$)



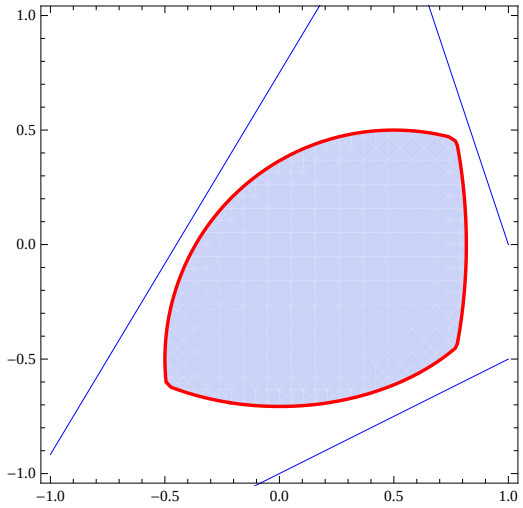
# Extended Cutting Plane Algorithm

- extension of Kelley's cutting plane method [Kelley, 1961]
- does not rely on solving NLP subproblems

## Extended Cutting Plane Algorithm

[Westerlund and Petterson, 1995]

1. initialize MIP outer-approximation by omitting all nonlinear constraints from MINLP
2. iterate:
   - 2.1 solve MIP relaxation to obtain $x^k \in X$ with $x_I^k \in \mathbb{Z}^{|I|}$
   - 2.2 if solution is feasible for MINLP, finish
   - 2.3 add linearization of $g_j(\cdot)$ in $x^k$ to MIP relaxation for most violated nonlinear constraint ($j = \text{argmax}_{j \in J}\{g_j(x^k)\}$)
     alternatively: add linearization for all or all violated constraints



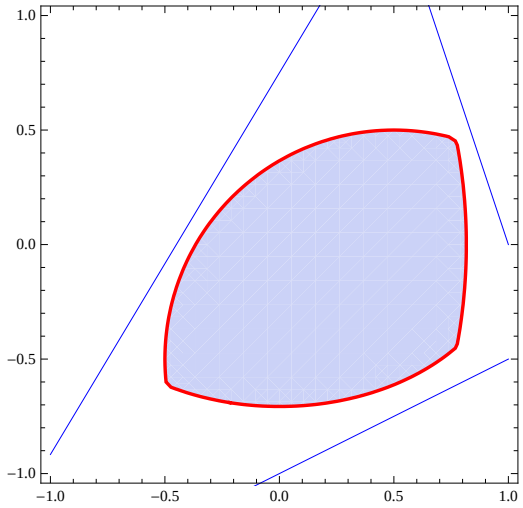
# Extended Cutting Plane Algorithm

- extension of Kelley's cutting plane method [Kelley, 1961]
- does not rely on solving NLP subproblems

## Extended Cutting Plane Algorithm

[Westerlund and Petterson, 1995]

1. initialize MIP outer-approximation by omitting all nonlinear constraints from MINLP
2. iterate:
   - 2.1 solve MIP relaxation to obtain $x^k \in X$ with $x_I^k \in \mathbb{Z}^{|I|}$
   - 2.2 if solution is feasible for MINLP, finish
   - 2.3 add linearization of $g_j(\cdot)$ in $x^k$ to MIP relaxation for most violated nonlinear constraint ($j = \text{argmax}_{j \in J}\{g_j(x^k)\}$)
     alternatively: add linearization for all or all violated constraints




- works well on almost linear problems

# LP/NLP based Branch-and-Bound

integrate Outer-Approximation into MIP alike branch-and-bound

## LP/NLP based Branch-and-Bound Alg. [Quesada and Grossmann, 1992]

1. solve NLP relaxation

$$x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$$

2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$

# LP/NLP based Branch-and-Bound

integrate Outer-Approximation into MIP alike branch-and-bound

## LP/NLP based Branch-and-Bound Alg. [Quesada and Grossmann, 1992]

1. solve NLP relaxation

$$x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$$

2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$
3. perform branch-and-bound:
   - 3.1 compute lower bound and $\tilde{x}^k$ by solving LP relaxation of current node
   - 3.2 if $\tilde{x}^k$ is feasible for MINLP, update upper bound, go to 3.1
   - 3.3 if $\tilde{x}^k_I$ is fractional, branch on some variable $x_i$, $i \in I$, go to 3.1

integrate Outer-Approximation into MIP alike branch-and-bound

## LP/NLP based Branch-and-Bound Alg. [Quesada and Grossmann, 1992]

1. solve NLP relaxation

$$x^1 := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X\}$$

2. initialize MIP outer-approximation with linearizations of $g_j(\cdot)$ in $x^1$
3. perform branch-and-bound:
   - 3.1 compute lower bound and $\tilde{x}^k$ by solving LP relaxation of current node
   - 3.2 if $\tilde{x}^k$ is feasible for MINLP, update upper bound, go to 3.1
   - 3.3 if $\tilde{x}^k_I$ is fractional, branch on some variable $x_i$, $i \in I$, go to 3.1
   - 3.4 solve NLP subproblem or NLP feasibility problem

$$x^k := \text{argmin}\{f(x) : g_j(x) \leq 0,\ j \in J, \quad x \in X, \quad x_I = \tilde{x}^k_I\}$$

$$\text{or} \quad x^k := \text{argmin}\{u : g_j(x) \leq u,\ j \in J,\ \ x \in X,\ \ u \in \mathbb{R},\ \ x_I = \tilde{x}^k_I\}$$

   - 3.5 add linearizations of $g_j(\cdot)$ in $x^k$ to LP relaxation

# Outline

# Handling Equalities

- assume presence of nonlinear equalities

$$h_j(x) = 0, \quad j \in J_=$$

$\Rightarrow$ generally introduce nonconvexities

# Handling Equalities

- assume presence of nonlinear equalities

$$h_j(x) = 0, \quad j \in J_=$$

$\Rightarrow$ generally introduce nonconvexities

- proposed relaxation [Kocis and Grossmann, 1987]:

$$\begin{aligned} \nabla h_j(x^k)^T(x - x^k) &\leq 0, \qquad \text{if } \lambda_j^k > 0, \\ \nabla h_j(x^k)^T(x - x^k) &\geq 0, \qquad \text{if } \lambda_j^k < 0, \end{aligned}$$

  where $\lambda_j^k$ is the dual multiplier associated with $h_j(x, y) = 0$ in the NLP, $j \in J_=$

- remains a rigorous method only in special cases

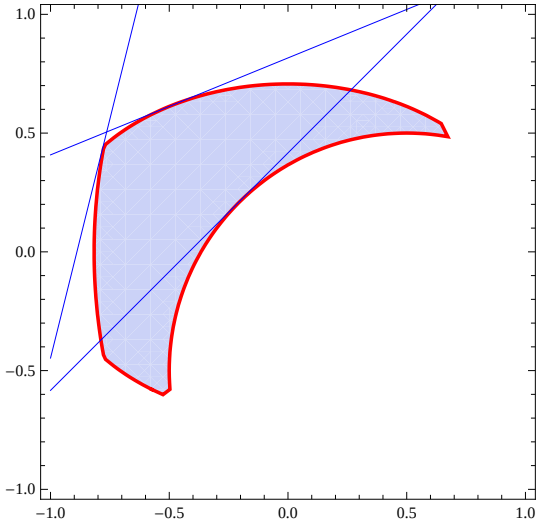
# Reducing Effect of Nonconvexity

Assume $f(\cdot)$ or $g_j(\cdot)$ nonconvex or nonlinear equalities $h(x, y) = 0$ present.

$\Rightarrow$ Outer-Approximation runs in two difficulties:

1. NLP subproblems may not be solved to global optimality
2. gradient-based linearizations may cut into feasible region

$\Rightarrow$ apply heuristic strategies to reduce effect of nonconvexity



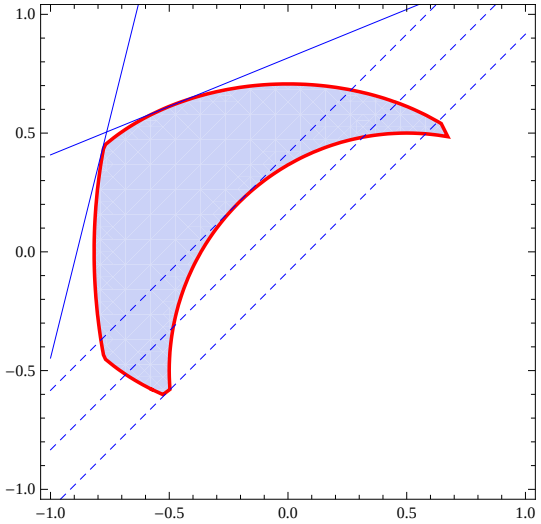
# Reducing Effect of Nonconvexity

Assume $f(\cdot)$ or $g_j(\cdot)$ nonconvex or nonlinear equalities $h(x, y) = 0$ present.

$\Rightarrow$ Outer-Approximation runs in two difficulties:

1. NLP subproblems may not be solved to global optimality
2. gradient-based linearizations may cut into feasible region

$\Rightarrow$ apply heuristic strategies to reduce effect of nonconvexity




## Augmented Penalty / Equality Relaxation MIP master problem

[Viswanathan and Grossmann, 1990]

$$
\begin{aligned}
\min\ & \alpha + \sum_{k=1}^{K} w_p^k p^k + w_q^k q^k \\
\text{s.t. } & \alpha \geq f(x^k) + \nabla f(x^k)^T (x - x^k), && k = 1, \ldots, K, \\
& g_j(x^k) + \nabla g_j(x^k)^T (x - x^k) \leq p^k, && j \in J, k = 1, \ldots, K, \\
& \text{sign}(\lambda_k) \nabla h_j(x^k)^T (x - x^k) \leq q^k, && j \in J_=, k = 1, \ldots, K, \\
& x \in X, \quad x_I \in \mathbb{Z}^{|I|}, \quad \alpha \in \mathbb{R}, \quad p^k, q^k \geq 0
\end{aligned}
$$

# Branching and Node Selection Strategies

Consider LP/NLP based Branch-and-Bound.

Similar Strategies as in MIP:

- pseudo-cost branching
- strong branching
- reliability branching

- depth first search
- best first search
- diving methods
- two-phase methods

# Cutting Planes

Consider LP/NLP based Branch-and-Bound.

Cutting Planes for convex MINLP:

- ▷ MIP cutting planes (Gomory, MIR, flowcover, ...)

# Cutting Planes

Consider LP/NLP based Branch-and-Bound.

Cutting Planes for convex MINLP:

- MIP cutting planes (Gomory, MIR, flowcover, ...)
- Gomory Cuts for conic constraints [Cezik and Iyengar, 2005]
- Mixed Integer Rounding for Second-Order-Cone Constraints: [Atamtürk and Narayanan, 2010]

Consider LP/NLP based Branch-and-Bound.

Cutting Planes for convex MINLP:

- MIP cutting planes (Gomory, MIR, flowcover, ...)
- Gomory Cuts for conic constraints [Cezik and Iyengar, 2005]
- Mixed Integer Rounding for Second-Order-Cone Constraints: [Atamtürk and Narayanan, 2010]
- disjunctive inequalities / lift-and-project: [Stubbs and Mehrotra, 1999, Drewes, 2009, Kilinç et al., 2010a]
- simple lift-and-project inequality from (strong) branching: [Kilinç et al., 2010b]

$$\alpha \geq \hat{\alpha}_i^- + (\hat{\alpha}_i^+ - \hat{\alpha}_i^-)x_i,$$

where $\hat{\alpha}_i^-$ = lower bound for $x_i = 0$; $\hat{\alpha}_i^+$ = lower bound for $x_i = 1$
20% improvement in mean solve time for FilMINT [Bonami et al., 2010]

Heuristics for MINLP:

- diving heuristics for NLP-based branch-and-bound [Bonami and Gonçalves, 2008]
  - as in MIP, but try to fix several variables at each iteration
  - dive such that all nonlinear integer variables get integral values, then fix all nonlinear variables (to current value in NLP relaxation), solve remaining MIP
  - 21% improvement in mean B&B time

Heuristics for MINLP:

- diving heuristics for NLP-based branch-and-bound [Bonami and Gonçalves, 2008]
  - as in MIP, but try to fix several variables at each iteration
  - dive such that all nonlinear integer variables get integral values, then fix all nonlinear variables (to current value in NLP relaxation), solve remaining MIP
  - 21% improvement in mean B&B time
- large neighborhood search (RENS, RINS, DINS, local branching, ...) [Berthold et al., 2009a, Nannicini et al., 2009, Liberti et al., 2009]

Heuristics for MINLP:

- diving heuristics for NLP-based branch-and-bound [Bonami and Gonçalves, 2008]
  - as in MIP, but try to fix several variables at each iteration
  - dive such that all nonlinear integer variables get integral values, then fix all nonlinear variables (to current value in NLP relaxation), solve remaining MIP
  - 21% improvement in mean B&B time
- large neighborhood search (RENS, RINS, DINS, local branching, ...) [Berthold et al., 2009a, Nannicini et al., 2009, Liberti et al., 2009]
- sub-MIP heuristic "Undercover" (later...) [Berthold and Gleixner, 2009]

Heuristics for MINLP:

- diving heuristics for NLP-based branch-and-bound [Bonami and Gonçalves, 2008]
  - as in MIP, but try to fix several variables at each iteration
  - dive such that all nonlinear integer variables get integral values, then fix all nonlinear variables (to current value in NLP relaxation), solve remaining MIP
  - 21% improvement in mean B&B time
- large neighborhood search (RENS, RINS, DINS, local branching, ...) [Berthold et al., 2009a, Nannicini et al., 2009, Liberti et al., 2009]
- sub-MIP heuristic "Undercover" (later...) [Berthold and Gleixner, 2009]
- Feasibility Pump [Bonami et al., 2009a, Bonami and Gonçalves, 2008]

# The Feasibility Pump for MINLP

[Bonami et al., 2009a, Bonami and Gonçalves, 2008]

Basic Idea: Create a sequence of integer feasible points $\hat{x}^k$ and nonlinear feasible points $x^k$ by alternating solution of

## MIP relaxation

$$\begin{aligned} \hat{x}^{K+1} := \text{argmin } & \|x - x^K\|_1 \\ \text{s.t. } & g_j(x^k) + \nabla g_j(x^k)^T(x - \bar{x}^k) \leq 0, \quad j \in J, \quad k \leq K \\ & x \in X, \qquad x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

## NLP relaxation

$$\begin{aligned} x^{K+1} := \text{argmin } & \|x - \hat{x}^K\|_2 \\ \text{s.t. } & g_j(x) \leq 0, \quad j \in J \\ & x \in X \end{aligned}$$

# The Feasibility Pump for MINLP

[Bonami et al., 2009a, Bonami and Gonçalves, 2008]

Basic Idea: Create a sequence of integer feasible points $\hat{x}^k$ and nonlinear feasible points $x^k$ by alternating solution of

## MIP relaxation

$$\begin{aligned} \hat{x}^{K+1} := \text{argmin } & \|x - x^K\|_1 \\ \text{s.t. } & g_j(x^k) + \nabla g_j(x^k)^T(x - \bar{x}^k) \leq 0, \quad j \in J, \quad k \leq K \\ & x \in X, \qquad x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

## NLP relaxation

$$\begin{aligned} x^{K+1} := \text{argmin } & \|x - \hat{x}^K\|_2 \\ \text{s.t. } & g_j(x) \leq 0, \quad j \in J \\ & x \in X \end{aligned}$$

- Convergence for convex problems: finds feasible point $x^K$ or proofs infeasibility
- Heuristic for nonconvex problems

# The Feasibility Pump for MINLP

[Bonami et al., 2009a, Bonami and Gonçalves, 2008]

Basic Idea: Create a sequence of integer feasible points $\hat{x}^k$ and nonlinear feasible points $x^k$ by alternating solution of

## Alternative: Rounding

$$\hat{x}^{K+1} := \text{argmin } \|x - x^K\|_1$$

s.t.

$$x_I \in \mathbb{Z}^{|I|}$$

## NLP relaxation

$$x^{K+1} := \text{argmin } \|x - \hat{x}^K\|_2$$

s.t. $g_j(x) \leq 0, \quad j \in J$

$$x \in X$$

- Convergence for convex problems: finds feasible point $x^K$ or proofs infeasibility
- Heuristic for nonconvex problems
- Rounding variant finds solution for 93% of the instances;
  2.7% improvement in mean branch-and-bound time
  [Bonami and Gonçalves, 2008]

Perspective Relaxation: [Günlük et al., 2007, Hijazi et al., 2009, Günlük and Linderoth, 2010]
Disjunctive Programming: [Grossmann and Lee, 2003, Vecchietti and Grossmann, 1999]

# Outline

# Solvers that implement NLP based Branch–and–Bound

**SBB = Simple Branch and Bound** [GAMS Development Corp., 2010]

- developed by A. Drud (ARKI Consulting)
- available within GAMS only

# Solvers that implement NLP based Branch–and–Bound

## SBB = Simple Branch and Bound [GAMS Development Corp., 2010]

- developed by A. Drud (ARKI Consulting)
- available within GAMS only

## Bonmin = Basic Open-source Nonlinear Mixed Integer Programming

[Bonami et al., 2008]

- COIN-OR solver developed by P. Bonami, A. Wächter, et.al.
- uses CBC, CPLEX for MIP and IPOPT, FilterSQP for NLP
- option B-BB = NLP-based Branch-and-Bound
- several heuristics, branching and node selection techniques

# Solvers that implement NLP based Branch–and–Bound

## SBB = Simple Branch and Bound [GAMS Development Corp., 2010]

- developed by A. Drud (ARKI Consulting)
- available within GAMS only

## Bonmin = Basic Open-source Nonlinear Mixed Integer Programming

[Bonami et al., 2008]

- COIN-OR solver developed by P. Bonami, A. Wächter, et.al.
- uses CBC, CPLEX for MIP and IPOPT, FilterSQP for NLP
- option B-BB = NLP-based Branch-and-Bound
- several heuristics, branching and node selection techniques

## MINLP_BB = Mixed Integer NonLinear Programming Branch&Bound

[Leyffer, 2001]

- developed by S. Leyffer
- based on FilterSQP and bqpd
- advanced (strong) branching and node selection techniques

# Solvers that implement Outer Approximation Algorithm

## DICOPT = Discrete and Continuous Optimizer

[Viswanathan and Grossmann, 1990]

- developed by J. Viswanathan and I.E. Grossmann (CMU)
- available within GAMS only
- implements augmented penalty / equation relaxation approach to cope with nonconvexities and equalities

# Solvers that implement Outer Approximation Algorithm

## DICOPT = Discrete and Continuous Optimizer

[Viswanathan and Grossmann, 1990]

- developed by J. Viswanathan and I.E. Grossmann (CMU)
- available within GAMS only
- implements augmented penalty / equation relaxation approach to cope with nonconvexities and equalities

## AOA = AIMMS Outer Approximation [Roelofs and Bisschop, 2009]

- developed by Paragon Decision Technology
- available as "open solver" inside AIMMS
- augmented penalty approach to cope with nonconvexities

# Solvers that implement Outer Approximation Algorithm

## DICOPT = Discrete and Continuous Optimizer

[Viswanathan and Grossmann, 1990]

- developed by J. Viswanathan and I.E. Grossmann (CMU)
- available within GAMS only
- implements augmented penalty / equation relaxation approach to cope with nonconvexities and equalities

## AOA = AIMMS Outer Approximation [Roelofs and Bisschop, 2009]

- developed by Paragon Decision Technology
- available as "open solver" inside AIMMS
- augmented penalty approach to cope with nonconvexities

## Bonmin = Basic Open-source Nonlinear Mixed Integer Programming

- COIN-OR solver developed by P. Bonami, A. Wächter, et.al.
- option B-OA = Outer Approximation Algorithm

# Solvers implementing Extended Cutting Plane Alg.

## AlphaECP = $\alpha$ Extended Cutting Plane [Westerlund and Pörn, 2002]

- developed by T. Westerlund and T. Lastusilta at Åbo Akademi University, Finland
- may also solve NLP subproblems occasionally

# Solvers implementing Extended Cutting Plane Alg.

## AlphaECP = $\alpha$ Extended Cutting Plane [Westerlund and Pörn, 2002]

- developed by T. Westerlund and T. Lastusilta at Åbo Akademi University, Finland
- may also solve NLP subproblems occasionally

## FICO Xpress-SLP = Sequential Linear Programming [FIC, 2008]

- developed by Dash Optimization (acquired by FICO)
- option "MIP within SLP" = solve MIPs as subproblems in SLP algorithm

# Solvers implementing Extended Cutting Plane Alg.

## AlphaECP = $\alpha$ Extended Cutting Plane [Westerlund and Pörn, 2002]

- developed by T. Westerlund and T. Lastusilta at Åbo Akademi University, Finland
- may also solve NLP subproblems occasionally

## FICO Xpress-SLP = Sequential Linear Programming [FIC, 2008]

- developed by Dash Optimization (acquired by FICO)
- option "MIP within SLP" = solve MIPs as subproblems in SLP algorithm

## Bonmin = Basic Open-source Nonlinear Mixed Integer Programming

- COIN-OR solver developed by P. Bonami, A. Wächter, et.al.
- option B-ECP = Extended Cutting Plane
- experimental

# Solvers implementing LP/NLP-based Branch&Bound

**FilMINT = Filter-Mixed INTeger optimizer** [Abhishek et al., 2010]

- developed by K. Abhishek, J.T. Linderoth (Lehigh), and S. Leyffer (Argonne)
- combines MINTO (Nemhauser et.al.) with filterSQP (Fletcher)
- includes feasibility pump and strong-branching disjunctive cuts


# Solvers implementing LP/NLP-based Branch&Bound

## FilMINT = Filter-Mixed INTeger optimizer [Abhishek et al., 2010]

- developed by K. Abhishek, J.T. Linderoth (Lehigh), and S. Leyffer (Argonne)
- combines MINTO (Nemhauser et.al.) with filterSQP (Fletcher)
- includes feasibility pump and strong-branching disjunctive cuts


## Bonmin = Basic Open-source Nonlinear Mixed Integer Programming

- COIN-OR solver developed by P. Bonami, A. Wächter, et.al.
- option B-QG = Quesada-Grossmann LP/NLP-based Branch-and-Bound
- option B-Hyb = alternate between LP and NLP relaxation in nodes
- implemented as extension of CBC

# Solvers implementing LP/NLP-based Branch&Bound

## KNITRO

[Byrd et al., 2006]

- developed by Ziena Optimization, Inc.
- can use both LPs and NLPs for bounding

# Solvers implementing LP/NLP-based Branch&Bound

## KNITRO [Byrd et al., 2006]

- developed by Ziena Optimization, Inc.
- can use both LPs and NLPs for bounding

## FICO Xpress-SLP = Sequential Linear Programming [FIC, 2008]

- developed by Dash Optimization (acquired by FICO)
- option "SLP within MIP" = solve NLPs as subproblems in MIP B&B

# Solvers implementing LP/NLP-based Branch&Bound

## KNITRO [Byrd et al., 2006]

- developed by Ziena Optimization, Inc.
- can use both LPs and NLPs for bounding

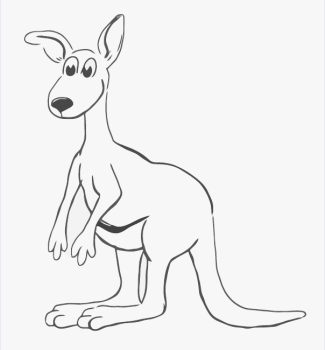
## FICO Xpress-SLP = Sequential Linear Programming [FIC, 2008]

- developed by Dash Optimization (acquired by FICO)
- option "SLP within MIP" = solve NLPs as subproblems in MIP B&B

## SCIP = Solving Constraint Integer Programs

[Achterberg, 2007, Berthold et al., 2009a]

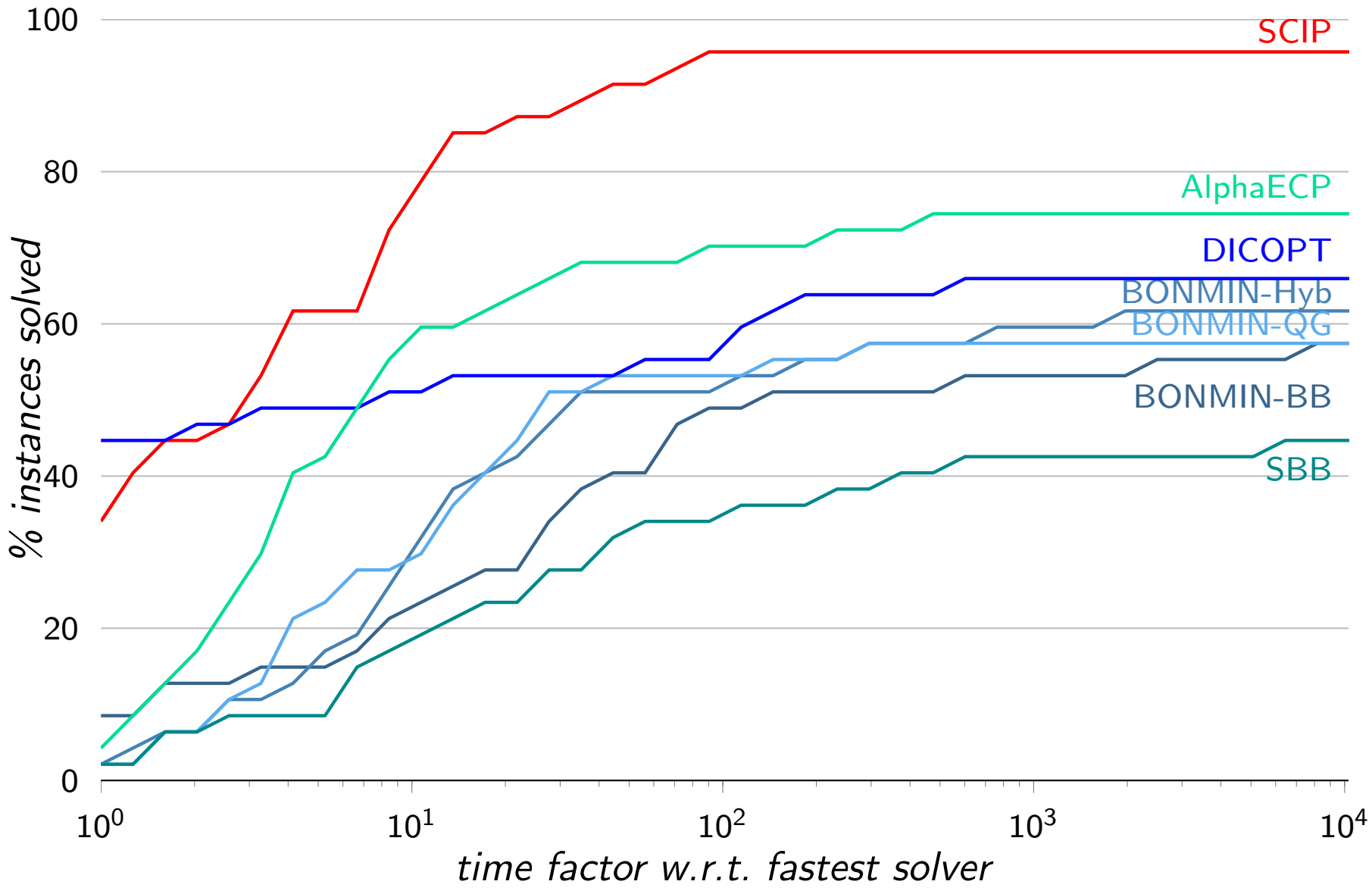
- since 1.2.0: support for MIQQP
- general MINLP in development
- close to extended cutting plane



Experimental Environment:

- test set from [Bonami et al., 2009b]: 48 convex MINLP instances from various applications
- solvers:
  - AlphaECP 1.75.04
    CPLEX for MIPs and CONOPT for NLPs
    option `ECPstrategy 1`
  - BONMIN 1.3 in variants B-BB, B-QG, and B-Hyb
    CBC for MIPs and IPOPT for NLPs
  - DICOPT 2x-C
    CPLEX for MIPs and CONOPT for NLPs
    options `stop 1`, `maxcycles 10000`
  - SBB
    CONOPT for NLPs
    option `memnodes 9999999`
  - SCIP 1.2.0.7 with experimental support for convex nonlinear constraints
    CPLEX for LPs and IPOPT for NLPs
- timelimit 1 hour, gap tolerance $10^{-4}$
- 2.5GHz Intel Core2 Duo, 4GB RAM

# Benchmark on 48 convex MINLPs



# Overview: Nonconvex MINLP

1. Convex MINLP
   - Algorithms
   - Extensions
   - Solvers
2. Nonconvex MINLP
   - Spatial Branch-and-Bound Algorithm
   - Solvers
   - Application: Mine Production Scheduling
   - Techniques

Some key papers:

- M. Tawarmalani and N. Sahinidis, Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications, 2002
- P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, Branching and bounds tightening techniques for non-convex MINLP, 2009

# Outline

# How to solve Nonconvex MINLPs?

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{such that} \quad & g_j(x) \leq 0, \qquad j \in J \\ & x \in X, \\ & x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

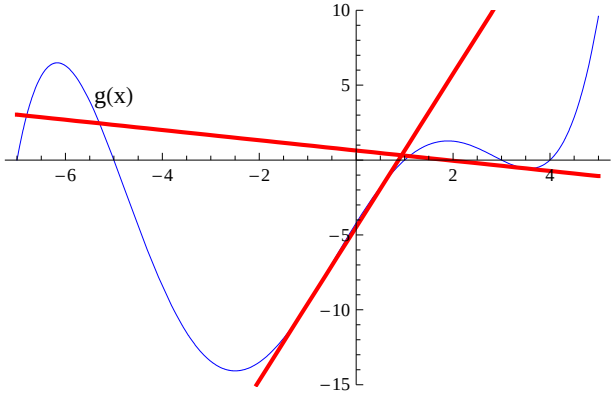
Now: some $g_j : \mathbb{R}^n \to \mathbb{R}$ may be nonconvex

# How to solve Nonconvex MINLPs?

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{such that} \quad & g_j(x) \leq 0, \qquad j \in J \\ & x \in X, \\ & x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

Now: some $g_j : \mathbb{R}^n \to \mathbb{R}$ may be nonconvex

$\Rightarrow$ inequalities $g_j(\hat{x}) + \nabla g_j(\hat{x})^T (x - \hat{x}) \leq 0$ may not be valid!



# How to solve Nonconvex MINLPs?

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{such that} \quad & g_j(x) \leq 0, \qquad j \in J \\ & x \in X, \\ & x_I \in \mathbb{Z}^{|I|} \end{aligned}$$

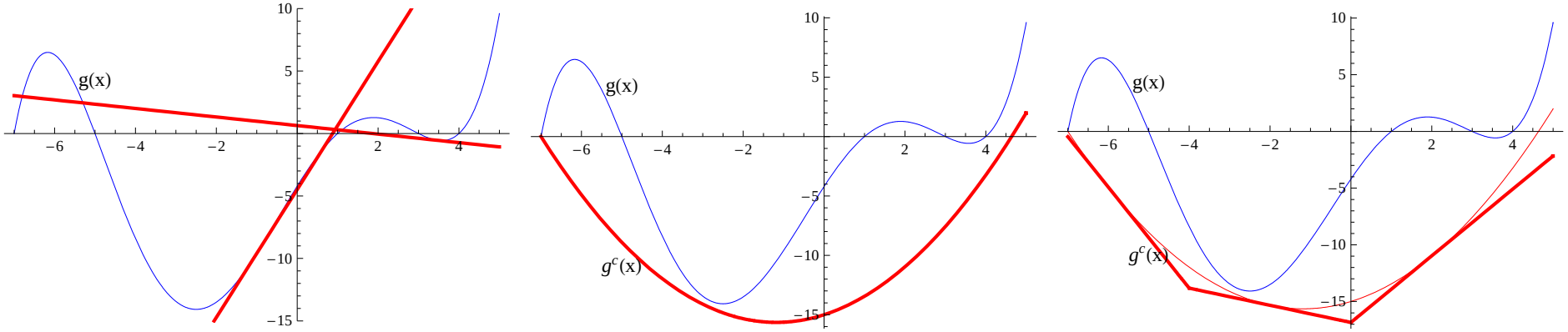
Now: some $g_j : \mathbb{R}^n \to \mathbb{R}$ may be nonconvex

$\Rightarrow$ inequalities $g_j(\hat{x}) + \nabla g_j(\hat{x})^T (x - \hat{x}) \leq 0$ may not be valid!

$\Rightarrow$ use techniques from global optimization of NLPs

$\Rightarrow$ use convex underestimator: convex and below $g(x)$ for all $x \in X$

$\Rightarrow$ introduces convexification gap



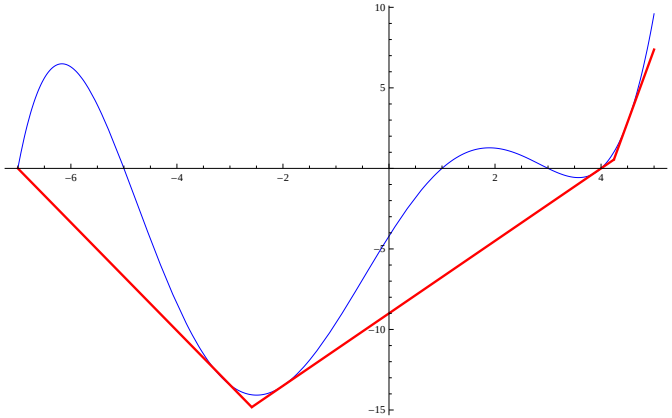
# Spatial branch-and-bound Algorithm

[Smith and Pantelides, 1999, Tawarmalani and Sahinidis, 2002]

In each node with local bounds $[\underline{x}, \overline{x}]$:

Bounding: generate a linear outer-approximation w.r.t. $[\underline{x}, \overline{x}]$ using convexification and linearization techniques

$$\begin{aligned} \min \quad & \alpha \\ \text{s.t.} \quad & Ax + A'\alpha \leq b, \\ & x \in X, \quad x \in [\underline{x}, \overline{x}] \end{aligned}$$



# Spatial branch-and-bound Algorithm

[Smith and Pantelides, 1999, Tawarmalani and Sahinidis, 2002]

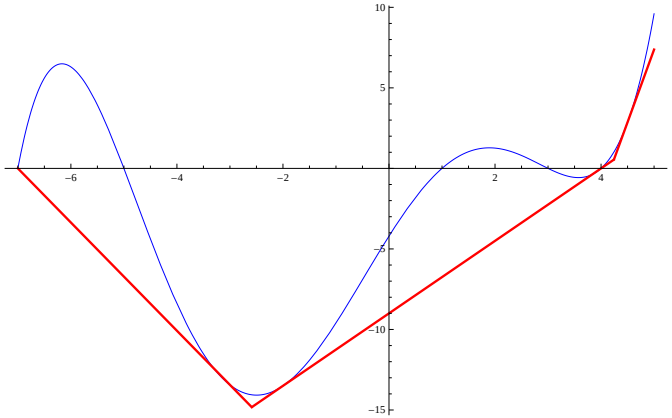
In each node with local bounds $[\underline{x}, \overline{x}]$:

Bounding: generate a linear outer-approximation w.r.t. $[\underline{x}, \overline{x}]$ using convexification and linearization techniques

$$\begin{aligned} \min \quad & \alpha \\ \text{s.t.} \quad & Ax + A'\alpha \leq b, \\ & x \in X, \quad x \in [\underline{x}, \overline{x}] \end{aligned}$$

Branching: close gap between relaxation and problem

- ▷ branch on integer variables with fractional value in LP



# Spatial branch-and-bound Algorithm

[Smith and Pantelides, 1999, Tawarmalani and Sahinidis, 2002]

In each node with local bounds $[\underline{x}, \overline{x}]$:

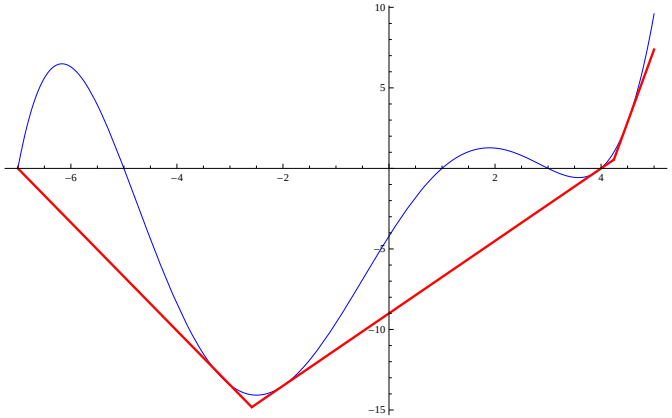
Bounding: generate a linear outer-approximation w.r.t. $[\underline{x}, \overline{x}]$ using convexification and linearization techniques

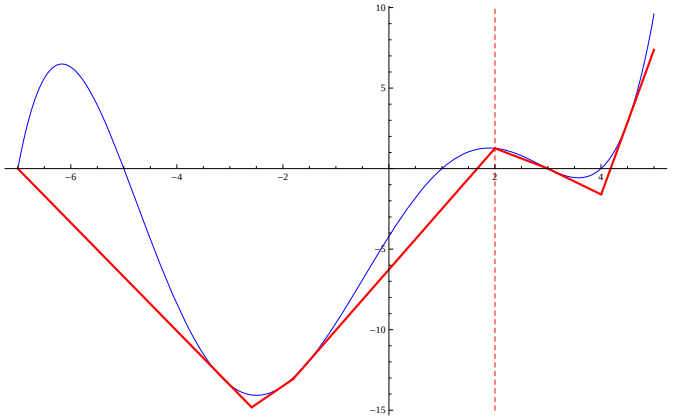
$$\begin{aligned} \min \quad & \alpha \\ \text{s.t.} \quad & Ax + A'\alpha \leq b, \\ & x \in X, \quad x \in [\underline{x}, \overline{x}] \end{aligned}$$

Branching: close gap between relaxation and problem

- ▷ branch on integer variables with fractional value in LP
- ▷ branch on continuous variables in nonconvex terms

  $\Rightarrow$ smaller domain $\Rightarrow$ tighter relaxation





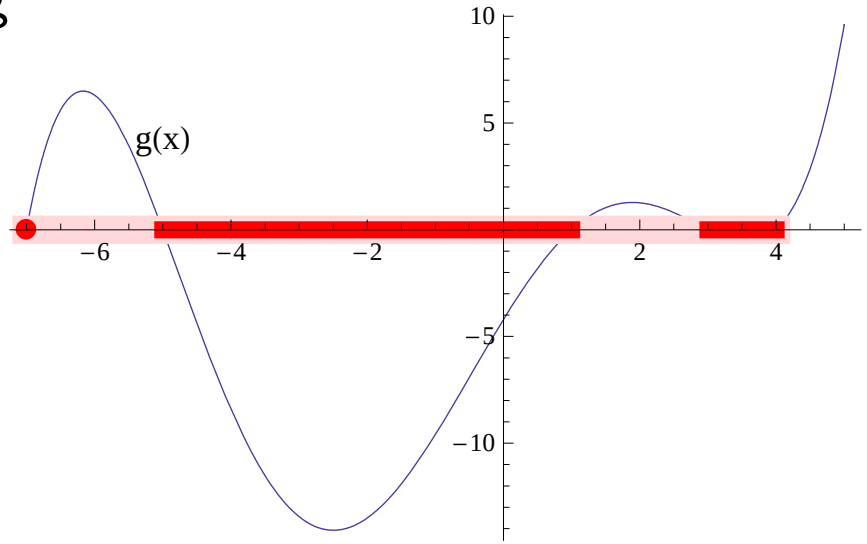
# The linear outer-approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [\underline{x}, \overline{x}]$$

Seek: A (linear) outer-approximation of

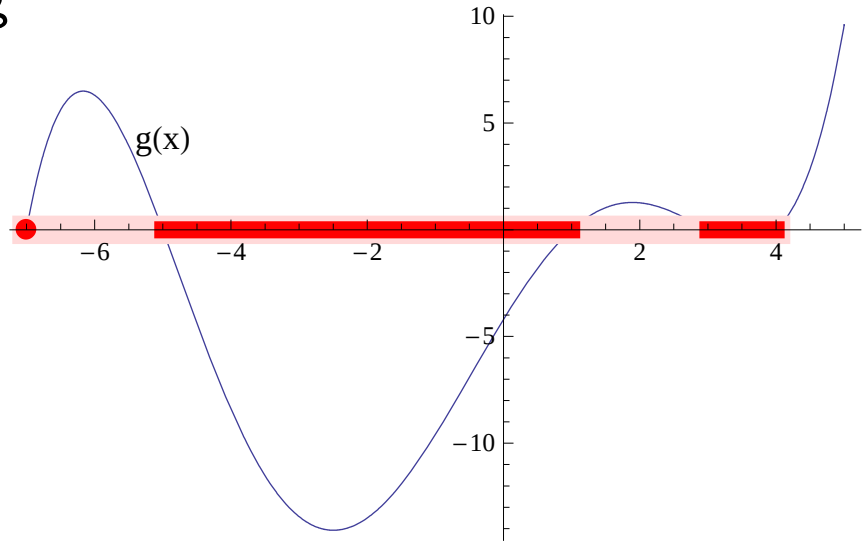
$$\mathrm{conv}\{x \in [\underline{x}, \overline{x}] : g(x) \leq 0\}$$



# The linear outer-approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [\underline{x}, \overline{x}]$$




Seek: A (linear) outer-approximation of

$$\mathrm{conv}\{x \in [\underline{x}, \overline{x}] : g(x) \leq 0\}$$

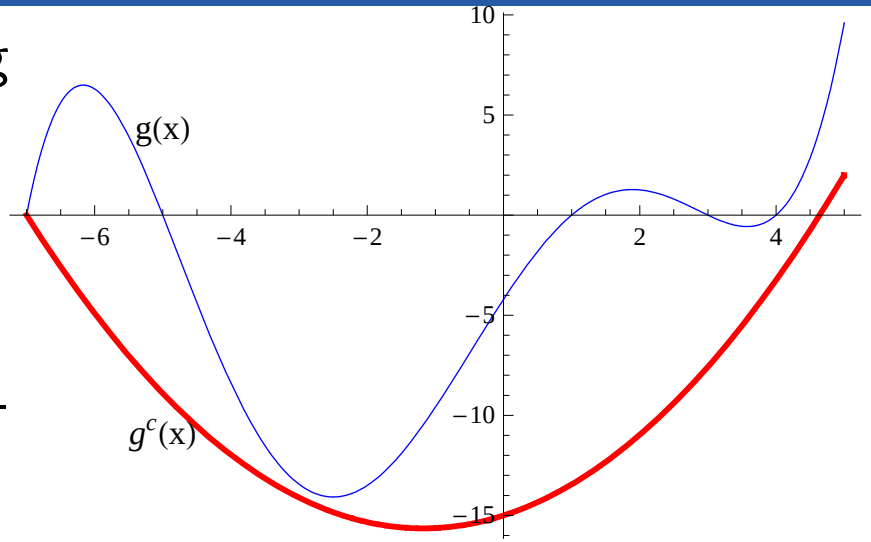
$\Rightarrow$ too difficult in general

# The linear outer-approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [\underline{x}, \overline{x}]$$

Seek: A convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[\underline{x}, \overline{x}]$:



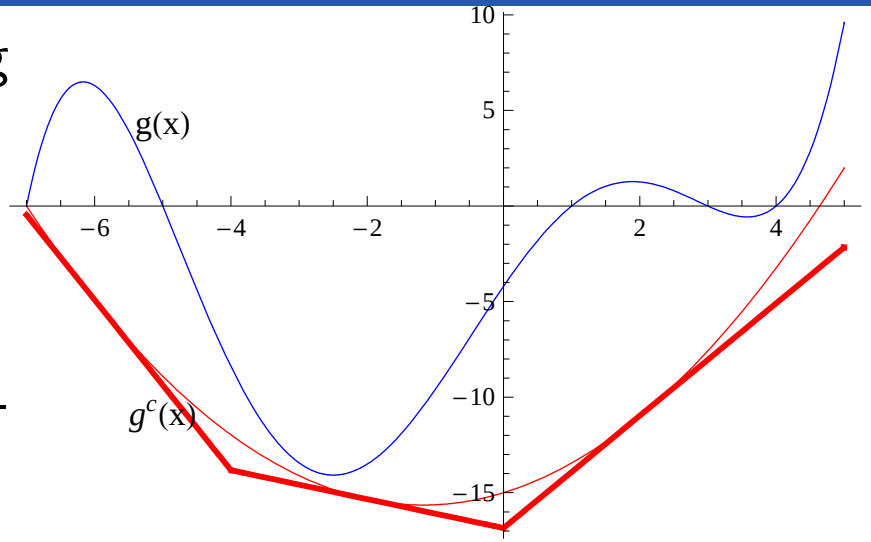

$$\{x \in [\underline{x}, \overline{x}] : g(x) \leq 0\} \subseteq \{x \in [\underline{x}, \overline{x}] : g^c(x) \leq 0\}$$

# The linear outer-approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [\underline{x}, \overline{x}]$$

Seek: A convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[\underline{x}, \overline{x}]$:

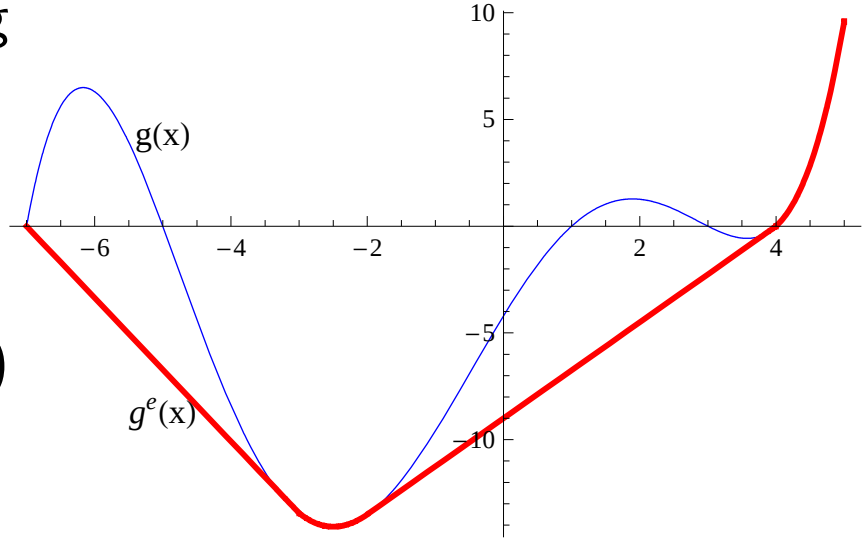



$$\{x \in [\underline{x}, \overline{x}] : g(x) \leq 0\} \subseteq \{x \in [\underline{x}, \overline{x}] : g^c(x) \leq 0\}$$
$$\subseteq \{x \in [\underline{x}, \overline{x}] : g^c(x^k) + \nabla g^c(x^k)^T (x - x^k) \leq 0,\ k \leq K\}$$

# The linear outer-approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [\underline{x}, \overline{x}]$$




Seek: The tightest convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[\underline{x}, \overline{x}]$:

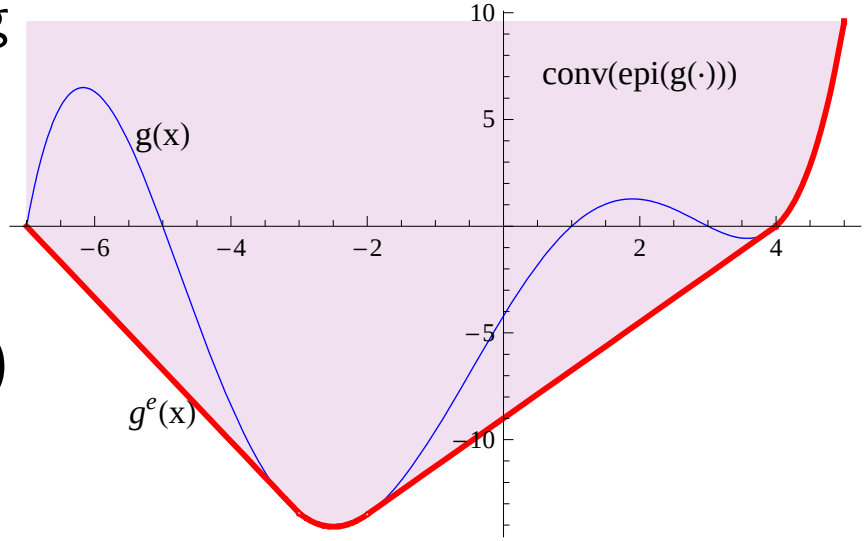
The convex envelope $g^e : [\underline{x}, \overline{x}] \to \mathbb{R}$ is a function such that

- ▷ analytically:
  - $g^e(x)$ is convex on $[\underline{x}, \overline{x}]$
  - $g^e(x) \leq g(x)$ for all $x \in [\underline{x}, \overline{x}]$
  - $\forall h : [\underline{x}, \overline{x}] \to \mathbb{R}$, $h(\cdot)$ convex, $h(\cdot) \leq g(\cdot)$: $h(\cdot) \leq g^e(\cdot)$

# The linear outer-approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [\underline{x}, \overline{x}]$$




Seek: The tightest convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[\underline{x}, \overline{x}]$:

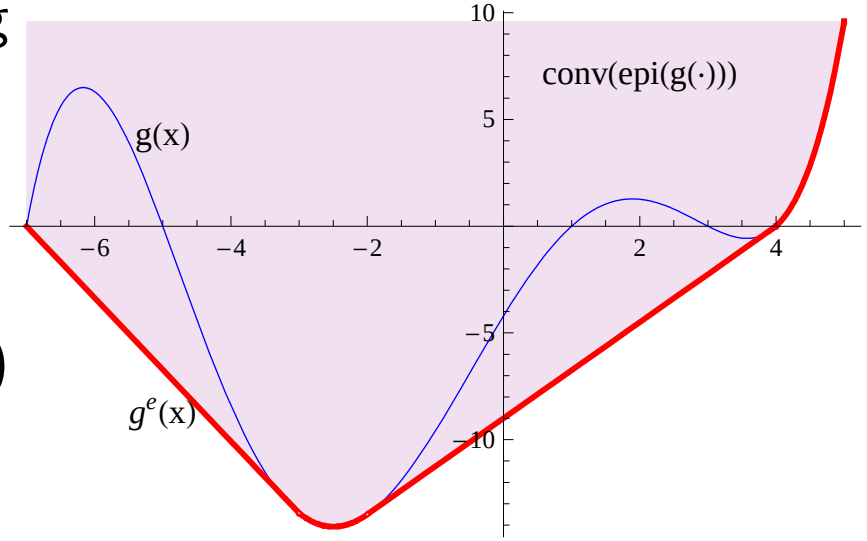
The convex envelope $g^e : [\underline{x}, \overline{x}] \to \mathbb{R}$ is a function such that

- ▷ analytically:
  - $g^e(x)$ is convex on $[\underline{x}, \overline{x}]$
  - $g^e(x) \leq g(x)$ for all $x \in [\underline{x}, \overline{x}]$
  - $\forall h : [\underline{x}, \overline{x}] \to \mathbb{R}$, $h(\cdot)$ convex, $h(\cdot) \leq g(\cdot)$: $h(\cdot) \leq g^e(\cdot)$
- ▷ geometrically: $\mathrm{epi}(g^e(\cdot)) = \mathrm{conv}(\mathrm{epi}(g(\cdot)))$

# The linear outer-approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [\underline{x}, \overline{x}]$$




Seek: The tightest convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[\underline{x}, \overline{x}]$:

The convex envelope $g^e : [\underline{x}, \overline{x}] \to \mathbb{R}$ is a function such that

- ▷ analytically:
  - $g^e(x)$ is convex on $[\underline{x}, \overline{x}]$
  - $g^e(x) \leq g(x)$ for all $x \in [\underline{x}, \overline{x}]$
  - $\forall h : [\underline{x}, \overline{x}] \to \mathbb{R}$, $h(\cdot)$ convex, $h(\cdot) \leq g(\cdot)$: $h(\cdot) \leq g^e(\cdot)$
- ▷ geometrically: $\mathrm{epi}(g^e(\cdot)) = \mathrm{conv}(\mathrm{epi}(g(\cdot)))$

In general: Convex envelopes are hard to find.

(finding the convex envelope of a function is as hard as finding its global min.)

# Convex envelopes for concave functions

# Convex envelopes for concave functions

**Theorem (Tawarmalani and Sahinidis [2002])**

*Let $v_1, \ldots, v_k$ be the vertices of a polytope $P$. The convex envelope $g^e(x)$ of a concave function $g(x)$ over $P$ can be expressed as*

$$g^e(x) = \min_{\alpha} \sum_{i=1}^{k} \lambda_i g(v_i)$$

$$\text{s.t.}\, x = \sum_{i=1}^{k} \lambda_i v_i, \qquad \sum_{i=1}^{k} \lambda_i = 1, \qquad \lambda_i \geq 0, \quad i = 1, \ldots, k$$

# Convex envelopes for concave functions

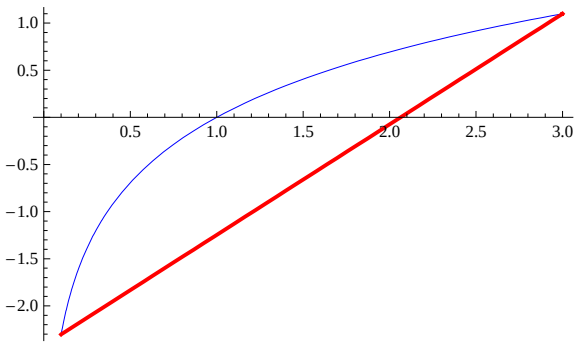
## Theorem (Tawarmalani and Sahinidis [2002])

*Let $v_1, \ldots, v_k$ be the vertices of a polytope $P$. The convex envelope $g^e(x)$ of a concave function $g(x)$ over $P$ can be expressed as*

$$g^e(x) = \min_{\alpha} \sum_{i=1}^{k} \lambda_i g(v_i)$$

$$\text{s.t.}\, x = \sum_{i=1}^{k} \lambda_i v_i, \qquad \sum_{i=1}^{k} \lambda_i = 1, \qquad \lambda_i \geq 0, \quad i = 1, \ldots, k$$

- For now: Special case $n = 1$, $P = [\underline{x}, \overline{x}]$
- Convex envelope for univariate concave functions is the secant between $\underline{x}$ and $\overline{x}$.
  $\Rightarrow \log(x), -\exp(x), -x^{2k}, \sqrt{x}, \ldots$



# Convex envelopes for concave functions

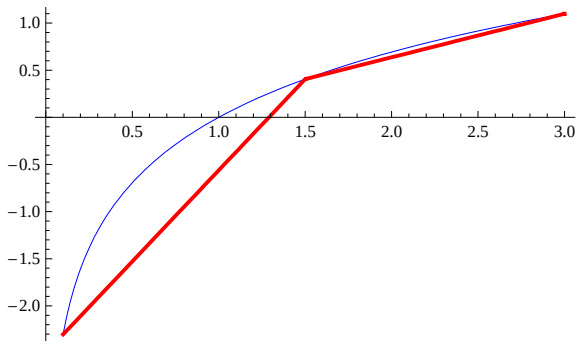
**Theorem (Tawarmalani and Sahinidis [2002])**

*Let $v_1, \ldots, v_k$ be the vertices of a polytope $P$. The convex envelope $g^e(x)$ of a concave function $g(x)$ over $P$ can be expressed as*

$$g^e(x) = \min_{\alpha} \sum_{i=1}^{k} \lambda_i g(v_i)$$

$$\text{s.t.}\, x = \sum_{i=1}^{k} \lambda_i v_i, \qquad \sum_{i=1}^{k} \lambda_i = 1, \qquad \lambda_i \geq 0, \quad i = 1, \ldots, k$$

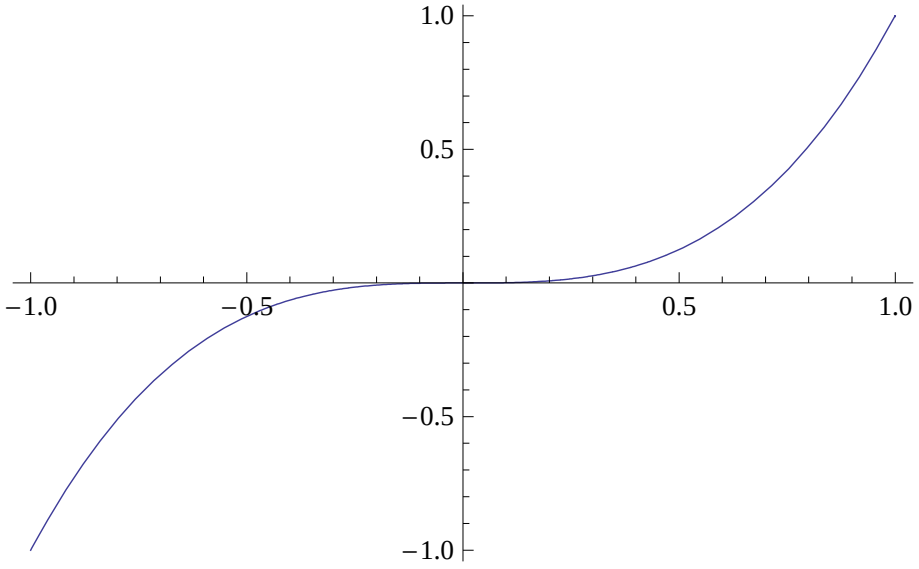
- For now: Special case $n = 1$, $P = [\underline{x}, \overline{x}]$
- Convex envelope for univariate concave functions is the secant between $\underline{x}$ and $\overline{x}$.
  $\Rightarrow \log(x), -\exp(x), -x^{2k}, \sqrt{x}, \ldots$
- In branch-and-bound: tighten by branching



# Convex envelope for monomials of odd degree

[Liberti and Pantelides, 2003]

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$



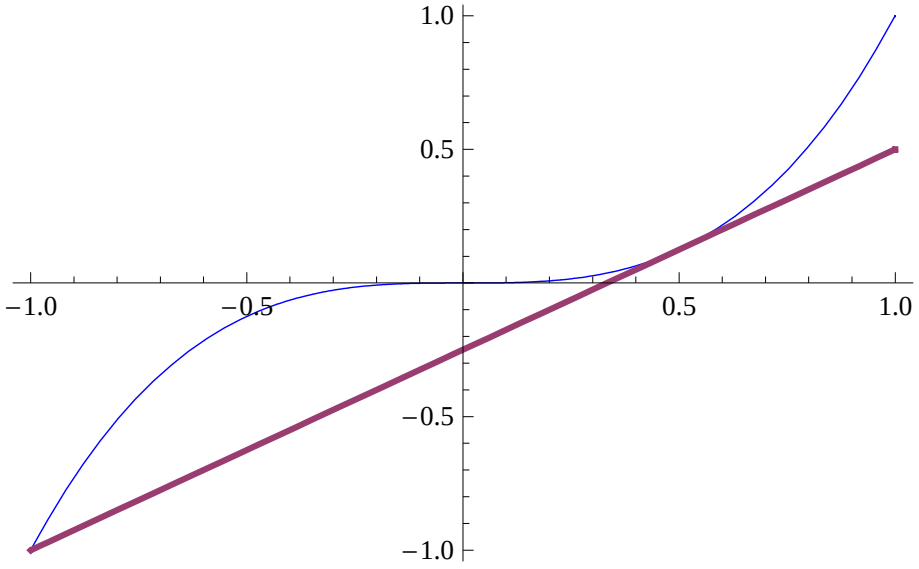
# Convex envelope for monomials of odd degree

[Liberti and Pantelides, 2003]

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$

- Seek for secant from $\underline{x}$ to $c$, where $c \geq 0$ is the coordinate where

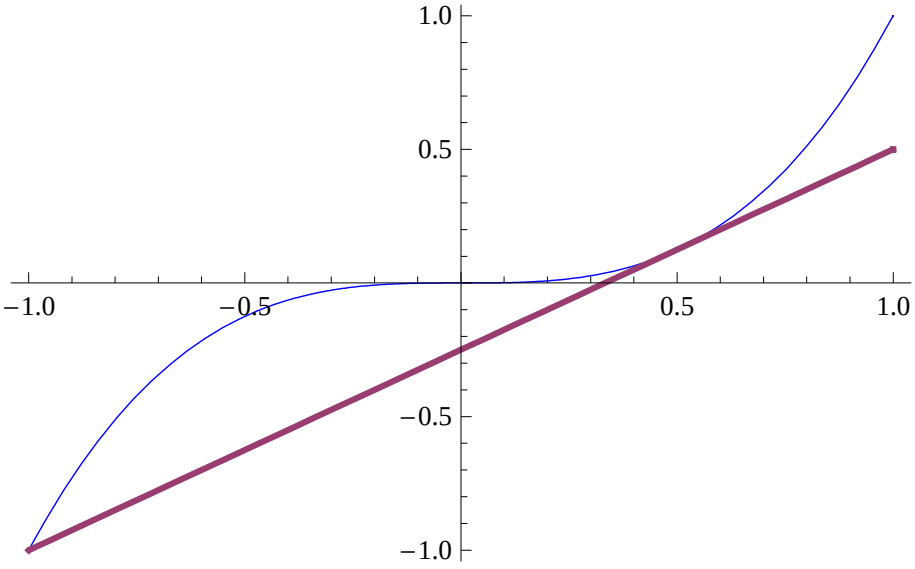
$$g'(c) = \frac{c^{2k+1} - \underline{x}^{2k+1}}{c - \underline{x}}$$



# Convex envelope for monomials of odd degree

[Liberti and Pantelides, 2003]

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$




- Seek for secant from $\underline{x}$ to $c$, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - \underline{x}^{2k+1}}{c - \underline{x}}$$

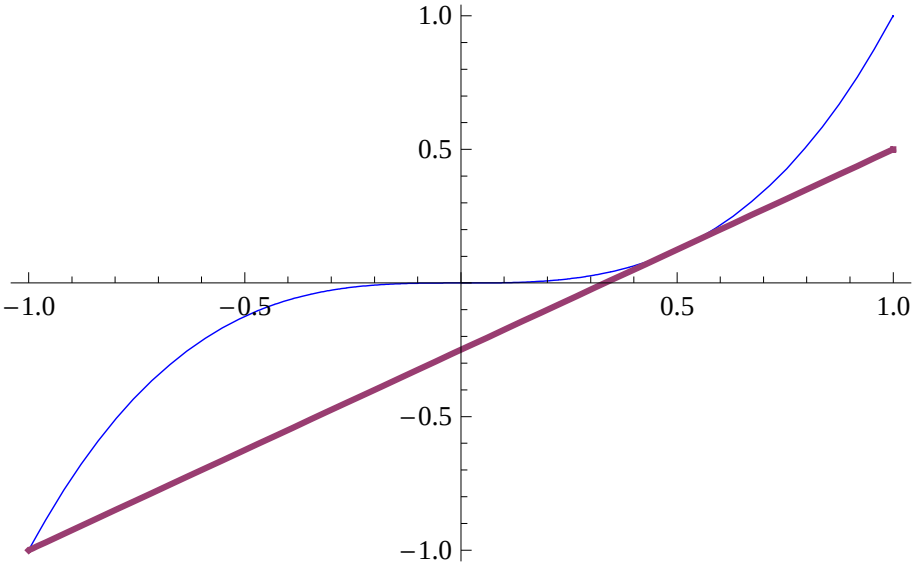
$\Rightarrow$ $c$ is the positive real root of the polynomial

$$2kx^{2k+1} - \underline{x}(2k+1)x^{2k} + \underline{x}^{2k+1}$$

# Convex envelope for monomials of odd degree

[Liberti and Pantelides, 2003]

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$




- Seek for secant from $\underline{x}$ to $c$, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - \underline{x}^{2k+1}}{c - \underline{x}}$$

$\Rightarrow$ $c$ is the positive real root of the polynomial

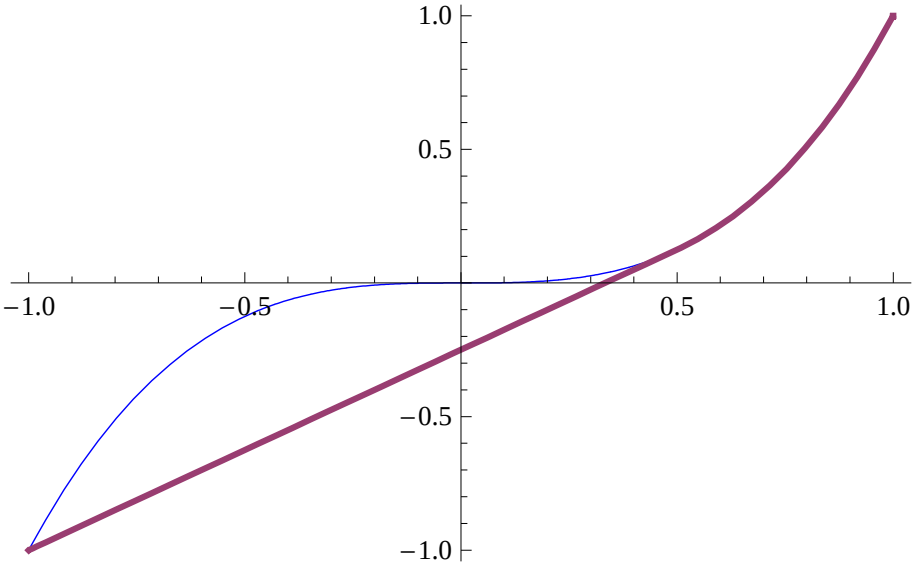
$$2kx^{2k+1} - \underline{x}(2k+1)x^{2k} + \underline{x}^{2k+1}$$

| $k$ | $c$ (for $\underline{x} = -1$) |
|---|---|
| 1 | 0.5000000000 |
| 2 | 0.6058295862 |
| 3 | 0.6703320476 |
| 4 | 0.7145377272 |
| 5 | 0.7470540749 |
| 6 | 0.7721416355 |
| 7 | 0.7921778546 |

# Convex envelope for monomials of odd degree

[Liberti and Pantelides, 2003]

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$




- Seek for secant from $\underline{x}$ to $c$, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - \underline{x}^{2k+1}}{c - \underline{x}}$$

$\Rightarrow$ $c$ is the positive real root of the polynomial

$$2kx^{2k+1} - \underline{x}(2k+1)x^{2k} + \underline{x}^{2k+1}$$

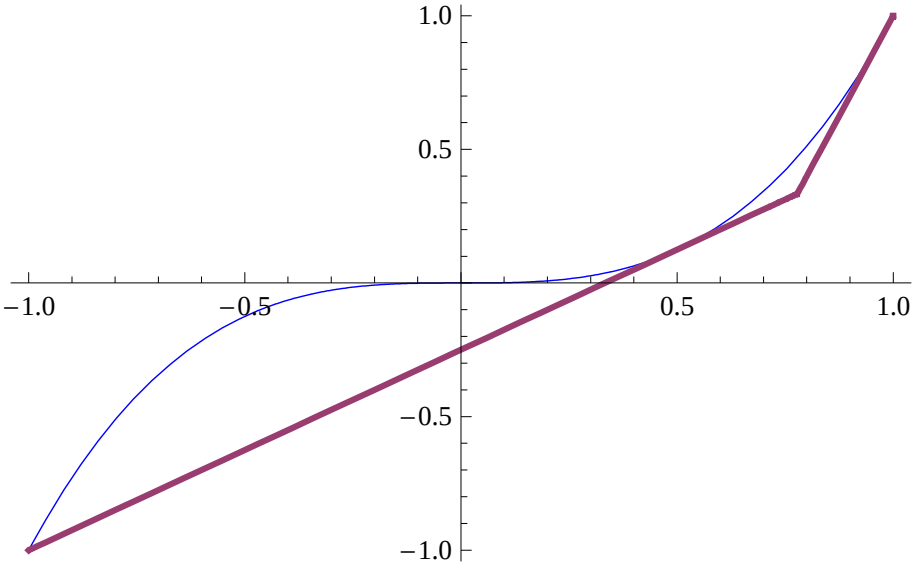
- obtain convex envelope

| $k$ | $c$ (for $\underline{x} = -1$) |
|---|---|
| 1 | 0.5000000000 |
| 2 | 0.6058295862 |
| 3 | 0.6703320476 |
| 4 | 0.7145377272 |
| 5 | 0.7470540749 |
| 6 | 0.7721416355 |
| 7 | 0.7921778546 |

# Convex envelope for monomials of odd degree

[Liberti and Pantelides, 2003]

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$




- Seek for secant from $\underline{x}$ to $c$, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - \underline{x}^{2k+1}}{c - \underline{x}}$$

$\Rightarrow$ $c$ is the positive real root of the polynomial

$$2kx^{2k+1} - \underline{x}(2k+1)x^{2k} + \underline{x}^{2k+1}$$

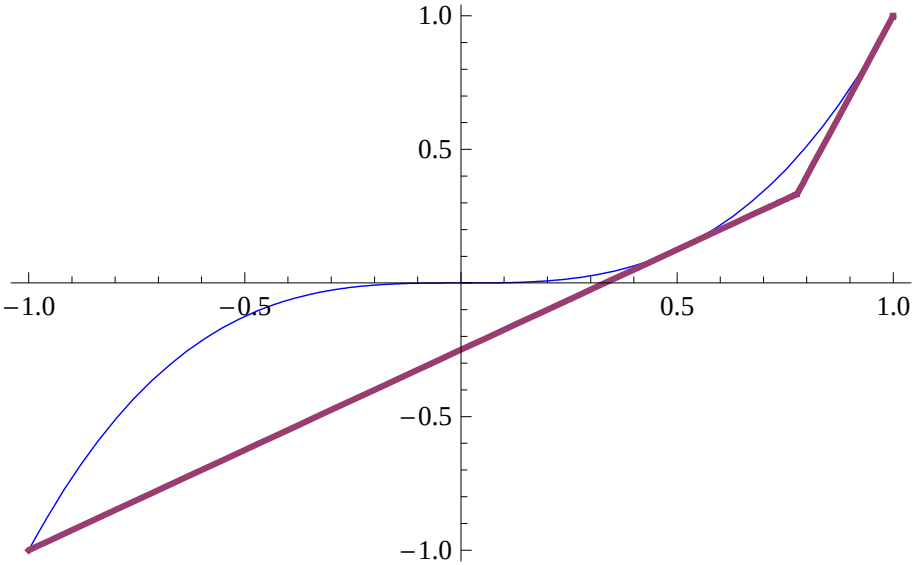
- obtain convex envelope and linear approximation

| $k$ | $c$ (for $\underline{x} = -1$) |
|---|---|
| 1 | 0.5000000000 |
| 2 | 0.6058295862 |
| 3 | 0.6703320476 |
| 4 | 0.7145377272 |
| 5 | 0.7470540749 |
| 6 | 0.7721416355 |
| 7 | 0.7921778546 |

# Convex envelope for monomials of odd degree

[Liberti and Pantelides, 2003]

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$




- Seek for secant from $\underline{x}$ to $c$, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - \underline{x}^{2k+1}}{c - \underline{x}}$$

$\Rightarrow$ $c$ is the positive real root of the polynomial

$$2kx^{2k+1} - \underline{x}(2k+1)x^{2k} + \underline{x}^{2k+1}$$

- obtain convex envelope and linear approximation
- easily generalized to $x|x|^{n-1} = \text{sign}(x)|x|^n$, $n \geq 1$ (application for flow of water and gas in pipes)

| $k$ | $c$ (for $\underline{x} = -1$) |
|---|---|
| 1 | 0.5000000000 |
| 2 | 0.6058295862 |
| 3 | 0.6703320476 |
| 4 | 0.7145377272 |
| 5 | 0.7470540749 |
| 6 | 0.7721416355 |
| 7 | 0.7921778546 |

# Convex envelope for $x \cdot y$

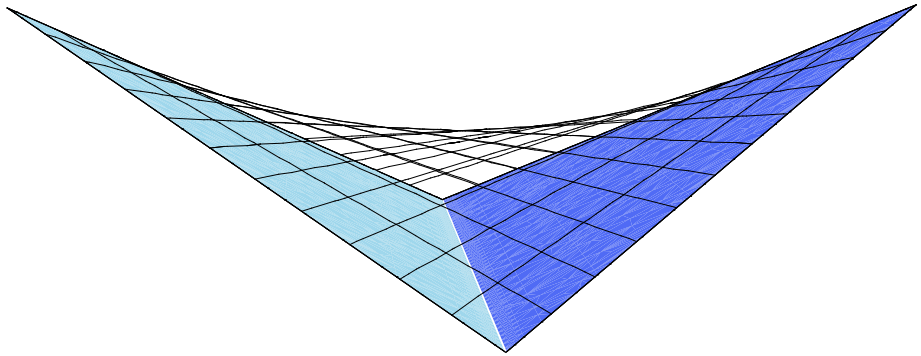
[McCormick, 1976]

McCormick: The convex envelope of

$$f(x) = xy$$

for $x \in [\underline{x}, \overline{x}]$, $y \in [\underline{y}, \overline{y}]$ is

$$xy \geq \max \left\{ \begin{array}{l} \overline{x}y + \overline{y}x - \overline{xy} \\ \underline{x}y + \underline{y}x - \underline{x}\underline{y} \end{array} \right\}$$



# Convex envelope for $x \cdot y$

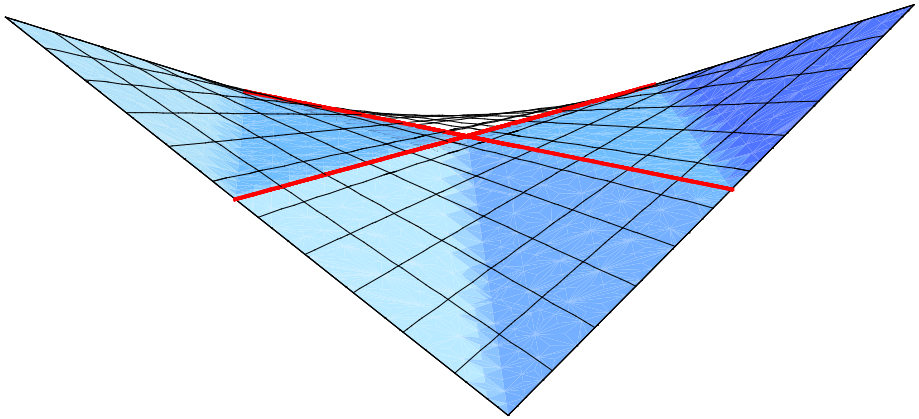
[McCormick, 1976]

McCormick: The convex envelope of

$$f(x) = xy$$

for $x \in [\underline{x}, \overline{x}]$, $y \in [\underline{y}, \overline{y}]$ is

$$xy \geq \max \left\{ \begin{array}{l} \overline{x}y + \overline{y}x - \overline{xy} \\ \underline{x}y + \underline{y}x - \underline{x}\underline{y} \end{array} \right\}$$



- apply McCormick underestimators to (nonconvex) quadratic forms

$$f(x) = \sum_{i,j} a_{i,j} x_i x_j$$

- branching on $x$ and $y$ changes variable bounds
  $\Rightarrow$ tighter underestimator $\Rightarrow$ improving dual bounds

# Reformulation for the general (factorable) case

[Tawarmalani and Sahinidis, 2004]

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

# Reformulation for the general (factorable) case

[Tawarmalani and Sahinidis, 2004]

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{\exp(y_1) y_2} \\
y_1 &= x_1 x_2 + x_3 \ln(x_4) \\
y_2 &= x_3^3
\end{aligned}
$$

# Reformulation for the general (factorable) case

[Tawarmalani and Sahinidis, 2004]

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{\exp(y_1) y_2} \\
y_1 &= x_1 x_2 + x_3 \ln(x_4) \\
y_2 &= x_3^3
\end{aligned}
$$

# Reformulation for the general (factorable) case

[Tawarmalani and Sahinidis, 2004]

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_3 y_2} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4)
\end{aligned}
$$

# Reformulation for the general (factorable) case

[Tawarmalani and Sahinidis, 2004]

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_3 y_2} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4)
\end{aligned}
$$

# Reformulation for the general (factorable) case

[Tawarmalani and Sahinidis, 2004]

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_5} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4) \\
y_5 &= y_3 y_2
\end{aligned}
$$

# Reformulation for the general (factorable) case

[Tawarmalani and Sahinidis, 2004]

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_5} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4) \\
y_5 &= y_3 y_2
\end{aligned}
$$

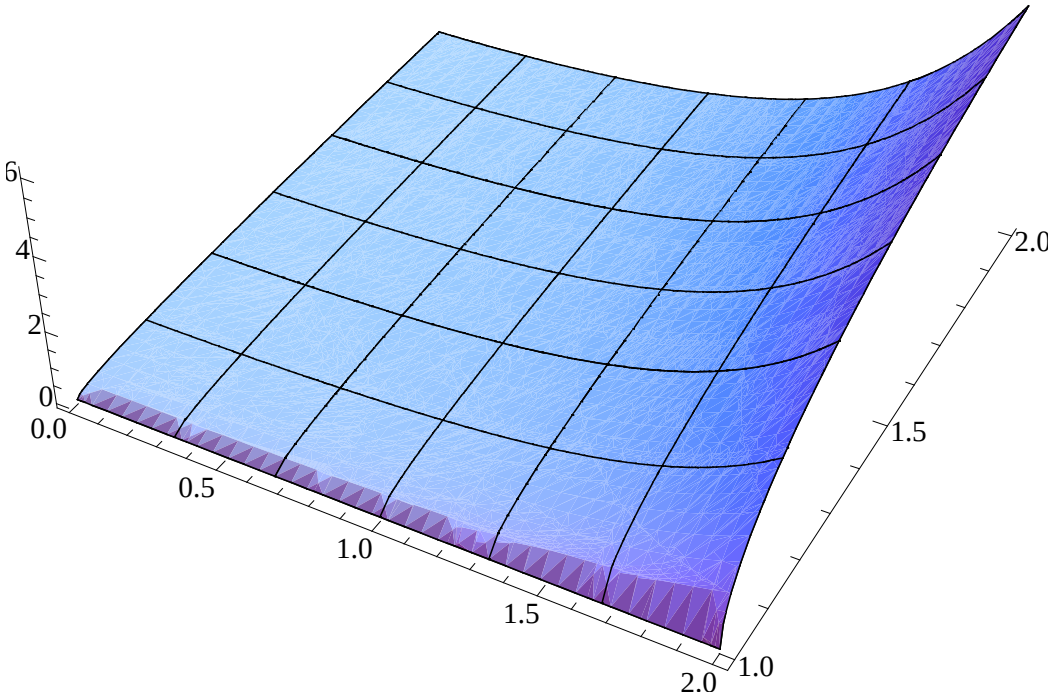
Convex outer-approximation by

- ▷ estimating univariate functions by known convex envelopes
- ▷ estimating bilinear terms by McCormick underestimators

# Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$

$$x_1 \in [0,2], \quad x_2 \in [1,2]$$



# Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$

$$x_1 \in [0,2], \quad x_2 \in [1,2]$$

Factorable Reformulation:

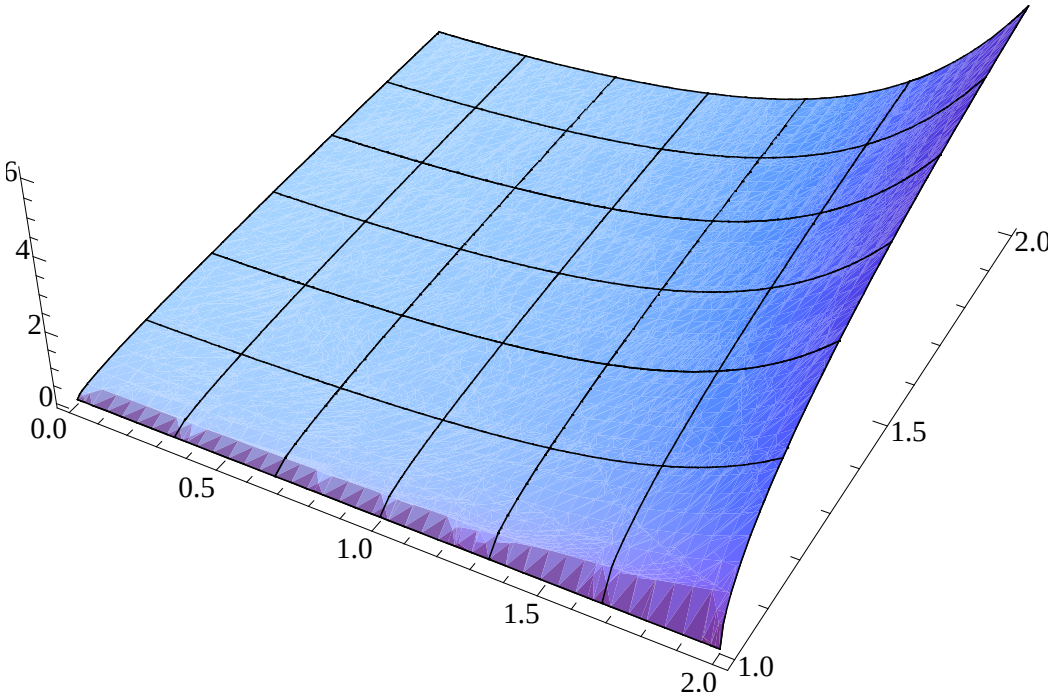
$$g = \sqrt{y_1}$$

$$y_1 = y_2 y_3$$

$$y_2 = \exp(y_4)$$

$$y_3 = \ln(x_2)$$

$$y_4 = x_1^2$$



# Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$

$$x_1 \in [0,2], \quad x_2 \in [1,2]$$

Factorable Reformulation:

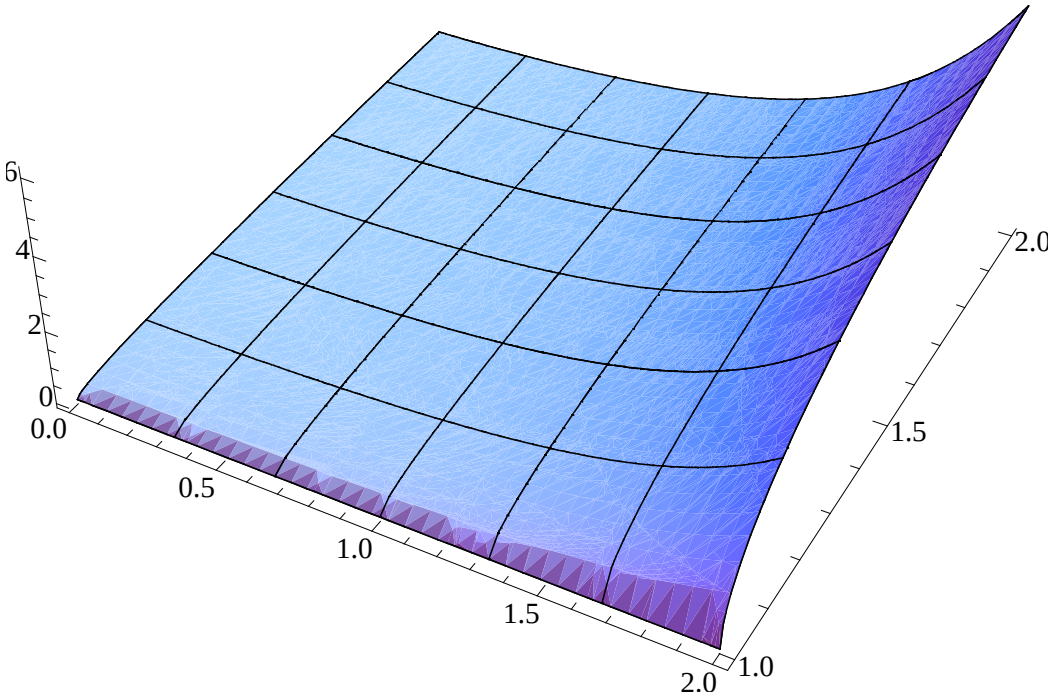
$$g = \sqrt{y_1}$$

$$y_1 = y_2 y_3 \qquad [0, \ln(2)e^4]$$

$$y_2 = \exp(y_4) \qquad [1, e^4]$$

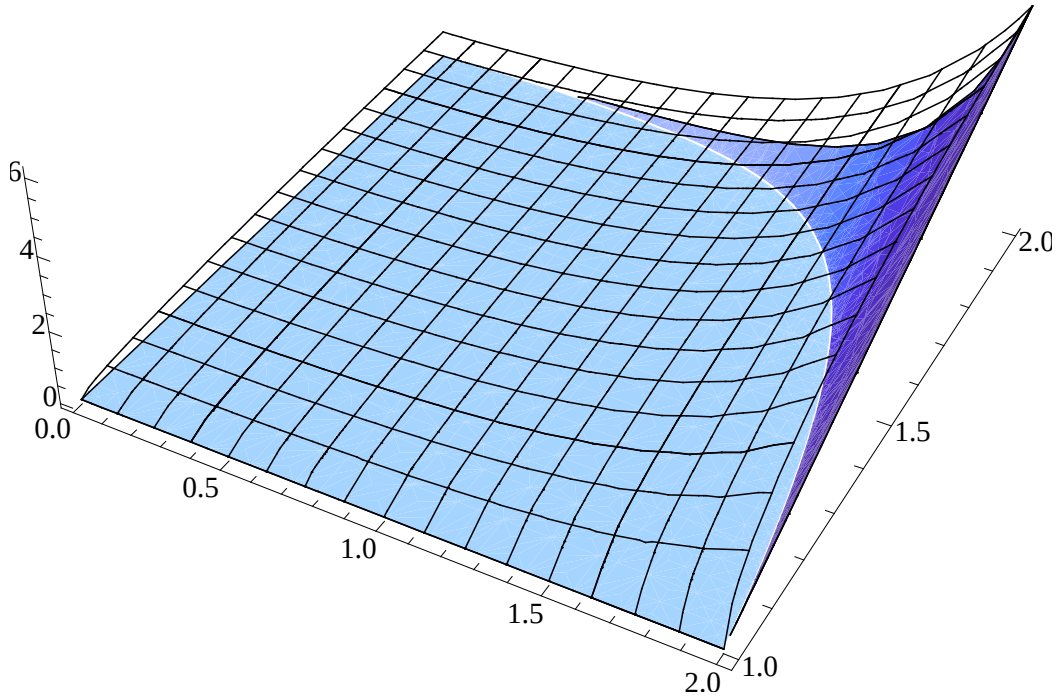
$$y_3 = \ln(x_2) \qquad [0, \ln(2)]$$

$$y_4 = x_1^2 \qquad [0, 4]$$



# Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$




Factorable Reformulation:

$$\begin{aligned}
g &= \sqrt{y_1} & & \\
y_1 &= y_2 y_3 & & [0, \ln(2)e^4] \\
y_2 &= \exp(y_4) & & [1, e^4] \\
y_3 &= \ln(x_2) & & [0, \ln(2)] \\
y_4 &= x_1^2 & & [0, 4]
\end{aligned}$$

Convex relaxation:

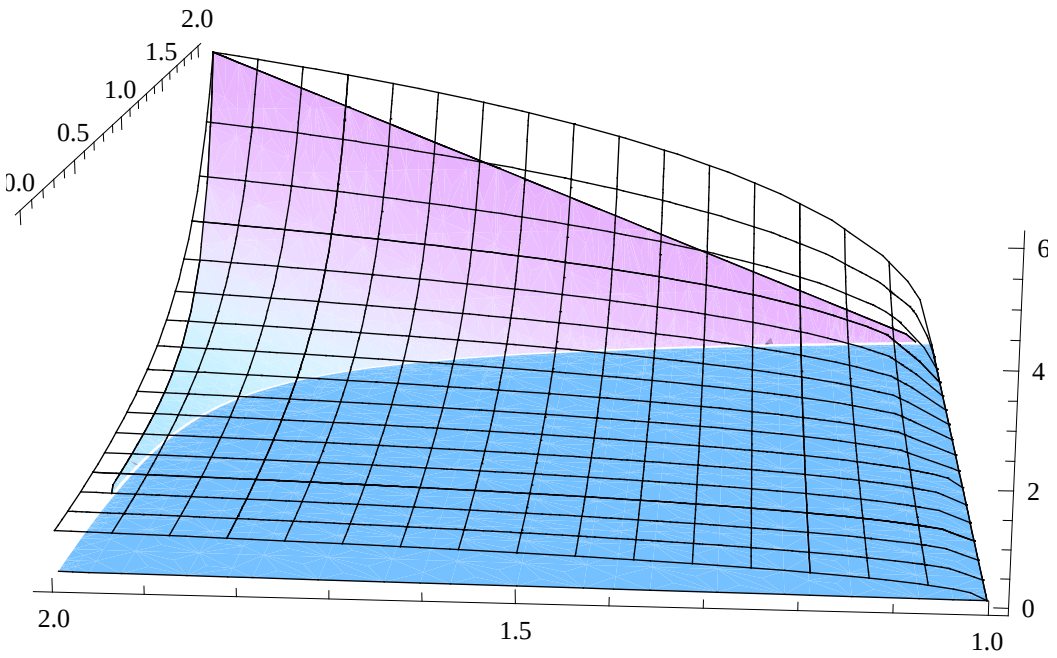
$$\sqrt{\underline{y_1}} + \frac{y_1 - \underline{y_1}}{\sqrt{\overline{y_1}} + \sqrt{\underline{y_1}}} \le g \le \sqrt{y_1}; \quad \ln \underline{x_1} + (x_2 - \underline{x_2})\frac{\ln \overline{x_2} - \ln \underline{x_2}}{\overline{x_2} - \underline{x_2}} \le y_3 \le \ln(x_2)$$

$$\max \left\{ \begin{array}{c} \overline{y_2} y_3 + \overline{y_3} y_2 - \overline{y_2 y_3} \\ \underline{y_2} y_3 + \underline{y_3} y_2 - \underline{y_2 y_3} \end{array} \right\} \le y_1 \le \min \left\{ \begin{array}{c} \overline{y_2} y_3 + \underline{y_3} y_2 - \overline{y_2}\,\underline{y_3} \\ \underline{y_2} y_3 + \overline{y_3} y_2 - \underline{y_2}\,\overline{y_3} \end{array} \right\}$$

...

# Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$




Factorable Reformulation:

$$\begin{aligned} g &= \sqrt{y_1} & \\ y_1 &= y_2 y_3 & [0, \ln(2)e^4] \\ y_2 &= \exp(y_4) & [1, e^4] \\ y_3 &= \ln(x_2) & [0, \ln(2)] \\ y_4 &= x_1^2 & [0, 4] \end{aligned}$$

Convex relaxation:

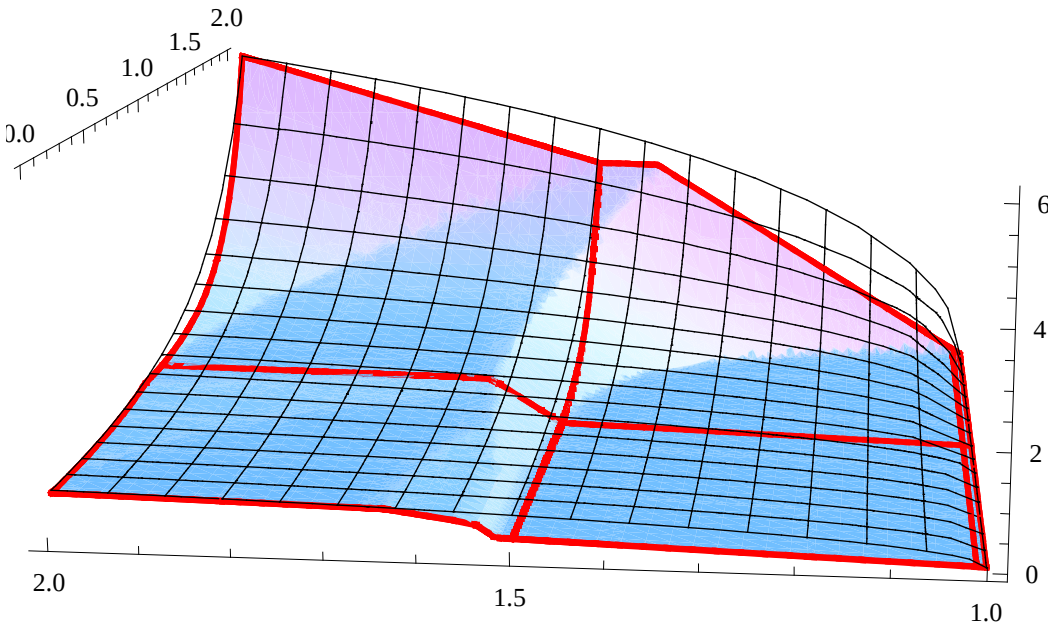
$$\sqrt{\underline{y_1}} + \frac{y_1 - \underline{y_1}}{\sqrt{\overline{y_1}} + \sqrt{\underline{y_1}}} \le g \le \sqrt{y_1}; \quad \ln \underline{x_1} + (x_2 - \underline{x_2})\frac{\ln \overline{x_2} - \ln \underline{x_2}}{\overline{x_2} - \underline{x_2}} \le y_3 \le \ln(x_2)$$

$$\max \left\{ \begin{array}{c} \overline{y_2} y_3 + \overline{y_3} y_2 - \overline{y_2 y_3} \\ \underline{y_2} y_3 + \underline{y_3} y_2 - \underline{y_2 y_3} \end{array} \right\} \le y_1 \le \min \left\{ \begin{array}{c} \overline{y_2} y_3 + \underline{y_3} y_2 - \overline{y_2}\,\underline{y_3} \\ \underline{y_2} y_3 + \overline{y_3} y_2 - \underline{y_2}\,\overline{y_3} \end{array} \right\}$$

...

# Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$




Factorable Reformulation:

$$\begin{aligned}
g &= \sqrt{y_1} & \\
y_1 &= y_2 y_3 & [0, \ln(2)e^4] \\
y_2 &= \exp(y_4) & [1, e^4] \\
y_3 &= \ln(x_2) & [0, \ln(2)] \\
y_4 &= x_1^2 & [0, 4]
\end{aligned}$$

Convex relaxation:

$$\sqrt{\underline{y_1}} + \frac{y_1 - \underline{y_1}}{\sqrt{\overline{y_1}} + \sqrt{\underline{y_1}}} \le g \le \sqrt{y_1}; \quad \ln \underline{x_1} + (x_2 - \underline{x_2})\frac{\ln \overline{x_2} - \ln \underline{x_2}}{\overline{x_2} - \underline{x_2}} \le y_3 \le \ln(x_2)$$

$$\max\left\{ \begin{array}{c} \overline{y_2}y_3 + \overline{y_3}y_2 - \overline{y_2}\overline{y_3} \\ \underline{y_2}y_3 + \underline{y_3}y_2 - \underline{y_2}\underline{y_3} \end{array} \right\} \le y_1 \le \min\left\{ \begin{array}{c} \overline{y_2}y_3 + \underline{y_3}y_2 - \overline{y_2}\underline{y_3} \\ \underline{y_2}y_3 + \overline{y_3}y_2 - \underline{y_2}\overline{y_3} \end{array} \right\}$$

...

# Outline

## BARON = Branch And Reduce Optimization Navigator

[Tawarmalani and Sahinidis, 2002]

- developed by N. Sahinidis, M. Tawarmalani
- state-of-the-art convexification techniques
- winner of Beale-Orchard-Hays Prize 2006

## BARON = Branch And Reduce Optimization Navigator

[Tawarmalani and Sahinidis, 2002]


- developed by N. Sahinidis, M. Tawarmalani
- state-of-the-art convexification techniques
- winner of Beale-Orchard-Hays Prize 2006

## Couenne = Convex Over + Under ENvelopes for Nonlinear Estimation

[Belotti et al., 2009]

- COIN-OR solver developed by P. Belotti et.al.
- methodology similar to BARON
- winner of COIN-OR cup competition 2010

## BARON = Branch And Reduce Optimization Navigator

[Tawarmalani and Sahinidis, 2002]


- developed by N. Sahinidis, M. Tawarmalani
- state-of-the-art convexification techniques
- winner of Beale-Orchard-Hays Prize 2006

## Couenne = Convex Over + Under ENvelopes for Nonlinear Estimation

[Belotti et al., 2009]

- COIN-OR solver developed by P. Belotti et.al.
- methodology similar to BARON
- winner of COIN-OR cup competition 2010

## LindoGlobal

[Lin and Schrage, 2009]

- developed by Y. Lin and L. Schrage
- some automatic reformulation techniques (SOC, `floor`, `abs`, `max`, ...)

LINDO SYSTEMS INC.

# Solvers for nonconvex MINLP in Development

MINOTAUR = Mixed-Integer Nonconvex Optimization Toolbox – Algorithms, Underestimators, Relaxations [Mahajan and Munson, 2010]

- developed by A. Mahajan, T. Munson, et.al.
- emphasis on reformulation techniques (multilinear forms, SOC)

# Solvers for nonconvex MINLP in Development

## MINOTAUR = Mixed-Integer Nonconvex Optimization Toolbox – Algorithms, Underestimators, Relaxations [Mahajan and Munson, 2010]

- developed by A. Mahajan, T. Munson, et.al.
- emphasis on reformulation techniques (multilinear forms, SOC)


## COCONUT = COntinuous CONstraints - Updating the Technology

[Neumaier, 2001]

- developed by H. Schichl, A. Neumaier, et.al. (Vienna)
- emphasis on rigorous computations
- many interval-arithmetic based techniques


# Solvers for nonconvex MINLP in Development

## MINOTAUR = Mixed-Integer Nonconvex Optimization Toolbox – Algorithms, Underestimators, Relaxations [Mahajan and Munson, 2010]

- developed by A. Mahajan, T. Munson, et.al.
- emphasis on reformulation techniques (multilinear forms, SOC)

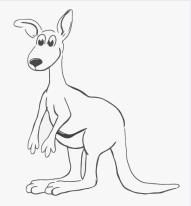

## COCONUT = COntinuous CONstraints - Updating the Technology

[Neumaier, 2001]

- developed by H. Schichl, A. Neumaier, et.al. (Vienna)
- emphasis on rigorous computations
- many interval-arithmetic based techniques


## SCIP = Solving Constraint Integer Programs [Berthold et al., 2009a]

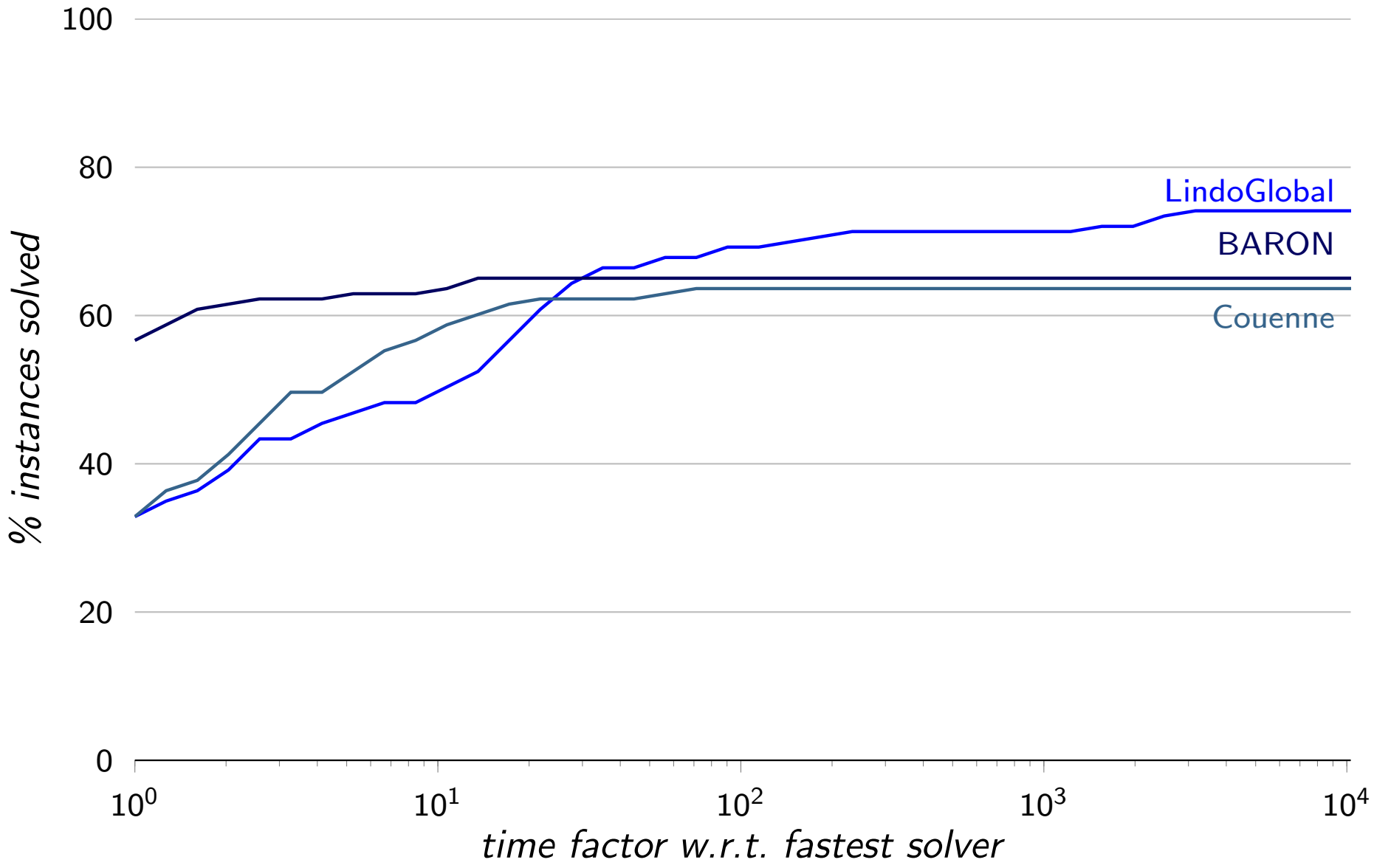
- support for MIQQP so far
- strong in MIP part



Experimental Environment:

- test set: 143 MINLP instances from MINLPLib [Bussieck et al., 2003]
  convex and nonconvex, easy and hard
- solvers:
  - BARON 9.0.8
    CPLEX for LPs and MINOS for NLPs
  - LindoGlobal 6.1.1
    CPLEX for LPs and CONOPT for NLPs
  - Couenne 0.3
    CLP for LPs and IPOPT for NLPs
- timelimit 1 hour, gap tolerance $10^{-4}$
- 3.0GHz Intel Core2 Extreme CPU X9650, 16GB RAM

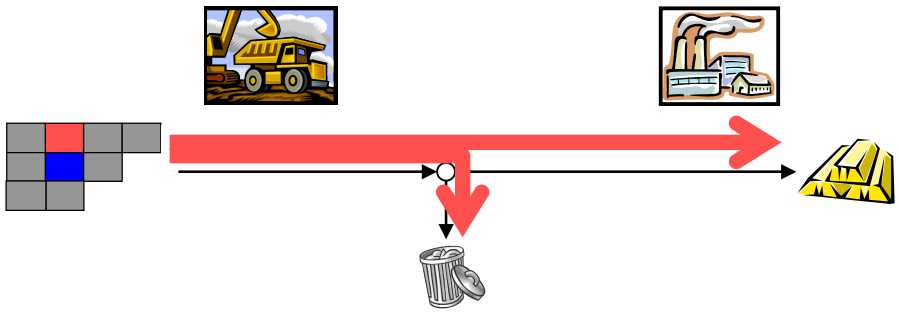
# Benchmark on 143 MINLPs



# Outline

## Open Pit Mine Production Scheduling with a Stockpile [Bley et al., 2009a]

# Open Pit Mine Production Scheduling – Basic Version



Variables:

$x_{i,t} \in \{0,1\}$ block i fully mined by t

$f^{m}_{i,t} \in [0,1]$ % of block i mined in t

$f^{p}_{i,t} \in [0,1]$ % of block i processed in t

Constraints:

- material flow conservation
- mining & processing capacities
- mining precedences

Given:

- Blocks (or aggregates)
  (rock, metal grade, precedences)
- Capacities, Costs
  (Mining, Processing; per rock ton)
- Metal Prices

Decisions:

- When to mine which block?
- Which block to process?

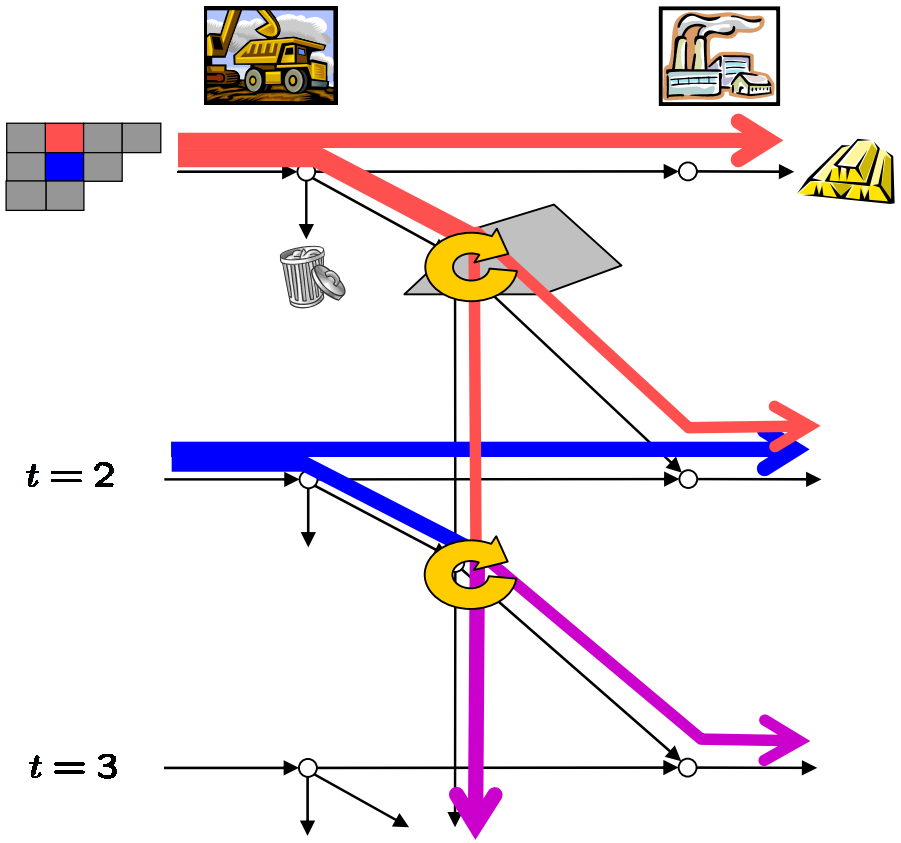
Goal:

max NPV

Solution Approach:

Time-indexed MILP formulation

# Open Pit Mine Production Scheduling with Stockpiles




Practical extension:

Stockpile for interim storage

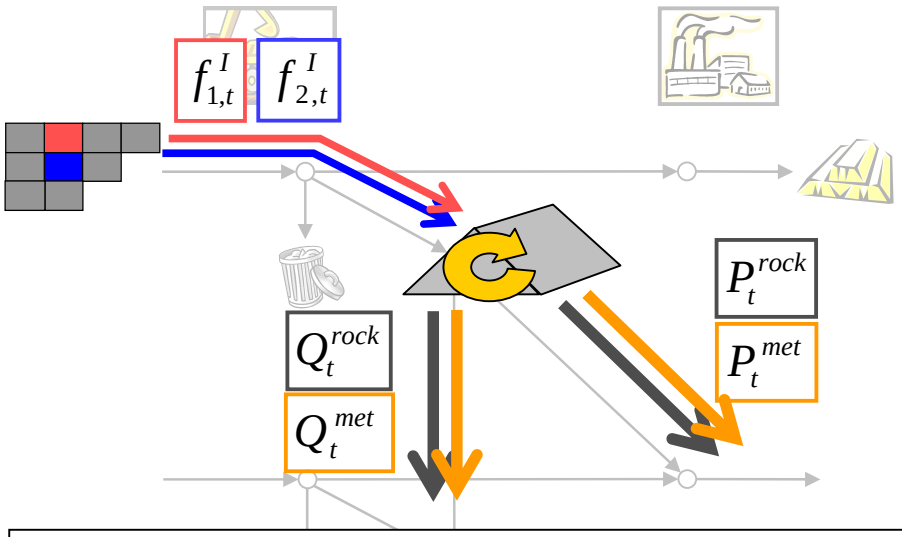
⇩

better use of capacities

Difficulty:

stockpile mixes material

# Open Pit Mine Production Scheduling with Stockpiles




Aggregated stockpile model

$f^I_{i,t} \in [0,1]$ % of block i into stockpiled

$Q^{rock}_t, Q^{met}_t$ total rock / metal tons held

$P^{rock}_t, P^{met}_t$ total rock / metal tons out

Mixing constraints:

$$\frac{P^{met}_t}{Q^{met}_t} = \frac{P^{rock}_t}{Q^{rock}_t}$$

(metal fraction out = rock fraction out)

## Practical extension:

Stockpile for interim storage

⇩

better use of capacities

## Difficulty:

stockpile mixes material

## Mathematical model:

Same rock / metal – mix held and taken out of stockpile

⇩

Nonlinear constraints

# Aggregate MINLP formulation

$$\max \quad \sum_{t \in years} \left[ v_t \left( P_t^{met} + \sum_{i \in blocks} \alpha_i^{met} f_{i,t}^P \right) - c_t^P \left( P_t^{rock} + \sum_{i \in blocks} \alpha_i^{rock} f_{i,t}^P \right) - c_t^M \sum_{i \in blocks} \alpha_i^{rock} f_{i,t}^M \right]$$

s.t.

$$\left.\begin{aligned} \sum_{i \in blocks} \alpha_i^{rock} f_{i,t}^M &\le MC_t \qquad \forall\, t \\ P_t^{rock} + \sum_{i \in blocks} \alpha_i^{rock} f_{i,t}^P &\le PC_t \qquad \forall\, t \end{aligned}\right\} \text{Capacities}$$

$$\left.\begin{aligned} \sum_{s=1,\ldots,t} f_{i,s}^M &\ge x_{i,t} \qquad \forall\, t,\ i \\ \sum_{s=1,\ldots,t} f_{i,s}^M &\le x_{j,t} \qquad \forall\, t,\ j \in \mathrm{Pred}(i) \end{aligned}\right\} \text{Precedences}$$

$$\left.\begin{aligned} \sum_{i \in blocks} \alpha_i^{rock} f_{i,t}^I + Q_t^{rock} - Q_{t+1}^{rock} - R_{t+1}^{rock} &= 0 \qquad \forall\, t \\ &\vdots \end{aligned}\right\} \text{Flow balance}$$

$$P_t^{met} Q_t^{rock} - Q_t^{met} P_t^{rock} = 0 \qquad \forall\, t \qquad \text{— Stockpile mixing}$$

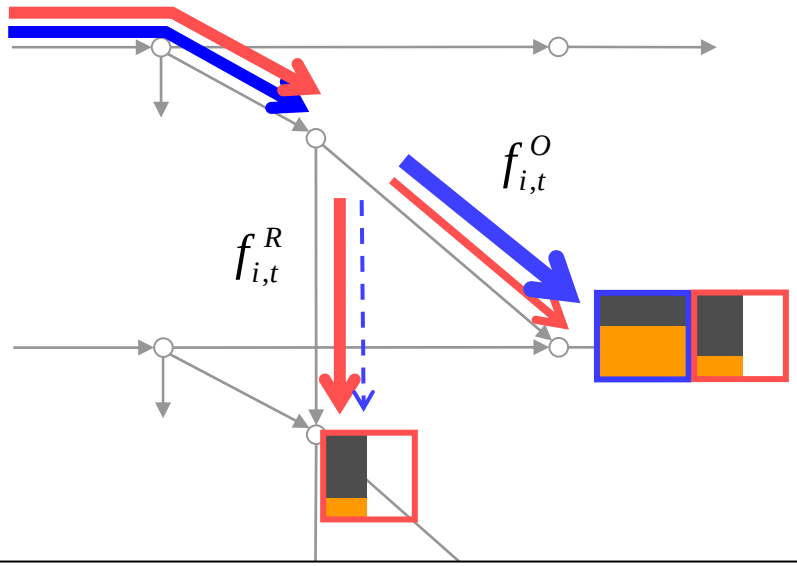
$$0 \le f_{i,t}^* \le 1$$

$$P_t^*, Q_t^* \ge 0$$

$$x_{i,t} \in \{0,1\}$$

# Extended MINLP formulation

- track each block's fractions individually through the stockpile
- best possible knowledge of the material in the stockpile




$f_{i,t}^{O} \in [0,1]$ % of block i out of stockpile

$f_{i,t}^{R} \in [0,1]$ % of block i kept in stockpile

$$P_t^{rock} = \sum_{i \in blocks} \alpha_i^{rock} f_{i,t}^{O}$$ total rock out

$\vdots$

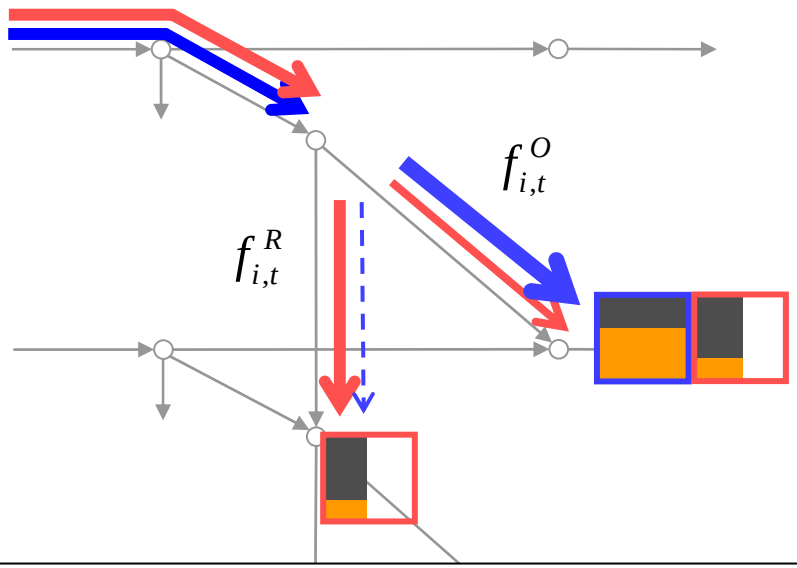
# Extended MINLP formulation

- track each block's fractions individually through the stockpile
- best possible knowledge of the material in the stockpile
- require

$$\frac{f_{i,t}^{O}}{f_{i,t}^{O} + f_{i,t}^{R}} = r_t$$

for each period and block

$\Rightarrow$ much tighter dual bound




$f_{i,t}^{O} \in [0,1]$ % of block i out of stockpile

$f_{i,t}^{R} \in [0,1]$ % of block i kept in stockpile

$$P_t^{rock} = \sum_{i \in blocks} \alpha_i^{rock} f_{i,t}^{O}$$ total rock out

$\vdots$

## State-of-the-art B&B-algorithm [Bley et al., 2009a]

- ignore nonlinearities to obtain linear relaxation
  and enforce these by special geometric branching scheme

## State-of-the-art B&B-algorithm [Bley et al., 2009a]

- ignore nonlinearities to obtain linear relaxation
  and enforce these by special geometric branching scheme
- application-specific heuristic for primal solutions

## State-of-the-art B&B-algorithm [Bley et al., 2009a]

- ignore nonlinearities to obtain linear relaxation
  and enforce these by special geometric branching scheme
- application-specific heuristic for primal solutions
- variable fixing and cutting planes from linear precedence constrained knapsack structure

## State-of-the-art B&B-algorithm [Bley et al., 2009a]

- ignore nonlinearities to obtain linear relaxation
  and enforce these by special geometric branching scheme
- application-specific heuristic for primal solutions
- variable fixing and cutting planes from linear precedence constrained knapsack structure

How close can generic MINLP solvers get without problem-specific knowledge? [Bley et al., 2009b]

# SCIP: Handling bilinear constraints

- Mixing constraints:

$$\frac{f_{i,t}^{O}}{f_{i,t}^{O} + f_{i,t}^{R}} = r_t$$

# SCIP: Handling bilinear constraints

- Mixing constraints:

$$f_{i,t}^{O} \quad = r_t(f_{i,t}^{O} + f_{i,t}^{R})$$

with bounds $r_t \in [\underline{r}_t, \overline{r}_t]$, $f_{i,t}^{O} + f_{i,t}^{R} \in [\underline{f}_{i,t}^{O} + \underline{f}_{i,t}^{R}, \overline{f}_{i,t}^{O} + \overline{f}_{i,t}^{R}]$.

# SCIP: Handling bilinear constraints

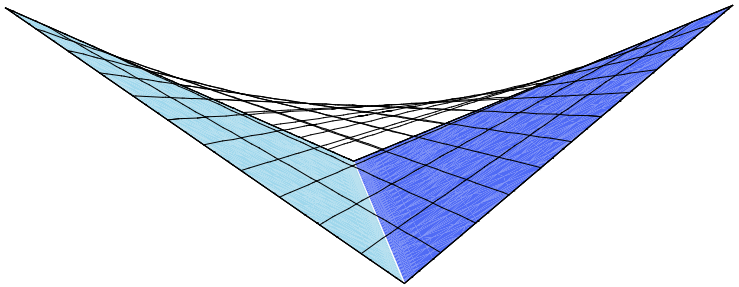
- Mixing constraints:

$$f_{i,t}^{O} \quad = r_t(f_{i,t}^{O} + f_{i,t}^{R})$$

with bounds $r_t \in [\underline{r}_t, \overline{r}_t]$, $f_{i,t}^{O} + f_{i,t}^{R} \in [\underline{f}_{i,t}^{O} + \underline{f}_{i,t}^{R}, \overline{f}_{i,t}^{O} + \overline{f}_{i,t}^{R}]$.

- McCormick underestimators provide a linear outer approximation for each bilinear term $(xy)$:

$$\begin{aligned}
xy &\geq \underline{x}y + \underline{y}x - \underline{x}\underline{y} \\
xy &\geq \overline{x}y + \overline{y}x - \overline{x}\overline{y} \\
xy &\leq \underline{x}y + \overline{y}x - \underline{x}\overline{y} \\
xy &\leq \overline{x}y + \underline{y}x - \overline{x}\underline{y}
\end{aligned}$$



# SCIP: Handling bilinear constraints

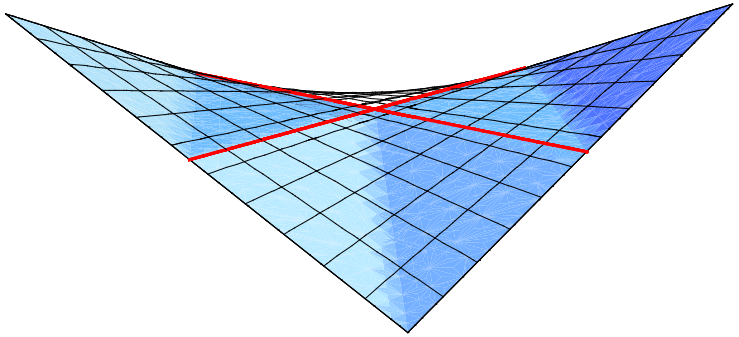
- Mixing constraints:

$$f_{i,t}^{O} \quad = r_t(f_{i,t}^{O} + f_{i,t}^{R})$$

with bounds $r_t \in [\underline{r}_t, \overline{r}_t]$, $f_{i,t}^{O} + f_{i,t}^{R} \in [\underline{f}_{i,t}^{O} + \underline{f}_{i,t}^{R}, \overline{f}_{i,t}^{O} + \overline{f}_{i,t}^{R}]$.

- McCormick underestimators provide a linear outer approximation for each bilinear term $(xy)$:

$$\begin{aligned} xy &\geq \underline{x}y + \underline{y}x - \underline{x}\underline{y} \\ xy &\geq \overline{x}y + \overline{y}x - \overline{x}\overline{y} \\ xy &\leq \underline{x}y + \overline{y}x - \underline{x}\overline{y} \\ xy &\leq \overline{x}y + \underline{y}x - \overline{x}\underline{y} \end{aligned}$$



- Separate dynamically if violated, otherwise perform spatial branching

# SCIP: Producing feasible solutions

Primal solutions from: feasible LP solution, MIP heuristics, NLP local search, extended LNS heuristics

New heuristic: Undercover [Berthold and Gleixner, 2009]

- idea: fix variables such as to obtain a linear subproblem
- generic: solve an auxiliary set covering problem to automatically detect a minimal set of variables to fix
- use LP or NLP relaxation for fixing values

# SCIP: Producing feasible solutions

Primal solutions from: feasible LP solution, MIP heuristics, NLP local search, extended LNS heuristics

New heuristic: Undercover [Berthold and Gleixner, 2009]

- idea: fix variables such as to obtain a linear subproblem
- generic: solve an auxiliary set covering problem to automatically detect a minimal set of variables to fix
- use LP or NLP relaxation for fixing values

Here: Fixing only $r_1, \ldots, r_T$ linearises the mixing constraints

$$f_{i,t}^O = r_t(f_{i,t}^O + f_{i,t}^R)$$

- use fixing values from NLP relaxation solved to local optimality
- solve resulting sub-MIP with node and stall node limit

# SCIP: Producing feasible solutions

Primal solutions from: feasible LP solution, MIP heuristics, NLP local search, extended LNS heuristics

New heuristic: Undercover [Berthold and Gleixner, 2009]

- idea: fix variables such as to obtain a linear subproblem
- generic: solve an auxiliary set covering problem to automatically detect a minimal set of variables to fix
- use LP or NLP relaxation for fixing values

Here: Fixing only $r_1, \ldots, r_T$ linearises the mixing constraints

$$f_{i,t}^O = r_t(f_{i,t}^O + f_{i,t}^R)$$

- use fixing values from NLP relaxation solved to local optimality
- solve resulting sub-MIP with node and stall node limit

Expensive, but effective heuristic: produces fully feasible solutions within 1% of the global optimum.

# Problem-specific vs. generic algorithms

Compare the performance of

- BARON 9.0.2
- Couenne 0.2
- SCIP 2.0

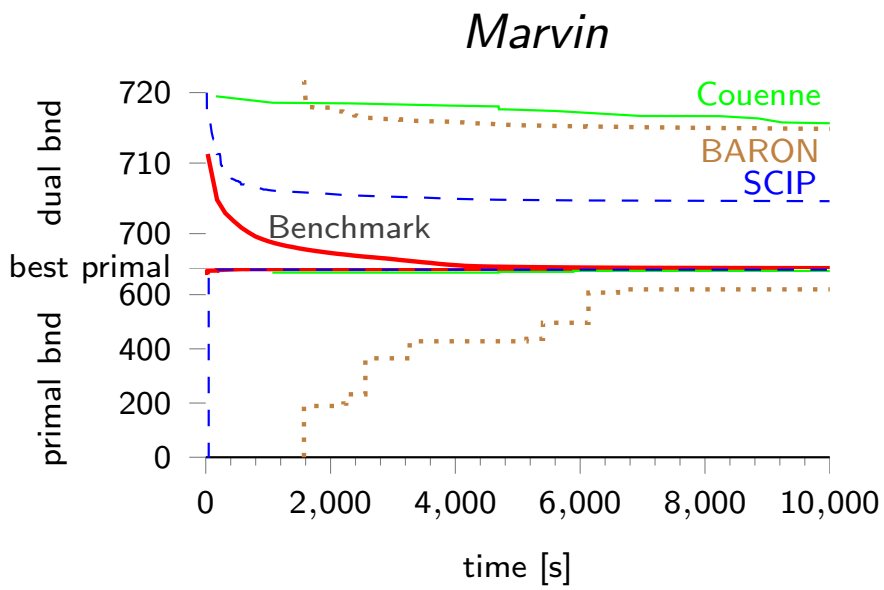
to the CPLEX-based implementation of Bley et al. [2009a],

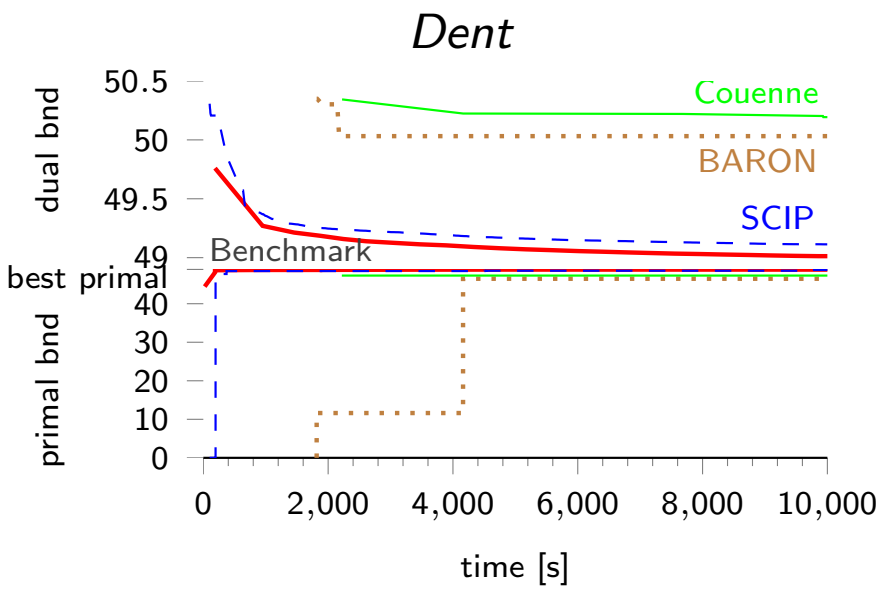
Compare the performance of

- BARON 9.0.2
- Couenne 0.2
- SCIP 2.0

to the CPLEX-based implementation of Bley et al. [2009a],
on 2 realistic mine production scheduling instances from industry:

| | *Marvin* | | | | *Dent* | | | |
|---|---|---|---|---|---|---|---|---|
| | no. variables | | no. constraints | | no. variables | | no. constraints | |
| | bin | cont | linear | quad | bin | cont | linear | quad |
| aggreg. model | 1445 | 4403 | 7582 | 16 | 3125 | 9475 | 15750 | 24 |
| ext. model | 1445 | 7242 | 9044 | 1360 | 3125 | 15650 | 18900 | 3000 |




- benchmark algo. closes gap up to 0.02% (Marvin) and 0.33% (Dent) after 10000 secs
- SCIP (with LP solver CPLEX) reaches 1.80% resp. 0.71%
- difference is also due to
  - underlying MIP solver (CPLEX vs. SCIP)
  - problem-specific MIP cuts/variable fixings

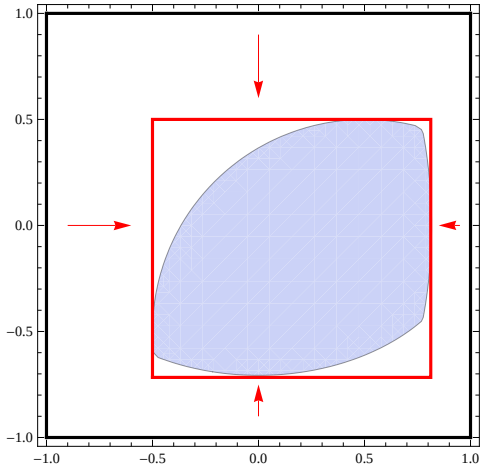
  i.e. not only/mainly due to the handling of the nonlinearities.

# Outline

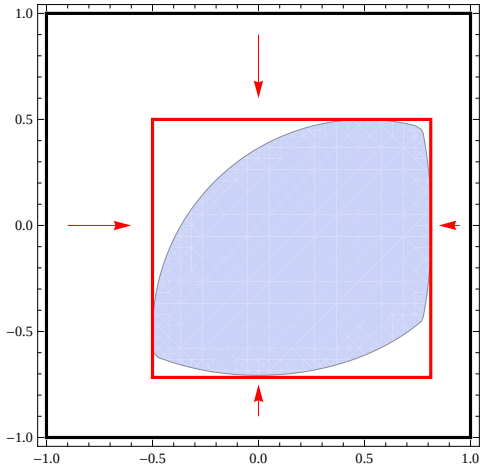
# Bounds Tightening (Domain Propagation)

Tighten variable bounds $[\underline{x}, \overline{x}]$ 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



# Bounds Tightening (Domain Propagation)



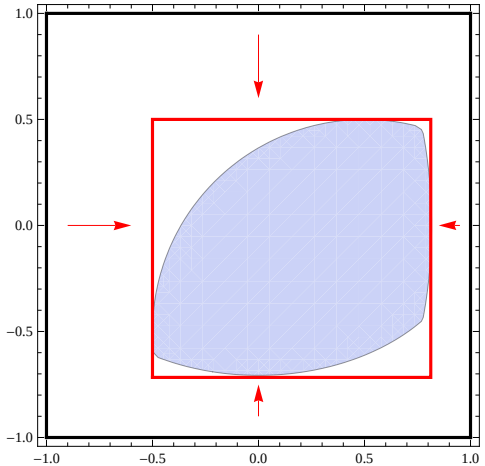

Tighten variable bounds $[\underline{x}, \overline{x}]$ such that

- 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

Bound tightening allows to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)

# Bounds Tightening (Domain Propagation)



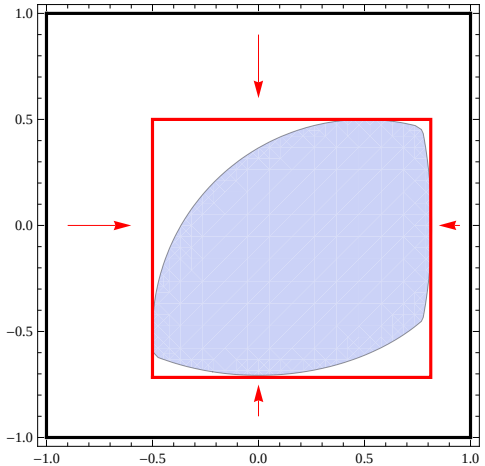

Tighten variable bounds $[\underline{x}, \overline{x}]$ such that

- 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

Bound tightening allows to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching

# Bounds Tightening (Domain Propagation)



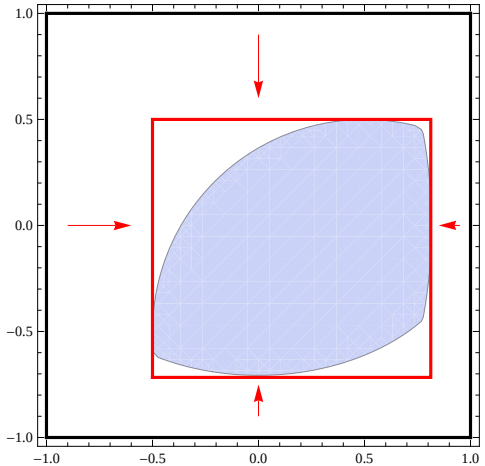

Tighten variable bounds $[\underline{x}, \overline{x}]$ such that

- 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

Bound tightening allows to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching
- very important: improve underestimators without branching $\Rightarrow$ tighter lower bounds

# Bounds Tightening (Domain Propagation)




Tighten variable bounds $[\underline{x}, \overline{x}]$ 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

Bound tightening allows to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching
- very important: improve underestimators without branching $\Rightarrow$ tighter lower bounds

Some techniques [Belotti et al., 2009]:

FBBT Feasibility-Based Bounds Tightening (cheap)

ABT Aggressive Feasibility-Based Bounds Tightening (expensive)

OBBT Optimality-Based Bounds Tightening (expensive)

# FBBT for Linear Constraints

Feasibility-Based Bound Tightening for a linear constraint:

$$\underline{b} \leq \sum_{i:a_i>0} a_i x_i + \sum_{i:a_i<0} a_i x_i \leq \overline{b},$$

Feasibility-Based Bound Tightening for a linear constraint:

$$\underline{b} \leq \sum_{i:a_i>0} a_i x_i + \sum_{i:a_i<0} a_i x_i \leq \overline{b},$$

$$\Rightarrow \quad x_j \leq \frac{1}{a_j} \begin{cases} \overline{b} - \sum_{i:a_i>0, i\neq j} a_i \underline{x}_i - \sum_{i:a_i<0} a_i \overline{x}_i, & \text{if } a_j > 0 \\ \underline{b} - \sum_{i:a_i>0} a_i \overline{x}_i - \sum_{i:a_i<0, i\neq j} a_i \underline{x}_i, & \text{if } a_j < 0 \end{cases}$$

$$x_j \geq \frac{1}{a_j} \begin{cases} \underline{b} - \sum_{i:a_i>0, i\neq j} a_i \overline{x}_i - \sum_{i:a_i<0} a_i \underline{x}_i, & \text{if } a_j > 0 \\ \overline{b} - \sum_{i:a_i>0} a_i \underline{x}_i - \sum_{i:a_i<0, i\neq j} a_i \overline{x}_i, & \text{if } a_j < 0 \end{cases}$$

# FBBT for Nonlinear Constraints

[Schichl and Neumaier, 2005, Vu et al., 2008]

Represent algebraic structure of problem in one directed acyclic graph:

- nodes: variables, operations, constraints
- arcs: flow of computation

# FBBT for Nonlinear Constraints

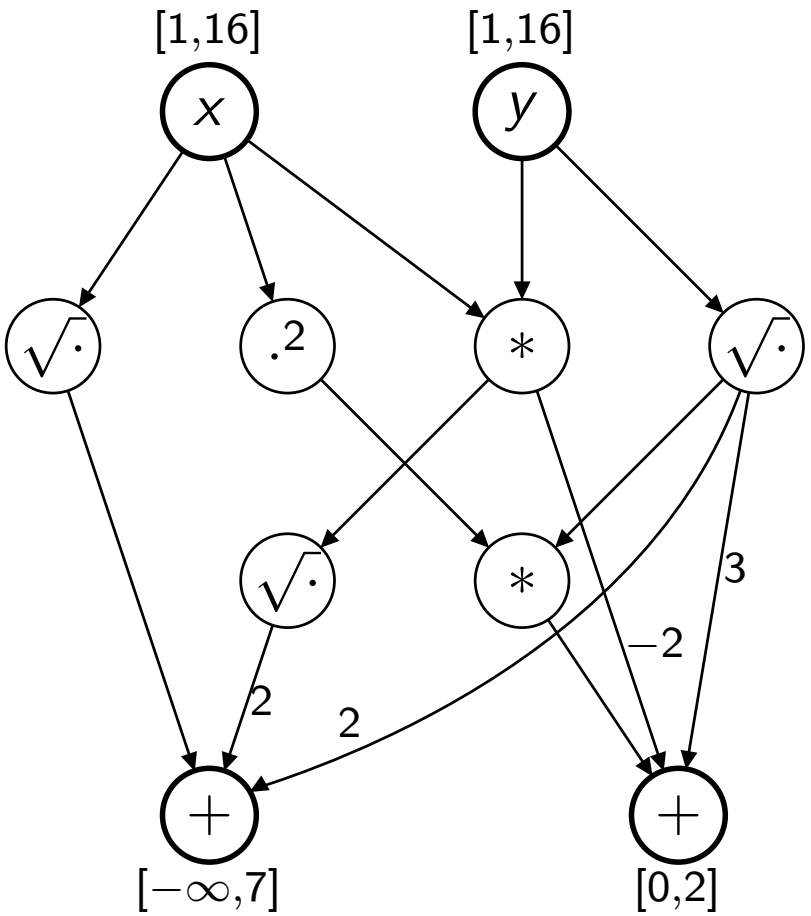
[Schichl and Neumaier, 2005, Vu et al., 2008]

Represent algebraic structure of problem in one directed acyclic graph:

- nodes: variables, operations, constraints
- arcs: flow of computation

Example:

$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$
$$x^2\sqrt{y} - 2xy + 3\sqrt{y} \in [0, 2]$$
$$x, y \in [1, 16]$$



# FBBT for Nonlinear Constraints

[Schichl and Neumaier, 2005, Vu et al., 2008]

Represent algebraic structure of problem in one directed acyclic graph:

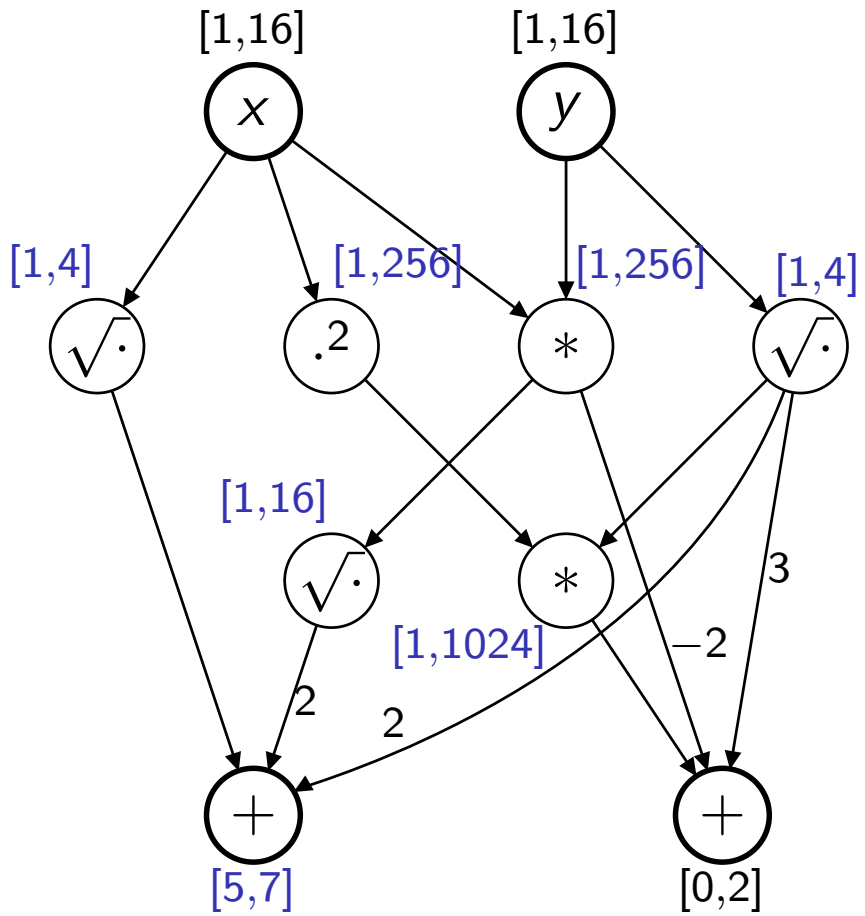
- nodes: variables, operations, constraints
- arcs: flow of computation

Example:

$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$
$$x^2\sqrt{y} - 2xy + 3\sqrt{y} \in [0, 2]$$
$$x, y \in [1, 16]$$

Forward propagation:

- compute bounds on intermediate nodes (top-down)



# FBBT for Nonlinear Constraints

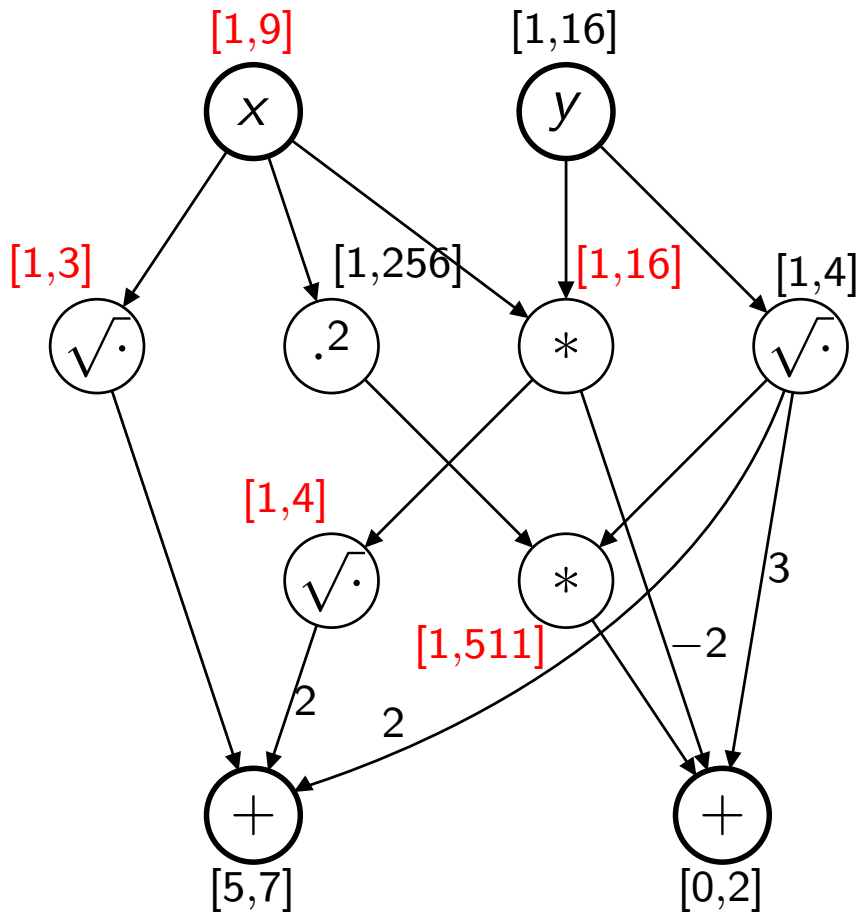
[Schichl and Neumaier, 2005, Vu et al., 2008]

Represent algebraic structure of problem in one directed acyclic graph:

- nodes: variables, operations, constraints
- arcs: flow of computation

Example:

$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$
$$x^2\sqrt{y} - 2xy + 3\sqrt{y} \in [0, 2]$$
$$x, y \in [1, 16]$$




Forward propagation:

- compute bounds on intermediate nodes (top-down)

Backward propagation:

- reduce bounds using reverse operations (bottom-up)

## Probing for MIP

- tentatively fix binary variable $x_i$ and apply domain propagation

## Probing for MIP

- tentatively fix binary variable $x_i$ and apply domain propagation
- if infeasibility is detected for $x_i = a$, then fix binary variable to $1 - a$ in main problem

## Probing for MIP

- tentatively fix binary variable $x_i$ and apply domain propagation
- if infeasibility is detected for $x_i = a$, then fix binary variable to $1 - a$ in main problem
- otherwise, use unification of bound tightenings for $x_i = 0$ and $x_i = 1$, deduce implications (used later for cut generation)

## Probing for MIP

- tentatively fix binary variable $x_i$ and apply domain propagation
- if infeasibility is detected for $x_i = a$, then fix binary variable to $1 - a$ in main problem
- otherwise, use unification of bound tightenings for $x_i = 0$ and $x_i = 1$, deduce implications (used later for cut generation)
- within strong branching: solve LP relaxation of tentative fixing $x_i = a$ and fix to $1 - a$ if lower bound exceeds upper bound

## Probing for MIP

- tentatively fix binary variable $x_i$ and apply domain propagation
- if infeasibility is detected for $x_i = a$, then fix binary variable to $1 - a$ in main problem
- otherwise, use unification of bound tightenings for $x_i = 0$ and $x_i = 1$, deduce implications (used later for cut generation)
- within strong branching: solve LP relaxation of tentative fixing $x_i = a$ and fix to $1 - a$ if lower bound exceeds upper bound

## Probing for MINLP [Tawarmalani and Sahinidis, 2004, Belotti et al., 2009]

- tentatively reduce bounds of a nonlinear variable, e.g., $x_i \leq 0.3\overline{x}_i$, and apply FBBT

## Probing for MIP

- tentatively fix binary variable $x_i$ and apply domain propagation
- if infeasibility is detected for $x_i = a$, then fix binary variable to $1 - a$ in main problem
- otherwise, use unification of bound tightenings for $x_i = 0$ and $x_i = 1$, deduce implications (used later for cut generation)
- within strong branching: solve LP relaxation of tentative fixing $x_i = a$ and fix to $1 - a$ if lower bound exceeds upper bound

## Probing for MINLP [Tawarmalani and Sahinidis, 2004, Belotti et al., 2009]

- tentatively reduce bounds of a nonlinear variable, e.g., $x_i \leq 0.3\overline{x}_i$, and apply FBBT
- if infeasibility is detected, tighten bound in main problem, e.g., $\underline{x}_i := 0.3\overline{x}_i$

# Aggressive Bound Tightening

## Probing for MIP

- tentatively fix binary variable $x_i$ and apply domain propagation
- if infeasibility is detected for $x_i = a$, then fix binary variable to $1 - a$ in main problem
- otherwise, use unification of bound tightenings for $x_i = 0$ and $x_i = 1$, deduce implications (used later for cut generation)
- within strong branching: solve LP relaxation of tentative fixing $x_i = a$ and fix to $1 - a$ if lower bound exceeds upper bound

## Probing for MINLP [Tawarmalani and Sahinidis, 2004, Belotti et al., 2009]

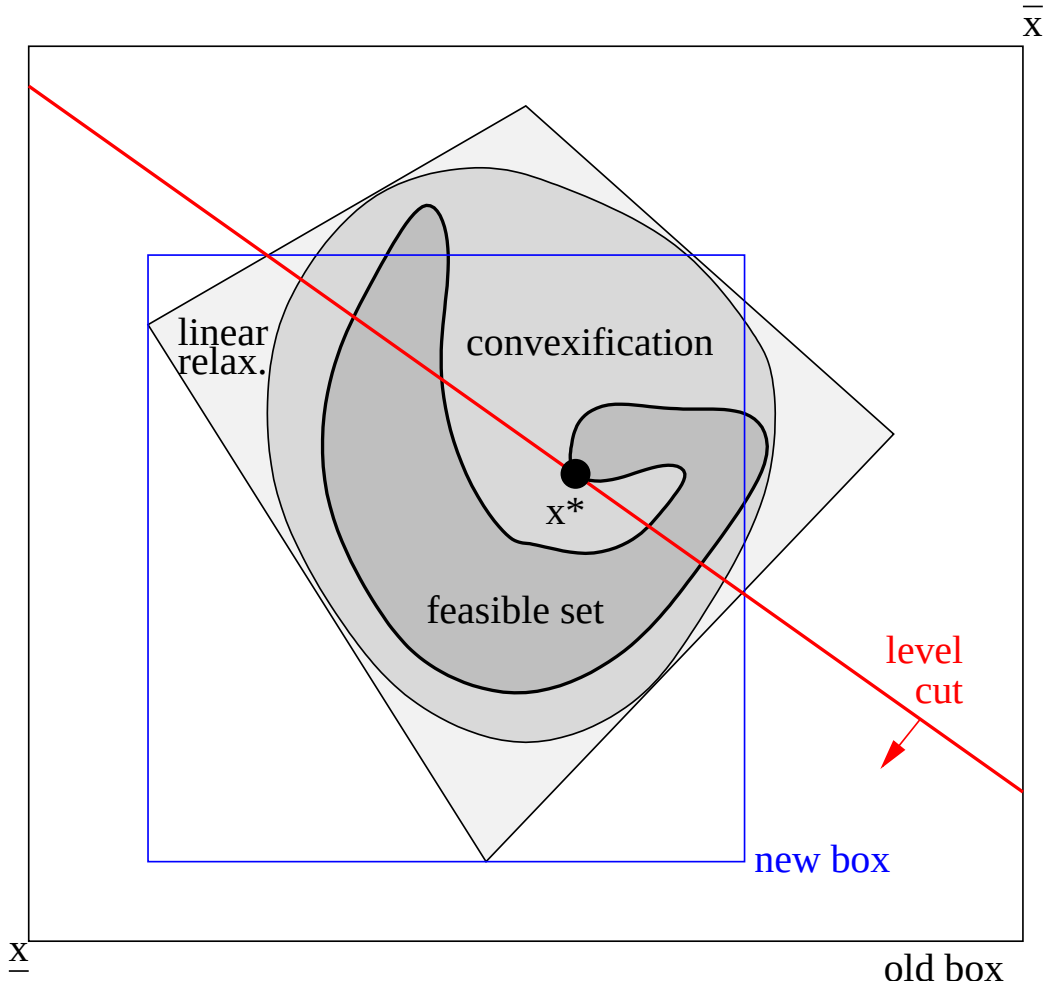
- tentatively reduce bounds of a nonlinear variable, e.g., $x_i \leq 0.3\overline{x}_i$, and apply FBBT
- if infeasibility is detected, tighten bound in main problem, e.g., $\underline{x}_i := 0.3\overline{x}_i$
- within B&B: update and solve LP relaxation to obtain lower bound for tentative reduction

# Optimality-Based Bound Tightening

- Consider a linear relaxation with constraints $Ax \leq b$.
- Let $x^*$ be the best solution found so far.

$$\underline{x}_i := \min x_i \quad \text{s.t. } Ax \leq b, \quad c^T x \leq c^T x^*$$

$$\overline{x}_i := \max x_i \quad \text{s.t. } Ax \leq b, \quad c^T x \leq c^T x^*$$



# Bound Tightening Methods: Computational Study

## Effect of disabling FBBT in SCIP on 89 MIQQPs [Berthold et al., 2010]

- number of instances solved: $-1$
- mean running time: $+6\%$
- time to first / optimal solution: $+13\%$ / $+7\%$
- number of nodes: $+39\%$

# Bound Tightening Methods: Computational Study

## Effect of disabling FBBT in SCIP on 89 MIQQPs [Berthold et al., 2010]

- number of instances solved: $-1$
- mean running time: $+6\%$
- time to first / optimal solution: $+13\%$ / $+7\%$
- number of nodes: $+39\%$

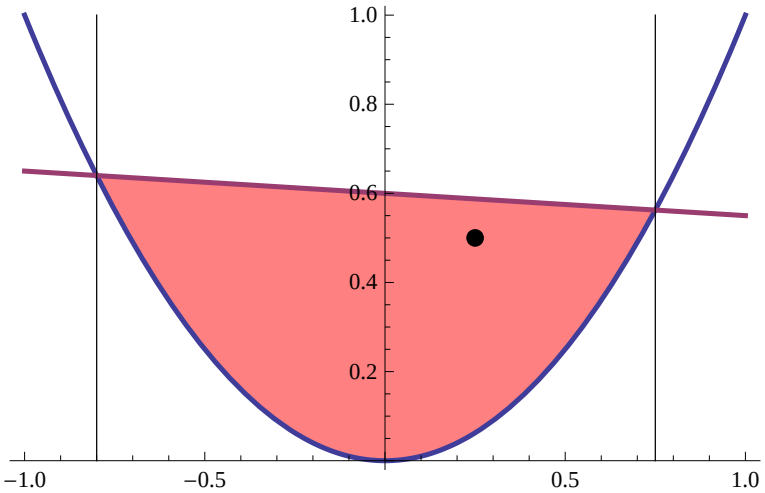
## Effect of bound tight. in Couenne on 21 (MI)NLPs [Belotti et al., 2009]

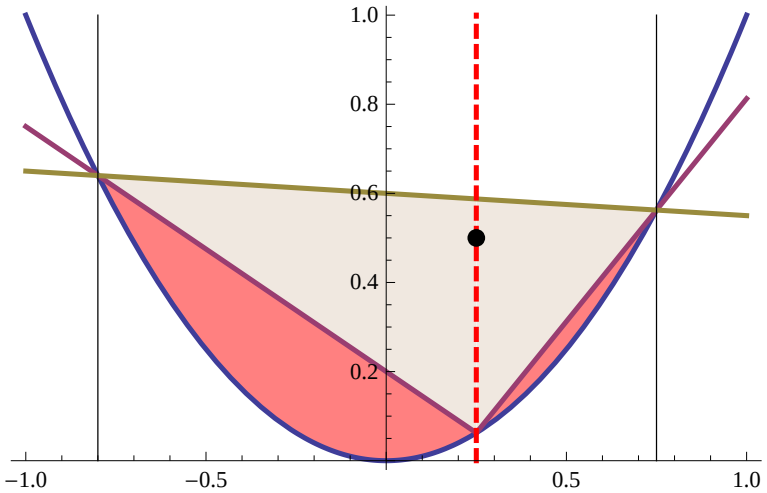
| Method | #solved | #best time | #best nodes | #best depth |
|---|---|---|---|---|
| FBBT + ABT + OBBT* | 19 | 12 | 14 | 16 |
| FBBT only | 16 | 2 | 4 | 6 |
| FBBT + ABT** | 14 | 1 | 3 | 6 |
| no bound tightening | 16 | 10 | 7 | 10 |
| FBBT + OBBT*** | 16 | 2 | 8 | 9 |

* "tuned" = apply ABT and OBBT at root or with prob. $2^{1-\text{depth}}$ and $2^{-\text{depth}}$, resp.
** apply ABT at root or with prob. $2^{3-\text{depth}}$, if depth $> 3$
*** apply OBBT at root or with prob. $2^{3-\text{depth}}$, if depth $> 3$

Branching in MINLP:

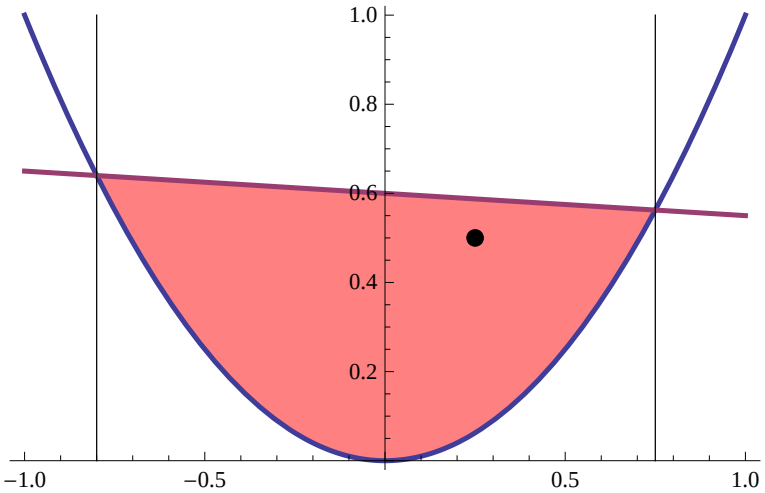
- on an integer variable to resolve fractionality
- on a nonlinear variable in a nonconvex term to allow tighter relaxation

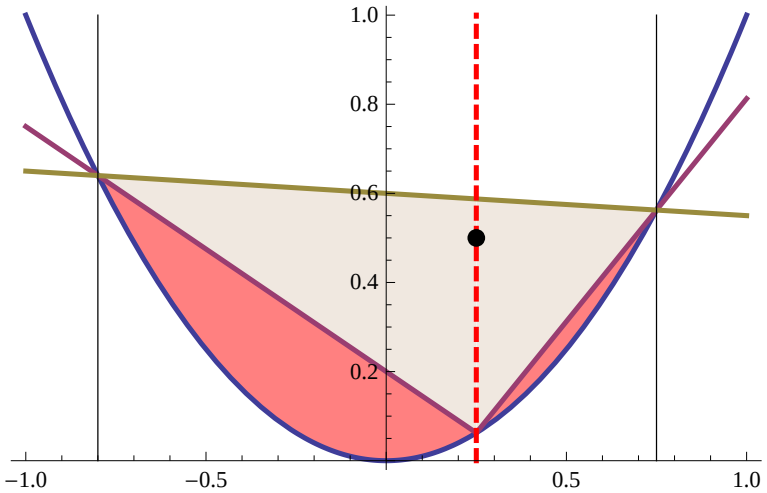


Branching in MINLP:

- on an integer variable to resolve fractionality
- on a nonlinear variable in a nonconvex term to allow tighter relaxation




**How to select the branching variable from a set of variable candidates?**

- give priority to integer variables
  apply MIP branching techniques (most fractional, reliability branching, strong branching, ...)
- extend these branching rules to continuous variables [Belotti et al., 2009]

# Branching Rule: Most violated

## "Most fractional" rule for integer variables

Branch on $x_{i^*}$, such that

$$i^* = \text{argmax}_{i \in I} \ \min\{\hat{x}_i - \lfloor \hat{x}_i \rfloor, \lceil \hat{x}_i \rceil - \hat{x}_i\},$$

where $\hat{x}$ is the current solution of the LP relaxation.

## "Most fractional" rule for integer variables

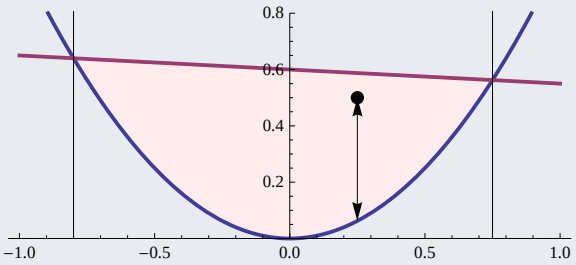
Branch on $x_{i^*}$, such that

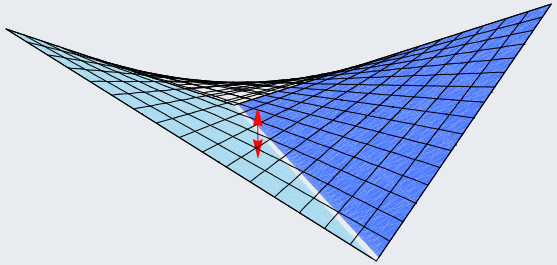
$$i^* = \text{argmax}_{i \in I} \ \min\{\hat{x}_i - \lfloor \hat{x}_i \rfloor, \lceil \hat{x}_i \rceil - \hat{x}_i\},$$

where $\hat{x}$ is the current solution of the LP relaxation.

## "Most violated" rule for nonlinear variables [Belotti et al., 2009]

Assign to each variable $x_i$ the set of "convexification gaps" $U_{i,j}(\hat{x})$ in $\hat{x}$ for all nonconvex terms $j$ in which $x_i$ is involved



Branch on $x_{i^*}$, such that

$$i^* = \text{argmax}_{i \in I} \ \frac{1}{10} \sum_j U_{i,j}(\hat{x}) + \frac{13}{10} \max_j U_{i,j}(\hat{x}) + \frac{8}{10} \min_j U_{i,j}(\hat{x})$$

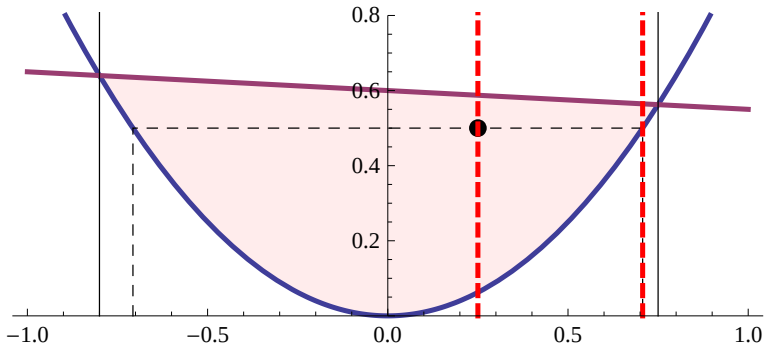
# Branching Rule: Violation transfer

[Tawarmalani and Sahinidis, 2004, Belotti et al., 2009]

Given $(\hat{x}, \hat{\pi})$ primal-dual solution of LP relaxation $Ax \leq b$, $A = (a_{ji})_{i,j}$.

For $x_i$, let $[\underline{\underline{x}}_i, \overline{\overline{x}}_i] \subseteq [\underline{x}_i, \overline{x}_i]$, s.t.

- $\hat{x}_j \in [\underline{x}_i, \overline{x}_i]$, and
- for every $x_j$, $x_j = f(x_i)$ (for some $f$), $\exists v \in [\underline{\underline{x}}_i, \overline{\overline{x}}_i] : f(v) = \hat{x}_j$, and
- as small as possible




$[\underline{\underline{x}}_i, \overline{\overline{x}}_i] = [0.25, 0.7]$

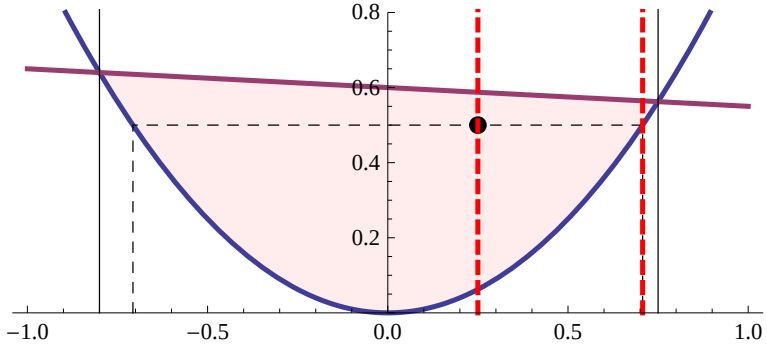
# Branching Rule: Violation transfer

[Tawarmalani and Sahinidis, 2004, Belotti et al., 2009]

Given $(\hat{x}, \hat{\pi})$ primal-dual solution of LP relaxation $Ax \leq b$, $A = (a_{ji})_{i,j}$.

For $x_i$, let $[\underline{\underline{x}}_i, \overline{\overline{x}}_i] \subseteq [\underline{x}_i, \overline{x}_i]$, s.t.

- $\hat{x}_j \in [\underline{x}_i, \overline{x}_i]$, and
- for every $x_j$, $x_j = f(x_i)$ (for some $f$), $\exists v \in [\underline{\underline{x}}_i, \overline{\overline{x}}_i] : f(v) = \hat{x}_j$, and
- as small as possible




$[\underline{\underline{x}}_i, \overline{\overline{x}}_i] = [0.25, 0.7]$

Consider segment $\quad X_i := \{x : x_j = \hat{x}_j, j \neq i,\ x_i \in [\underline{\underline{x}}_i, \overline{\overline{x}}_i]\}$.

Estimate change in lower bound by range of Lagrange function of LP relaxation w.r.t. $X_i$:

$$\max_{x \in X_R^i} \mathcal{L}(x) - \min_{x \in X_R^i} \mathcal{L}(x) = |\overline{\overline{x}}_i - \underline{\underline{x}}_i| \, |\textstyle\sum_j \hat{\pi}_j a_{ji}|.$$

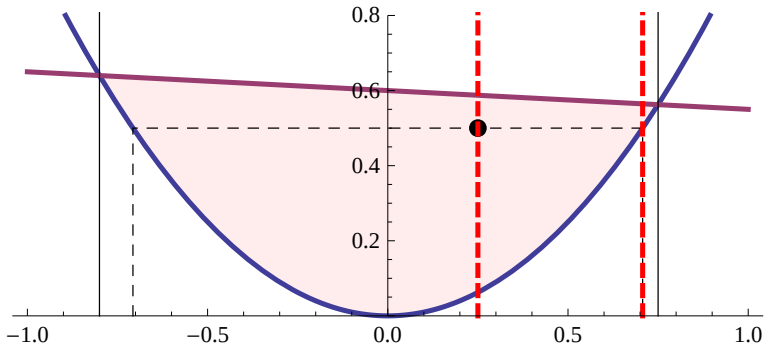
# Branching Rule: Violation transfer

[Tawarmalani and Sahinidis, 2004, Belotti et al., 2009]

Given $(\hat{x}, \hat{\pi})$ primal-dual solution of LP relaxation $Ax \leq b$, $A = (a_{ji})_{i,j}$.

For $x_i$, let $[\underline{\underline{x}}_i, \overline{\overline{x}}_i] \subseteq [\underline{x}_i, \overline{x}_i]$, s.t.

- $\hat{x}_j \in [\underline{x}_i, \overline{x}_i]$, and
- for every $x_j$, $x_j = f(x_i)$ (for some $f$), $\exists v \in [\underline{\underline{x}}_i, \overline{\overline{x}}_i] : f(v) = \hat{x}_j$, and
- as small as possible




$[\underline{\underline{x}}_i, \overline{\overline{x}}_i] = [0.25, 0.7]$

Consider segment $\quad X_i := \{x : x_j = \hat{x}_j, j \neq i,\ x_i \in [\underline{\underline{x}}_i, \overline{\overline{x}}_i]\}$.

Estimate change in lower bound by range of Lagrange function of LP relaxation w.r.t. $X_i$:

$$\max_{x \in X_R^i} \mathcal{L}(x) - \min_{x \in X_R^i} \mathcal{L}(x) = |\overline{\overline{x}}_i - \underline{\underline{x}}_i| \, |\textstyle\sum_j \hat{\pi}_j a_{ji}|.$$

Branch on $x_{i^*}$, such that

$$i^* = \text{argmax}_{i \in I} \, |\overline{\overline{x}}_i - \underline{\underline{x}}_i| \, \textstyle\sum_j |\hat{\pi}_j a_{ji}|.$$

# Branching Rule: Pseudo Costs

**Integer variables $x_i$, $i \in I$**

- ▷ when branching, memorize resulting change in lower bound
- ▷ average bound improvements over all down- and up-branches of $x_i$ so far

$\Rightarrow$ estimates $\psi_i^-$, $\psi_i^+$ for lower bound improvement per unit of change in $x_i$

# Branching Rule: Pseudo Costs

**Integer variables $x_i$, $i \in I$**

- when branching, memorize resulting change in lower bound
- average bound improvements over all down- and up-branches of $x_i$ so far

$\Rightarrow$ estimates $\psi_i^-$, $\psi_i^+$ for lower bound improvement per unit of change in $x_i$

- branch on $x_i$ that has fractional value in LP solution $\hat{x}$ and maximizes
  $\alpha\,\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor) + (1-\alpha)\,\psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$ or $\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor) \cdot \psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$

# Branching Rule: Pseudo Costs

## Integer variables $x_i$, $i \in I$

- ▷ when branching, memorize resulting change in lower bound
- ▷ average bound improvements over all down- and up-branches of $x_i$ so far

⇒ estimates $\psi_i^-$, $\psi_i^+$ for lower bound improvement per unit of change in $x_i$

- ▷ branch on $x_i$ that has fractional value in LP solution $\hat{x}$ and maximizes
  $\alpha\,\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor) + (1-\alpha)\,\psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$ or $\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor) \cdot \psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$

## Continuous variables in MINLP

- ▷ branching does not result in immediate change of variable's value in LP

⇒ cannot estimate bound changes as $\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor)$, $\psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$

# Branching Rule: Pseudo Costs

## Integer variables $x_i$, $i \in I$

- when branching, memorize resulting change in lower bound
- average bound improvements over all down- and up-branches of $x_i$ so far

$\Rightarrow$ estimates $\psi_i^-$, $\psi_i^+$ for lower bound improvement per unit of change in $x_i$

- branch on $x_i$ that has fractional value in LP solution $\hat{x}$ and maximizes
  $\alpha\,\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor) + (1-\alpha)\,\psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$ or $\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor) \cdot \psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$

## Continuous variables in MINLP

- branching does not result in immediate change of variable's value in LP

$\Rightarrow$ cannot estimate bound changes as $\psi_i^-(\hat{x}_i - \lfloor \hat{x}_i \rfloor)$, $\psi_i^+(\lceil \hat{x}_i \rceil - \hat{x}_i)$

- Belotti et al. [2009] proposed several alternatives:
  - multiply pseudocosts by variable infeasibility (analog to fractionality):
    $$\psi_i^{-,+}\left(\tfrac{1}{10}\textstyle\sum_j U_{i,j}(\hat{x}) + \tfrac{13}{10}\max_j U_{i,j}(\hat{x}) + \tfrac{8}{10}\min_j U_{i,j}(\hat{x})\right)$$
  - multiply by domain width after branching:
    $$\psi_i^-(\hat{x}_i - \underline{x}_i),\ \psi_i^+(\overline{x}_i - \hat{x}_i) \quad \text{or} \quad \psi_i^-(\overline{x}_i - \hat{x}_i),\ \psi_i^+(\hat{x}_i - \underline{x}_i)$$
  - multiply by width of intervals from violation transfer:
    $$\psi_i^-(\hat{x}_i - \underline{\underline{x}}_i),\ \psi_i^+(\overline{\overline{x}}_i - \hat{x}_i)$$

# Branching Rule: Reliability Branching

## Strong Branching

- at each node, compute lower bound for all (or many) possible branchings (i.e., construct two branches, update and solve relaxation) and choose the one with best bound improvement
- expensive, but best reduction in number of nodes

# Branching Rule: Reliability Branching

## Strong Branching

- at each node, compute lower bound for all (or many) possible branchings (i.e., construct two branches, update and solve relaxation) and choose the one with best bound improvement
- expensive, but best reduction in number of nodes

## Reliability branching

- compute exact bound improvement only for variables with a low number of branchings so far
- otherwise, assume pseudo costs are reliable and use them to evaluate potential bound improvement

$\Rightarrow$ initialization of pseudo costs by strong branching

Effect of branching techniques in Couenne on 33 (MI)NLPs [Belotti et al., 2009]:

| Method | #solved | #best time | #best nodes | #largest depth |
|---|---|---|---|---|
| variable infeasibility | 23 | 14 | 7 | 15 |
| violation transfer | 26 | 13 | 4 | 20 |
| rel.br.-var.infeas.* | 23 | 11 | 4 | 8 |
| rel.br.-dom.width** | 25 | 9 | 5 | 8 |
| strong branching | 25 | 1 | 17 | 11 |

* "rel.br.-infeas." = reliability branching with variable infeasibilities as pseudo costs multipliers

** "rel.br.-width" = reliability branching with domain width after branching as pseudo costs multipliers: $\psi_i^-(\overline{x}_i - \hat{x}_i)$, $\psi_i^+(\hat{x}_i - \underline{x}_i)$

# Reformulations for MIQCPs: SOC

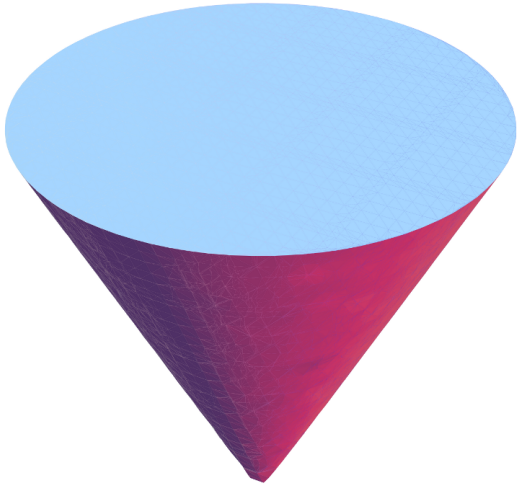
Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0$$

with $\alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0$ describe a convex feasible region:

$$\sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1}} x_{N+1}$$

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



feasible region

"ice cream cone"

# Reformulations for MIQCPs: SOC

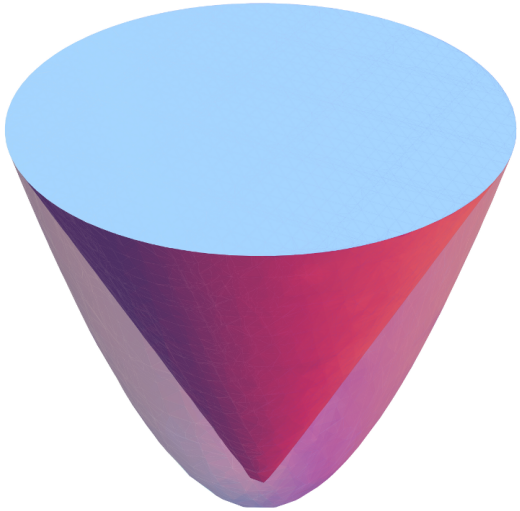
Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0$$

with $\alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0$ describe a convex feasible region:

$$\sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1}} x_{N+1}$$

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



not recognizing SOC:
using termwise underestimator:

$$\left\{ \begin{array}{c} x^2 + y^2 + w \leq 1 \\ \frac{\underline{z}^2 - \overline{z}^2}{\overline{z} - \underline{z}}(z - \underline{z}) - \underline{z}^2 \leq w \end{array} \right\}$$

# Reformulations for MIQCPs: SOC

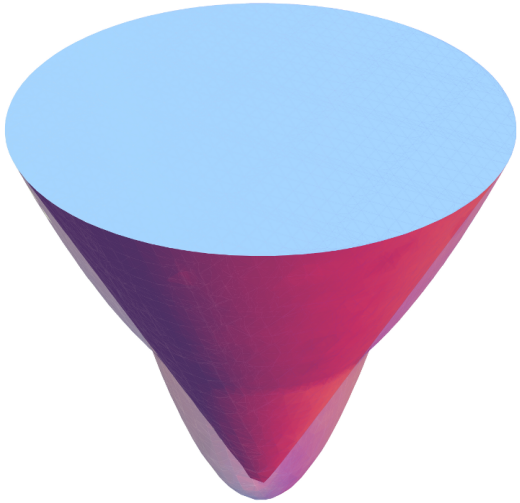
Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0$$

with $\alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0$ describe a convex feasible region:

$$\sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1}} x_{N+1}$$

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



not recognizing SOC:
using termwise underestimator:

$$\left\{ \begin{array}{c} x^2 + y^2 + w \leq 1 \\ \frac{\underline{z}^2 - \overline{z}^2}{\overline{z} - \underline{z}}(z - \underline{z}) - \underline{z}^2 \leq w \end{array} \right\}$$

after branching on $z = 0.5$

# Reformulations for MIQCPs: SOC

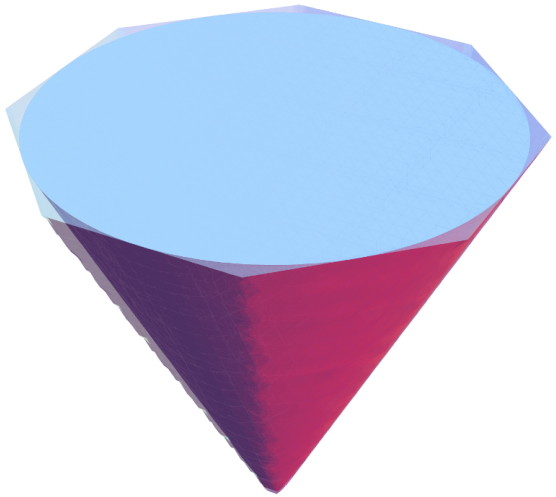
Quadratic constraints of the form

$$\sum_{k=1}^{N} \alpha_k x_k^2 - \alpha_{N+1} x_{N+1}^2 \leq 0$$

with $\alpha_1, \ldots, \alpha_{N+1} \geq 0, L_{N+1} \geq 0$ describe a convex feasible region:

$$\sqrt{\sum_{k=1}^{N} \alpha_k x_k^2} \leq \sqrt{\alpha_{N+1}} x_{N+1}$$

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



recognizing SOC:

initial relaxation
no spatial branching necessary

# Reformulations for MIQCPs: SOC

More general [Mahajan and Munson, 2010]:

- Given a quadratic constraint $x^T Ax + c^T x + d \leq 0$.

# Reformulations for MIQCPs: SOC

More general [Mahajan and Munson, 2010]:

- Given a quadratic constraint $x^T Ax + c^T x + d \leq 0$.
- Let $A = QDQ^T$ with $Q$ orthogonal and $D$ diagonal matrix.
- Let $D = RER$, $b = R^{-1}Q^T c$ with $E_{ii} = \text{sign}(D_{ii})$, $R_{ii} = \begin{cases} \sqrt{|D_{ii}|}, & \text{if } D_{ii} \neq 0, \\ 1, & \text{if } D_{ii} = 0. \end{cases}$
- Then constraint is equivalent to

$$\begin{aligned} y^T Ey + b^T y = d &\leq 0 \\ RQ^T x &= y \end{aligned}$$

# Reformulations for MIQCPs: SOC

More general [Mahajan and Munson, 2010]:

- Given a quadratic constraint $x^T Ax + c^T x + d \leq 0$.
- Let $A = QDQ^T$ with $Q$ orthogonal and $D$ diagonal matrix.
- Let $D = RER$, $b = R^{-1}Q^T c$ with $E_{ii} = \text{sign}(D_{ii})$, $R_{ii} = \begin{cases} \sqrt{|D_{ii}|}, & \text{if } D_{ii} \neq 0, \\ 1, & \text{if } D_{ii} = 0. \end{cases}$
- Then constraint is equivalent to

$$\sum_{i:E_{ii}>0} (y_i^2 + b_i y_i) - \sum_{i:E_{ii}<0} (y_i^2 - b_i y_i) + \sum_{i:E_{ii}=0} (b_i y_i) + d \leq 0$$

$$RQ^T x = y$$

# Reformulations for MIQCPs: SOC

More general [Mahajan and Munson, 2010]:

- Given a quadratic constraint $x^T Ax + c^T x + d \leq 0$.
- Let $A = QDQ^T$ with $Q$ orthogonal and $D$ diagonal matrix.
- Let $D = RER$, $b = R^{-1}Q^T c$ with $E_{ii} = \text{sign}(D_{ii})$, $R_{ii} = \begin{cases} \sqrt{|D_{ii}|}, & \text{if } D_{ii} \neq 0, \\ 1, & \text{if } D_{ii} = 0. \end{cases}$
- Then constraint is equivalent to

$$\sum_{i:E_{ii}>0} \left(y_i + \frac{b_i}{2}\right)^2 - \sum_{i:E_{ii}<0} \left(y_i - \frac{b_i}{2}\right)^2 + \sum_{i:E_{ii}=0} (b_i y_i) + d - \frac{1}{4} \sum_i \text{sign}(D_{ii}) b_i^2 \leq 0$$

$$RQ^T x = y$$

# Reformulations for MIQCPs: SOC

More general [Mahajan and Munson, 2010]:

- Given a quadratic constraint $x^T Ax + c^T x + d \leq 0$.
- Let $A = QDQ^T$ with $Q$ orthogonal and $D$ diagonal matrix.
- Let $D = RER$, $b = R^{-1}Q^T c$ with $E_{ii} = \text{sign}(D_{ii})$, $R_{ii} = \begin{cases} \sqrt{|D_{ii}|}, & \text{if } D_{ii} \neq 0, \\ 1, & \text{if } D_{ii} = 0. \end{cases}$
- Then constraint is equivalent to

$$\sum_{i:E_{ii}>0} \left(y_i + \frac{b_i}{2}\right)^2 - \sum_{i:E_{ii}<0} \left(y_i - \frac{b_i}{2}\right)^2 + \sum_{i:E_{ii}=0} (b_i y_i) + d - \frac{1}{4} \sum_i \text{sign}(D_{ii}) b_i^2 \leq 0$$

$$RQ^T x = y$$

- If $|\{i : E_{ii} < 0\}| = 0$, then quadratic function is convex.

# Reformulations for MIQCPs: SOC

More general [Mahajan and Munson, 2010]:

- Given a quadratic constraint $x^T Ax + c^T x + d \leq 0$.
- Let $A = QDQ^T$ with $Q$ orthogonal and $D$ diagonal matrix.
- Let $D = RER$, $b = R^{-1}Q^T c$ with $E_{ii} = \text{sign}(D_{ii})$, $R_{ii} = \begin{cases} \sqrt{|D_{ii}|}, & \text{if } D_{ii} \neq 0, \\ 1, & \text{if } D_{ii} = 0. \end{cases}$
- Then constraint is equivalent to

$$\sum_{i:E_{ii}>0} \left(y_i + \frac{b_i}{2}\right)^2 - \sum_{i:E_{ii}<0} \left(y_i - \frac{b_i}{2}\right)^2 + \sum_{i:E_{ii}=0} (b_i y_i) + d - \frac{1}{4}\sum_i \text{sign}(D_{ii}) b_i^2 \leq 0$$

$$RQ^T x = y$$

- If $|\{i : E_{ii} < 0\}| = 0$, then quadratic function is convex.
- If $|\{i : E_{ii} < 0\}| = 1$ (w.l.o.g. $E_{11} < 0$) and $h := d - \frac{1}{4}\sum_i \text{sign}(D_{ii}) b_i^2 \geq 0$, then constraint is equivalent to

$$\left\| \begin{pmatrix} E_+(y + \frac{b}{2}) \\ \sqrt{h} \end{pmatrix} \right\|_2 \leq \left| y_1 - \frac{b_1}{2} \right|,$$

  where $E$ is diagonal matrix with $E_{ii}^+ = \max(0, E_{ii})$.
- Thus, constraint is equivalent to a union of two second order cones.

# Reformulations for MIQCPs: Convexify 0-1 QCPs

Assume $x \in \{0,1\}^n$ and consider the quadratic function

$$x^T A x, \quad \text{with} \quad A \not\succeq 0.$$

Eigenvalue shifting: For $\alpha \leq \lambda_1(A)$ (minimal eigenvalue of $A$),

$$(A - \alpha I) \succeq 0.$$

# Reformulations for MIQCPs: Convexify 0-1 QCPs

Assume $x \in \{0,1\}^n$ and consider the quadratic function

$$x^T A x, \quad \text{with} \quad A \not\succeq 0.$$

Eigenvalue shifting: For $\alpha \leq \lambda_1(A)$ (minimal eigenvalue of $A$),

$$(A - \alpha I) \succeq 0.$$

Reformulate using convex quadratic function: Since $x_i^2 = x_i$,

$$x^T A x = x^T (A - \alpha I) x + \alpha \sum_i x_i.$$

- CPLEX uses this simple trick

# Reformulations for MIQCPs: Convexify 0-1 QCPs

Assume $x \in \{0,1\}^n$ and consider the quadratic function

$$x^T A x, \quad \text{with} \quad A \not\succeq 0.$$

Eigenvalue shifting: For $\alpha \leq \lambda_1(A)$ (minimal eigenvalue of $A$),

$$(A - \alpha I) \succeq 0.$$

Reformulate using convex quadratic function: Since $x_i^2 = x_i$,

$$x^T A x = x^T (A - \alpha I) x + \alpha \sum_i x_i.$$

- CPLEX uses this simple trick
- it is not always the best strategy:

| instance | lower bound after 1 hour |
|---|---|
| qap | -1171771 |
| pb302035 | -13223300 |
| pb351535 | -9870063 |

# Reformulations for MIQCPs: Convexify 0-1 QCPs

Assume $x \in \{0,1\}^n$ and consider the quadratic function

$$x^T A x, \quad \text{with} \quad A \not\succeq 0.$$

Eigenvalue shifting: For $\alpha \leq \lambda_1(A)$ (minimal eigenvalue of $A$),

$$(A - \alpha I) \succeq 0.$$

Reformulate using convex quadratic function: Since $x_i^2 = x_i$,

$$x^T A x = x^T (A - \alpha I) x + \alpha \sum_i x_i.$$

- CPLEX uses this simple trick
- it is not always the best strategy:

| instance | lower bound after 1 hour |
|---|---|
| qap | -1171771 |
| pb302035 | -13223300 |
| pb351535 | -9870063 |

BUT: the trivial lower bound is 0.0! ($A$ has only positive entries)

# Reformulations for MIQCPs: Convexify 0-1 QCPs

Assume $x \in \{0,1\}^n$ and consider the quadratic function

$$x^T A x, \quad \text{with} \quad A \not\succeq 0.$$

Eigenvalue shifting: For $\alpha \leq \lambda_1(A)$ (minimal eigenvalue of $A$),

$$(A - \alpha I) \succeq 0.$$

Reformulate using convex quadratic function: Since $x_i^2 = x_i$,

$$x^T A x = x^T (A - \alpha I) x + \alpha \sum_i x_i.$$

- CPLEX uses this simple trick
- it is not always the best strategy:

| instance | lower bound after 1 hour | SCIP lower bound |
|---|---|---|
| qap | -1171771 | 79071 |
| pb302035 | -13223300 | 1151477 |
| pb351535 | -9870063 | 1786035 |

BUT: the trivial lower bound is 0.0! ($A$ has only positive entries)

# Reformulations for MIQCPs: Convexify 0-1 QCPs

Assume $x \in \{0,1\}^n$ and consider the quadratic function

$$x^T A x, \quad \text{with} \quad A \not\succeq 0.$$

Eigenvalue shifting: For $\alpha \leq \lambda_1(A)$ (minimal eigenvalue of $A$),

$$(A - \alpha I) \succeq 0.$$

Reformulate using convex quadratic function: Since $x_i^2 = x_i$,

$$x^T A x = x^T (A - \alpha I) x + \alpha \sum_i x_i.$$

- CPLEX uses this simple trick
- it is not always the best strategy:

| instance | lower bound after 1 hour | SCIP lower bound | best upper bound |
|---|---|---|---|
| qap | -1171771 | 79071 | 388214 |
| pb302035 | -13223300 | 1151477 | 3379359 |
| pb351535 | -9870063 | 1786035 | 3599602 |

BUT: the trivial lower bound is 0.0! ($A$ has only positive entries)

# Reformulations for MIQCPs: Products with 0-1 Variables

A quadratic term

$$x \cdot \sum_{k=1}^{N} a_k y_k \quad \text{with} \quad x \in \{0, 1\}$$

can equivalently be replaced by

- an auxiliary variable $w$
- and the additional linear constraints

$$M^L x \leq w \leq M^U x,$$

$$\sum_{k=1}^{N} a_k y_k - M^U(1-x) \leq w \leq \sum_{k=1}^{N} a_k y_k - M^L(1-x),$$

where $M^L$ and $M^U$ are bounds on $\sum_{k=1}^{N} a_k y_k$.

$\Rightarrow$ linear reformulation

# Reformulations for MIQCPs: Products with 0-1 Variables

A quadratic term

$$x \cdot \sum_{k=1}^{N} a_k y_k \quad \text{with} \quad x \in \{0,1\}$$

can equivalently be replaced by

- ▷ an auxiliary variable $w$
- ▷ and the additional linear constraints

$$M^L x \leq w \leq M^U x,$$

$$\sum_{k=1}^{N} a_k y_k - M^U(1-x) \leq w \leq \sum_{k=1}^{N} a_k y_k - M^L(1-x),$$

where $M^L$ and $M^U$ are bounds on $\sum_{k=1}^{N} a_k y_k$.

$\Rightarrow$ linear reformulation

Products of $n$ binary variables $\Rightarrow$ Pseudo-Boolean optimization

- ▷ linearize dynamically [Berthold et al., 2009b]

Tighter relaxations: E.g.,

- better handling of quadratic constraints: lift-and-project [Saxena et al., 2010], multiterm relaxations [Bao et al., 2009]
- convexification of trilinear $(x \cdot y \cdot z)$, quadrilinear $(w \cdot x \cdot y \cdot z)$, and multilinear forms: [Meyer and Floudas, 2004, Cafieri et al., 2010, Luedtke et al., 2010, Belotti et al., 2010]
- linear inequalities for orthogonal disjunctions and bilinear covering sets $(\sum_i a_i x_i y_i + b_i x_i + c_i y_i \geq r)$: [Tawarmalani et al., 2008]
- convexification of $n-1$–convex functions: [Jach et al., 2008]

# Further topics

Tighter relaxations: E.g.,

- better handling of quadratic constraints: lift-and-project [Saxena et al., 2010], multiterm relaxations [Bao et al., 2009]
- convexification of trilinear $(x \cdot y \cdot z)$, quadrilinear $(w \cdot x \cdot y \cdot z)$, and multilinear forms: [Meyer and Floudas, 2004, Cafieri et al., 2010, Luedtke et al., 2010, Belotti et al., 2010]
- linear inequalities for orthogonal disjunctions and bilinear covering sets $(\sum_i a_i x_i y_i + b_i x_i + c_i y_i \geq r)$: [Tawarmalani et al., 2008]
- convexification of $n-1$–convex functions: [Jach et al., 2008]

Symmetry breaking:

- use automorphisms in expression trees: [Liberti, 2008]

# Further topics

Tighter relaxations: E.g.,

- better handling of quadratic constraints: lift-and-project [Saxena et al., 2010], multiterm relaxations [Bao et al., 2009]
- convexification of trilinear $(x \cdot y \cdot z)$, quadrilinear $(w \cdot x \cdot y \cdot z)$, and multilinear forms: [Meyer and Floudas, 2004, Cafieri et al., 2010, Luedtke et al., 2010, Belotti et al., 2010]
- linear inequalities for orthogonal disjunctions and bilinear covering sets $(\sum_i a_i x_i y_i + b_i x_i + c_i y_i \geq r)$: [Tawarmalani et al., 2008]
- convexification of $n-1$–convex functions: [Jach et al., 2008]

Symmetry breaking:

- use automorphisms in expression trees: [Liberti, 2008]

Convexity detection:

- estimate spectrum of Hessian: [Nenov et al., 2004, Mönnigmann, 2008]
- walk expression tree and apply convexity rules for function compositions: not conclusive, fast [Fourer et al., 2009]
- apply quantifier elimination: conclusive, slower, only rational functions [Neun et al., 2010]

# Further topics

Tighter relaxations: E.g.,

- better handling of quadratic constraints: lift-and-project [Saxena et al., 2010], multiterm relaxations [Bao et al., 2009]
- convexification of trilinear $(x \cdot y \cdot z)$, quadrilinear $(w \cdot x \cdot y \cdot z)$, and multilinear forms: [Meyer and Floudas, 2004, Cafieri et al., 2010, Luedtke et al., 2010, Belotti et al., 2010]
- linear inequalities for orthogonal disjunctions and bilinear covering sets $(\sum_i a_i x_i y_i + b_i x_i + c_i y_i \geq r)$: [Tawarmalani et al., 2008]
- convexification of $n-1$–convex functions: [Jach et al., 2008]

Symmetry breaking:

- use automorphisms in expression trees: [Liberti, 2008]

Convexity detection:

- estimate spectrum of Hessian: [Nenov et al., 2004, Mönnigmann, 2008]
- walk expression tree and apply convexity rules for function compositions: not conclusive, fast [Fourer et al., 2009]
- apply quantifier elimination: conclusive, slower, only rational functions [Neun et al., 2010]

Nonlinear Integer Programming: [Hemmecke et al., 2009]

# LP/MIP based techniques for solving MINLPs

Stefan Vigerske

Humboldt-Universität zu Berlin

**DFG Research Center MATHEON**
**Mathematics for key technologies**

17th December 2010, Corporate Technology, Siemens AG

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

T. Achterberg. *Constraint Integer Programming*. PhD thesis, TU Berlin, 2007.

A. Atamtürk and V. Narayanan. Conic mixed integer rounding cuts. *Mathematical Programming*, 122:1–20, 2010.

X. Bao, N.V. Sahinidis, and M. Tawarmalani. Multiterm polyhedral relaxations for nonconvex, quadratically-constrained quadratic programs. *Optimization Methods and Software*, 24:485–504, 2009.

P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter. Branching and bounds tightening techniques for non-convex MINLP. *Optimization Methods and Software*, 24 (4-5):597–634, 2009. URL `http://www.optimization-online.org/DB_HTML/2008/08/2059.html`.

Pietro Belotti, Andrew Miller, and Mahdi Namazifar. Valid inequalities for sets defined by multilinear functions. In Pierre Bonami, Leo Liberti, Andrew J. Miller, and Annick Sartenaer, editors, *Proceedings of the European Workshop on Mixed Integer Nonlinear Programming*, pages 13–17, 2010.

T. Berthold and A.M. Gleixner. Undercover – a primal heuristic for MINLP based on sub-MIPs generated by set covering. ZIB-Report 09-40, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), December 2009. `http://opus.kobv.de/zib/volltexte/2009/1222`.

T. Berthold, S. Heinz, and S. Vigerske. Extending a CIP framework to solve MIQCPs. ZIB-Report 09-23, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 2009a. `http://opus.kobv.de/zib/volltexte/2009/1186`.

T. Berthold, S. Heinz, A. Gleixner, and S. Vigerske. On the computational impact of miqcp solver components. In *Proceedings of Hybrid Methods for Nonlinear Combinatorial Problems, workshop at CPAIOR 2010*, 2010.

Timo Berthold, Stefan Heinz, and Marc E. Pfetsch. Nonlinear pseudo-boolean optimization: relaxation or propagation? In Oliver Kullmann, editor, *Theory and Applications of Satisfiability Testing – SAT 2009*, number 5584 in LNCS, pages 441–446. Springer, 2009b.

Andreas Bley, Natashia Boland, Gary Froyland, and Mark Zuckerberg. Solving mixed integer nonlinear programming problems for mine production planning with a single stockpile. Technical Report 2009/21, Institute of Mathematics, TU Berlin, 2009a.

Andreas Bley, Ambros M. Gleixner, Thorsten Koch, and Stefan Vigerske. Comparing MIQCP solvers to a specialised algorithm for mine production scheduling. ZIB-Report 09-32, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), October 2009b. http://opus.kobv.de/zib/volltexte/2009/1206.

P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204, 2008.

P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot. A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming*, 119:331–352, 2009a.

P. Bonami, M. Kilinç, and J. Linderoth. Algorithms and software for convex mixed integer nonlinear programs. Technical Report 1664, Computer Sciences Department, University of Wisconsin-Madison, 2009b. available at http://www.optimization-online.org/DB_HTML/2009/10/2429.html.

P. Bonami, M. Kilinç, and J. Linderoth. Algorithms and software for convex mixed integer nonlinear programs. In *IMA Volumes*, volume to appear. 2010. available at http://www.optimization-online.org/DB_HTML/2009/10/2429.html.

Pierre Bonami and João P. M. Gonçalves. Primal heuristics for mixed integer nonlinear programs. Technical Report RC24639, IBM Research, 2008.

M. R. Bussieck, A. S. Drud, and A. Meeraus. MINLPLib - A Collection of Test Models for Mixed-Integer Nonlinear Programming. *INFORMS Journal on Computing*, 15(1): 114–119, 2003.

M.R. Bussieck and S. Vigerske. MINLP solver software. accepted for publication in Wiley EORMS, `http://www.math.hu-berlin.de/~stefan/minlpsoft.pdf`, 2010.

R.H. Byrd, J. Nocedal, and R.A. Waltz. KNITRO: An integrated package for nonlinear optimization. In G. di Pillo and M. Roma, editors, *Large-Scale Nonlinear Optimization*, pages 35–59. Springer, 2006.

Sonia Cafieri, Jon Lee, and Leo Liberti. On convex relaxations of quadrilinear terms. *Journal of Global Optimization*, 47(4):661–685, 2010.

M.T. Cezik and G. Iyengar. Cuts for mixed 0-1 conic programming. *Mathematical Programming*, 104:179–202, 2005.

S. Drewes. *Mixed Integer Second Order Cone Programming*. PhD thesis, Technische Universität Darmstadt, 2009.

M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

*Xpress-SLP Program Reference Manual*. FICO, 1.41 edition, 2008. `http://www.fico.com/xpress`.

R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming*, 66(3(A)):327–349, 1994.

R. Fourer, C. Maheshwari, A. Neumaier, D. Orban, and H. Schichl. Convexity and concavity detection in computational graphs: Tree walks for convexity assessment. *INFORMS Journal on Computing*, 22(1):26–43, 2009.

GAMS Development Corp. *GAMS - The Solver Manuals*. Washington DC, 2010.

A. M. Geoffrion. General Benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 1972.

I. E. Grossmann. Review of nonlinear mixed-integer and disjunctive programming techiques. *Optimization and Engineering*, 3:227–252, 2002.

I. E. Grossmann and S. Lee. Generalized disjunctive programming: Nonlinear convex hull relaxation and algorithms. *Computational Optimization and Applications*, (26): 83–100, 2003.

O. Günlük and J.T. Linderoth. Perspective relaxation of mixed integer nonlinear programs with indicator variables. *Mathematical Programming Series B*, 104:183–206, 2010.

O. Günlük, Jon Lee, and Robert Weismantel. MINLP strengthening for separable convex quadratic transportation-cost UFL. Technical Report RC24213, IBM Research, 2007. `http://www.optimization-online.org/DB_HTML/2007/03/1615.html`.

Raymond Hemmecke, Matthias Köppe, Jon Lee, and Robert Weismantel. *Nonlinear Integer Programming*, volume 3: Current Topics of *50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art*, chapter 15, pages 561–618. Springer, 2009. available at `http://front.math.ucdavis.edu/0906.5171`.

Hassan Hijazi, Pierre Bonami, Gérard Cornuéjols, and Adam Ouorou. Mixed integer nonlinear programs featuring "on/off" constraints: convex analysis and applications. `http://www.optimization-online.org/DB_HTML/2009/10/2442.html`, 2009.

M. Jach, D. Michaels, and R. Weismantel. The convex envelope of $(n–1)$-convex functions. *SIAM Journal on Optimization*, 19(3):1451–1466, 2008.

J. E. Kelley. The cutting-plane method for solving convex programs. *Journal of SIAM*, 8: 703–712, 1961.

M. Kilinç, J. Linderoth, and J. Luedtke. Effective separation of disjunctive cuts for convex mixed integer nonlinear programs. available at `http://www.optimization-online.org/DB_HTML/2010/11/2808.html`, 2010a.

M. Kilinç, J. Linderoth, J. Luedtke, and A. Miller. Strong branching inequalities for mixed integer nonlinear programs. working paper, 2010b.

G. R. Kocis and I. E. Grossmann. Relaxation strategy for the structural optimization of nonconvex MINLP problems in process flow sheets. *Ind. Eng. Chem. Res.*, 26(9):1869, 1987.

S. Leyffer. Deterministic methods for mixed-integer nonlinear programming. Technical report, Ph.D. thesis, Department of Mathematics and Computer Science, University of Dundee, Dundee, 1993.

S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18:295–309, 2001.

L. Liberti. Reformulations in mathematical programming: Symmetry. `http://www.optimization-online.org/DB_HTML/2008/12/2165.html`, 2008.

L. Liberti and C. Pantelides. Convex envelopes of monomials of odd degree. *J. Global Optimization*, 25:157–168, 2003.

Leo Liberti, Giacomo Nannicini, and Nenad Mladenovic. A good recipe for solving minlps. In V. Maniezzo, T. Stützle, and S. Voss, editors, *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, volume 10 of *Annals of Information Systems*. Springer, 2009.

Y. Lin and L. Schrage. The global solver in the LINDO API. *Optimization Methods & Software*, 24(4–5):657–668, 2009.

James Luedtke, Mahdi Namazifar, and Jeff Linderoth. Some results on the strength of relaxations of multilinear functions. Technical Report 1678, Computer Sciences Department, University of Wisconsin-Madison, 2010. available at `http://www.optimization-online.org/DB_HTML/2010/08/2711.html`.

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

G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I-Convex Underestimating Problems. *Mathematical Programming*, 10:147–175, 1976.

Clifford A. Meyer and Christodoulos A. Floudas. Trilinear monomials with mixed sign domains: Facets of the convex and concave envelopes. *Journal of Global Optimization*, 29:125–155, 2004. ISSN 0925-5001. URL http://dx.doi.org/10.1023/B:JOGO.0000042112.72379.e6.

M. Mönnigmann. Efficient calculation of bounds on spectra of Hessian matrices. *SIAM Journal on Scientific Computing*, 30(5):2340–2357, 2008. ISSN 1064-8275. doi: http://dx.doi.org/10.1137/070704186.

G. Nannicini, P. Belotti, and L. Liberti. A local branching heuristic for MINLPs. ArXiv, paper 0812.2188, 2009. URL http://arxiv.org/abs/0812.2188.

I. P. Nenov, D. H. Fylstra, and L. V. Kolev. Convexity determination in the Microsoft Excel solver using automatic differentiation techniques. Technical report, Frontline Systems Inc., 2004.

A. Neumaier. Constrained Global Optimization. In *COCONUT Deliverable D1, Algorithms for Solving Nonlinear Constrained and Optimization Problems: The State of The Art*, pages 55–111. http://www.mat.univie.ac.at/∼neum/glopt/coconut/StArt.html, 2001.

W. Neun, T. Sturm, and S. Vigerske. Supporting global numerical optimization of rational functions by generic symbolic convexity tests. In V.P. Gerdt, W. Koepf, E.W. Mayr, and E.H. Vorozhtsov, editors, *Computer Algebra in Scientific Computing*, volume 6244 of *Lecture Notes in Computer Science*. Springer, 2010. Proceedings of 12th International Workshop on Computer Algebra in Scientific Computing, CASC 2010, Tsakhadzor, Armenia.

I. Quesada and I. E. Grossmann. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Comp. Chem. Eng.*, 16:937–947, 1992.

M. Roelofs and J. Bisschop. *AIMMS 3.9 – The Language Reference*. Paragon Decision Technology B.V., Haarlem, The Netherlands, 2009.

Anureet Saxena, Pierre Bonami, and Jon Lee. Convex relaxations of non-convex mixed integer quadratically constrained programs: projected formulations. *Mathematical Programming*, pages 1–55, 2010. ISSN 0025-5610. URL `http://dx.doi.org/10.1007/s10107-010-0340-3`.

H. Schichl and A. Neumaier. Interval analysis on directed acyclic graphs for global optimization. *J. Global Optimization*, 33:541–562, 2005.

E. M. B. Smith and C. C. Pantelides. A Symbolic Reformulation/Spatial Branch-and-Bound Algorithm for the Global Optimization of nonconvex MINLPs. *Computers & Chemical Engineering*, 23:457–478, 1999.

R. Stubbs and S. Mehrotra. A branch-and-cut method for 0-1 mixed convex programming. *Mathematical Programming*, 86:515–532, 1999.

M. Tawarmalani and N.V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, 2002.

M. Tawarmalani and N.V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

M. Tawarmalani, J.-Ph. P. Richard, and K. Chung. Strong valid inequalities for orthogonal disjunctions and bilinear covering sets. `http://www.optimization-online.org/DB_HTML/2008/09/2100.html`, 2008.

A. Vecchietti and I. E. Grossmann. LOGMIP: A Disjunctive 0-1 Nonlinear Optimizer for Process System Models. *Comp. Chem. Eng.*, 23:555–565, 1999.

J. Viswanathan and I. E. Grossmann. A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization. *Computers & Chemical Engineering*, 14(7):769–782, 1990.

X. Vu, H. Schichl, and D. Sam-Haroud. Interval propagation and search on directed acyclic graphs for numerical constraint solving. available online at `http://www.mat.univie.ac.at/~herman/papers.html`, 2008.

T. Westerlund and F. Petterson. An extended cutting plane method for solving convex MINLP problems. *Computers & Chemical Engineering*, 21:131–136, 1995.

T. Westerlund and R. Pörn. Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering*, 3: 253–280, 2002.

X. Yuan, S. Zhang, L. Piboleau, and S. Domenech. Une methode d'optimisation nonlineare en variables mixtes pour la conception de procedes. *RAIRO*, 22:331, 1988.