# Bound reduction techniques for global optimization solvers

**Pietro Belotti** (Lehigh University, Bethlehem PA)

Joint work with: J. Lee, L. Liberti, F. Margot, A. Wächter

BR-OPT Workshop
CPAIOR, Pittsburgh, 28 May 2009

# Mixed-Integer Nonlinear Programming (MINLP)

$$\begin{array}{llll} \mathbf{P}_0) & \min & f(x) & \\ & s.t. & g_i(x) \leq 0 & i \in I \\ & & x_j^l \leq x_j \leq x_j^u & j \in N_0 = 1, 2 \ldots, n \\ & & x_j \in \mathbb{Z} & j \in J_0 \subseteq N_0 \end{array}$$

$f$ and $g_i$'s are, in general, nonconvex.

# Mixed-Integer Nonlinear Programming (MINLP)

$$
\begin{array}{llll}
\mathbf{P}_0) & \min & f(x) & \\
 & s.t. & g_i(x) \leq 0 & i \in I \\
 & & x_j^l \leq x_j \leq x_j^u & j \in N_0 = 1, 2 \ldots, n \\
 & & x_j \in \mathbb{Z} & j \in J_0 \subseteq N_0
\end{array}
$$

$f$ and $g_i$'s are, in general, nonconvex.

In many applications, integrality **and** nonlinearity are **both** essential for capturing the main aspects of the problem.

- Water treatment: Design of water networks (minimize the consumption of fresh water)
- Scheduling and blending for gasoline production plants
- Portfolio optimization. Discrete with transaction (fixed) costs, nonconvex with robustness

# Branch-and-bound for nonconvex MINLP

- Relaxing integrality returns a nonconvex NLP

# Branch-and-bound for nonconvex MINLP

- Relaxing integrality returns a nonconvex NLP

⇒ Branch-and-bound: enumerate implicitly all local minima, use a convex (linear) relaxation to find lower bounds (Smith&Pantelides 1997, Tawarmalani&Sahinidis 2002)

# Branch-and-bound for nonconvex MINLP

- Relaxing integrality returns a nonconvex NLP

⇒ Branch-and-bound: enumerate implicitly all local minima, use a convex (linear) relaxation to find lower bounds (Smith&Pantelides 1997, Tawarmalani&Sahinidis 2002)

- Components:
  - linearization (or convexification) for lower bounds
  - heuristics for upper bounds
  - branching rules to **partition** the solution set

# Branch-and-bound for nonconvex MINLP

- Relaxing integrality returns a nonconvex NLP

$\Rightarrow$ Branch-and-bound: enumerate implicitly all local minima, use a convex (linear) relaxation to find lower bounds (Smith&Pantelides 1997, Tawarmalani&Sahinidis 2002)

- Components:
  - linearization (or convexification) for lower bounds
  - heuristics for upper bounds
  - branching rules to **partition** the solution set
  - bound reduction to **reduce** the solution set

# Branch-and-bound for nonconvex MINLP

- Relaxing integrality returns a nonconvex NLP

⇒ Branch-and-bound: enumerate implicitly all local minima, use a convex (linear) relaxation to find lower bounds (Smith&Pantelides 1997, Tawarmalani&Sahinidis 2002)

- Components:
  - linearization (or convexification) for lower bounds
  - heuristics for upper bounds
  - branching rules to **partition** the solution set
  - bound reduction to **reduce** the solution set

Solvers:

- $\alpha$**-BB** (Adjiman et al. 1998)
- **Baron** (Tawarmalani&Sahinidis 2002)
- **Couenne** (B., Lee, Liberti, Margot, Wächter 2008)
- (upcoming) **Minotaur** (Linderoth, Leyffer, Munson)
- **LaGO** (Nowak&Vigerske 2003)

# Branch-and-bound for nonconvex MINLP

- Relaxing integrality returns a nonconvex NLP

⇒ Branch-and-bound: enumerate implicitly all local minima, use a convex (linear) relaxation to find lower bounds (Smith&Pantelides 1997, Tawarmalani&Sahinidis 2002)

- Components:
  - linearization (or convexification) for lower bounds
  - heuristics for upper bounds
  - branching rules to **partition** the solution set
  - bound reduction to **reduce** the solution set

Solvers:

- $\alpha$**-BB** (Adjiman et al. 1998)
- **Baron** (Tawarmalani&Sahinidis 2002)
- **Couenne** (B., Lee, Liberti, Margot, Wächter 2008)
- (upcoming) **Minotaur** (Linderoth, Leyffer, Munson)
- **LaGO** (Nowak&Vigerske 2003)
- convex: **Bonmin** (Bonami et al.), **FilMINT** (Abhishek et al.)

# Couenne, a solver for nonconvex MINLPs

`Couenne`[1] is a Branch&Bound for nonconvex MINLPs. Written in C++, available as Open Source in Coin-OR (www.coin-or.org), it implements

- linearization of nonconvex functions
- heuristics for upper bound
- specialized branching rules
- bound reduction

It uses code from `Bonmin` (MINLP B&B), `Cbc` (Branch&Bound), `Cgl` (Cut generation), `Clp` (LP solver), `Ipopt` (NLP solver), and `LaGO` (quadratic forms).

[1] **C**onvex **O**ver/**U**nder **EN**velopes for **N**onlinear **E**stimation

# Outline

- Reformulation and linearization
- Bound reduction
- Use of bound reduction:
  - Meta-BR
  - Branching techniques
  - Disjunctive cuts
- Computational results

# Reformulation

$$
\begin{array}{lll}
\mathbf{P}_0: & \min & f(x) \\
& s.t. & g_i(x) \leq 0 \qquad i \in I \\
& & x_j^l \leq x_j \leq x_j^u \quad j \in N_0 = 1, 2 \ldots, n \\
& & x_j \in \mathbb{Z} \qquad j \in J_0 \subseteq N_0
\end{array}
$$

# Reformulation

$$
\begin{array}{llll}
\mathbf{P}_0: & \min & f(x) & \\
& s.t. & g_i(x) \leq 0 & i \in I \\
& & x_j^l \leq x_j \leq x_j^u & j \in N_0 = 1, 2 \ldots, n \\
& & x_j \in \mathbb{Z} & j \in J_0 \subseteq N_0
\end{array}
$$

is **reformulated** as an equivalent problem

$$
\begin{array}{llll}
\mathbf{P}: & \min & x_{n+q} & \\
& s.t. & x_k = \vartheta_k(x_1, x_2 \ldots, x_{k-1}) & k = n+1, n+2 \ldots, n+q \\
& & x_j^l \leq x_j \leq x_j^u & j \in N = 1, 2 \ldots, n+q \\
& & x_j \in \mathbb{Z} & j \in J \subseteq N
\end{array}
$$

# Reformulation

$$\begin{array}{llll} \mathbf{P}_0: & \min & f(x) & \\ & s.t. & g_i(x) \le 0 & i \in I \\ & & x_j^l \le x_j \le x_j^u & j \in N_0 = 1, 2 \ldots, n \\ & & x_j \in \mathbb{Z} & j \in J_0 \subseteq N_0 \end{array}$$

is **reformulated** as an equivalent problem

$$\begin{array}{llll} \mathbf{P}: & \min & x_{n+q} & \\ & s.t. & x_k = \vartheta_k(x_1, x_2 \ldots, x_{k-1}) & k = n+1, n+2 \ldots, n+q \\ & & x_j^l \le x_j \le x_j^u & j \in N = 1, 2 \ldots, n+q \\ & & x_j \in \mathbb{Z} & j \in J \subseteq N \end{array}$$

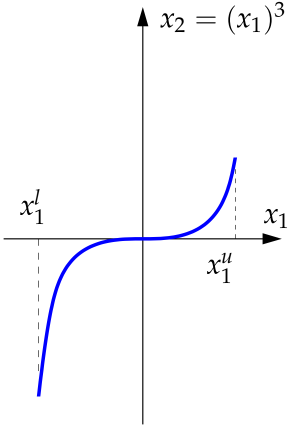
and each $x_k = \vartheta_k(x_1, x_2, \ldots, x_{k-1})$ is **linearized**:

$$\begin{array}{llll} \mathbf{LP}: & \min & x_{n+q} & \\ & s.t. & a^k x_k + A^k x \ge b^k & k = n+1, n+2 \ldots, n+q \\ & & x_j^l \le x_j \le x_j^u & j \in N = 1, 2 \ldots, n+q \\ & & x_j \in \mathbb{Z} & j \in J \subseteq N \end{array}$$

# Linearization[2,3,4]

Find a **polyhedral** set $S$ s.t.

$$S \supseteq \{(x_1, x_2) \in \mathbb{R}^2 : x_2 = \vartheta_2(x_1) = (x_1)^3, x_1^l \leq x_1 \leq x_1^u\}$$

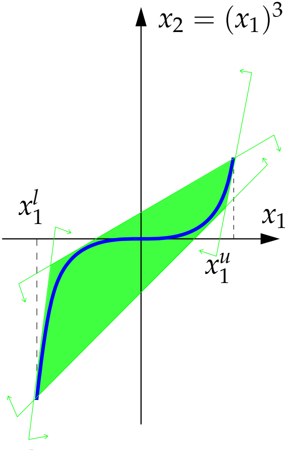



[2] E. M. B. Smith and C. C. Pantelides, *Computers & Chemical Eng*. 23, 1999
[3] M. Tawarmalani, N. Sahinidis, *Math. Prog*. 99, 2004
[4] B., Lee, Liberti, Margot, Wächter, *opt-online* 2008

# Linearization[2,3,4]

Find a **polyhedral** set $S$ s.t.

$$S \supseteq \{(x_1, x_2) \in \mathbb{R}^2 : x_2 = \vartheta_2(x_1) = (x_1)^3, x_1^l \le x_1 \le x_1^u\}$$

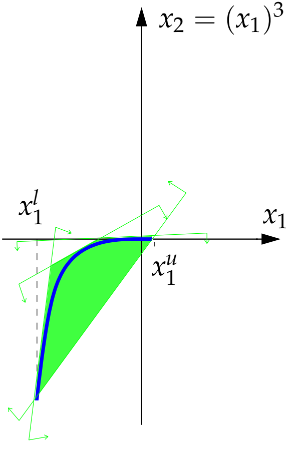



[2] E. M. B. Smith and C. C. Pantelides, *Computers & Chemical Eng.* 23, 1999
[3] M. Tawarmalani, N. Sahinidis, *Math. Prog.* 99, 2004
[4] B., Lee, Liberti, Margot, Wächter, *opt-online* 2008

# Linearization[2,3,4]

Find a **polyhedral** set $S$ s.t.

$$S \supseteq \{(x_1, x_2) \in \mathbb{R}^2 : x_2 = \vartheta_2(x_1) = (x_1)^3, x_1^l \le x_1 \le x_1^u\}$$




[2] E. M. B. Smith and C. C. Pantelides, *Computers & Chemical Eng.* 23, 1999
[3] M. Tawarmalani, N. Sahinidis, *Math. Prog.* 99, 2004
[4] B., Lee, Liberti, Margot, Wächter, *opt-online* 2008

# Bound Reduction (BR)

Quality of linearization depends on the variables bounds

$\Rightarrow$ BR can dramatically improve performance of MINLP solvers

# Bound Reduction (BR)

Quality of linearization depends on the variables bounds
$\Rightarrow$ BR can dramatically improve performance of MINLP solvers

Two types:

OBBR: Optimality Based BR, slow but very effective

# Bound Reduction (BR)

Quality of linearization depends on the variables bounds

$\Rightarrow$ BR can dramatically improve performance of MINLP solvers

Two types:

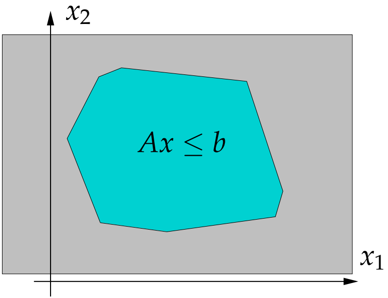
OBBR: Optimality Based BR, slow but very effective

FBBR: Feasibility Based BR, fast but not tightening much

# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

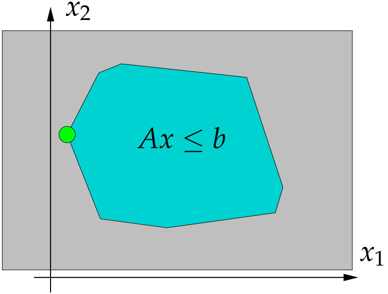
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

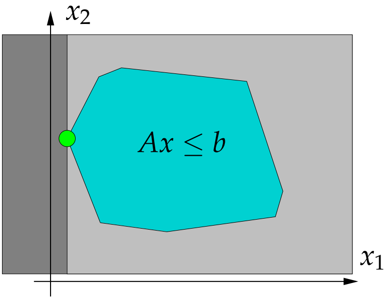
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

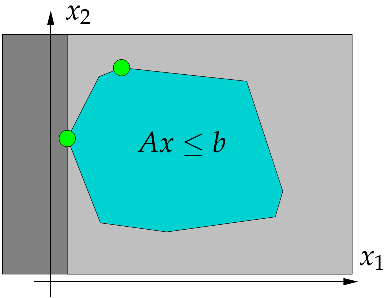
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

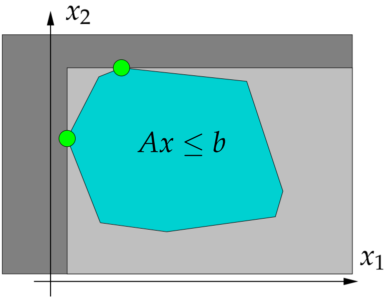
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

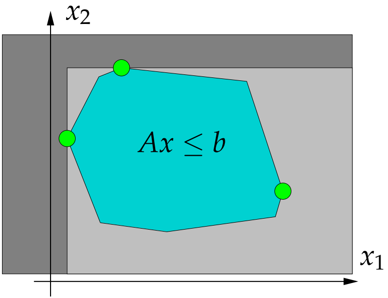
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

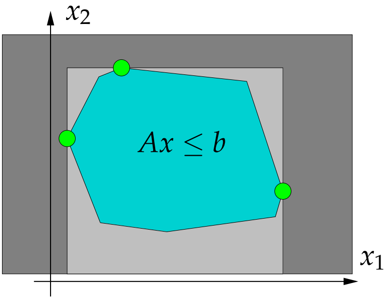
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

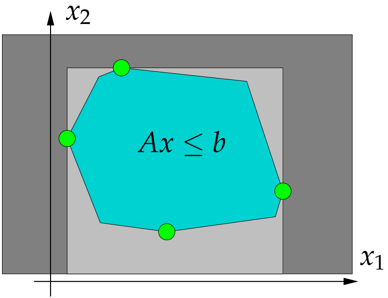
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

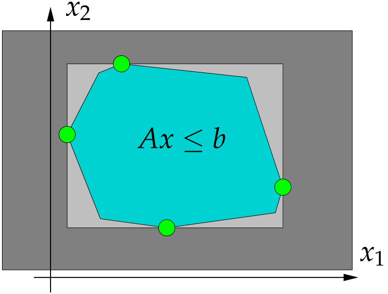
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization $Ax \leq b$, for each variable $x_j$

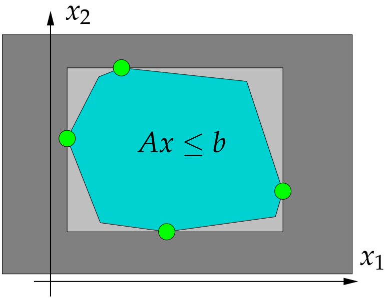
$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$



# Optimality Based BR (OBBR)

Given current linearization
$Ax \leq b$, for each variable $x_j$

$$x_j \geq \min\{x_j : Ax \leq b\}$$
$$x_j \leq \max\{x_j : Ax \leq b\}$$




- ☺ very effective
- ☹ $2n'$ LPs to solve ($n'$ = original + auxiliary)

# Feasibility Based BR (FBBR)

- Given $x_k = \vartheta(x_1, x_2 \ldots, x_{k-1})$, we can obtain better bounds on $x_k$ from bounds on $x_1, x_2 \ldots, x_{k-1}$, and viceversa

# Feasibility Based BR (FBBR)

- Given $x_k = \vartheta(x_1, x_2 \ldots, x_{k-1})$, we can obtain better bounds on $x_k$ from bounds on $x_1, x_2 \ldots, x_{k-1}$, and viceversa

e.g. using Interval Analysis methods

- For $x_k = a_0 + \sum_{j=1}^{k-1} a_j x_j$, well known technique in LP

# Feasibility Based BR (FBBR)

- Given $x_k = \vartheta(x_1, x_2 \ldots, x_{k-1})$, we can obtain better bounds on $x_k$ from bounds on $x_1, x_2 \ldots, x_{k-1}$, and viceversa

e.g. using Interval Analysis methods

- For $x_k = a_0 + \sum_{j=1}^{k-1} a_j x_j$, well known technique in LP
- In MINLP, apply it to all operators. Examples:

$$(\rightarrow) \quad x_2 = x_1^2 \qquad -2 \leq x_1 \leq 3 \quad \Rightarrow \quad 0 \leq x_2 \leq 9$$

# Feasibility Based BR (FBBR)

- Given $x_k = \vartheta(x_1, x_2 \ldots, x_{k-1})$, we can obtain better bounds on $x_k$ from bounds on $x_1, x_2 \ldots, x_{k-1}$, and viceversa

e.g. using Interval Analysis methods

- For $x_k = a_0 + \sum_{j=1}^{k-1} a_j x_j$, well known technique in LP
- In MINLP, apply it to all operators. Examples:

$$
\begin{array}{lllll}
(\rightarrow) & x_2 = x_1^2 & -2 \leq x_1 \leq 3 & \Rightarrow & 0 \leq x_2 \leq 9 \\
(\leftarrow) & x_3 = \log x_2 & x_3 \leq -5 & \Rightarrow & x_2 \leq e^{-5}
\end{array}
$$

# Feasibility Based BR (FBBR)

- Given $x_k = \vartheta(x_1, x_2 \ldots, x_{k-1})$, we can obtain better bounds on $x_k$ from bounds on $x_1, x_2 \ldots, x_{k-1}$, and viceversa

e.g. using Interval Analysis methods

- For $x_k = a_0 + \sum_{j=1}^{k-1} a_j x_j$, well known technique in LP
- In MINLP, apply it to all operators. Examples:

$$\begin{array}{llccc} (\rightarrow) & x_2 = x_1^2 & -2 \leq x_1 \leq 3 & \Rightarrow & 0 \leq x_2 \leq 9 \\ (\leftarrow) & x_3 = \log x_2 & x_3 \leq -5 & \Rightarrow & x_2 \leq e^{-5} \end{array}$$

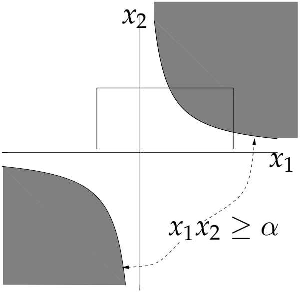
A fast, flexible tool to eliminate portions of the solution space.

# FBBR: an example

- $x_3 = x_1 x_2$
- $x_3 \geq \alpha > 0$

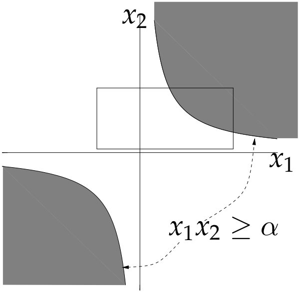
# FBBR: an example

- $x_3 = x_1 x_2$
- $x_3 \geq \alpha > 0$
- $(x_1, x_2) \in [x_1^{\mathsf{L}}, x_1^{\mathsf{U}}] \times [x_2^{\mathsf{L}}, x_2^{\mathsf{U}}]$



# FBBR: an example

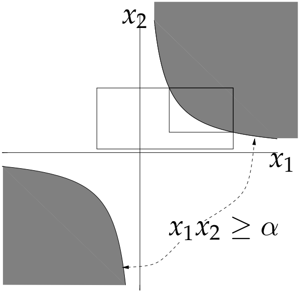
- $x_3 = x_1 x_2$
- $x_3 \geq \alpha > 0$
- $(x_1, x_2) \in [x_1^{\mathsf{L}}, x_1^{\mathsf{U}}] \times [x_2^{\mathsf{L}}, x_2^{\mathsf{U}}]$
- if $x_1^{\mathsf{L}} x_2^{\mathsf{L}} < \alpha$
  (i.e. $(x_1^{\mathsf{L}}, x_2^{\mathsf{L}})$ infeasible)



# FBBR: an example

- $x_3 = x_1 x_2$
- $x_3 \geq \alpha > 0$
- $(x_1, x_2) \in [x_1^{\mathsf{L}}, x_1^{\mathsf{U}}] \times [x_2^{\mathsf{L}}, x_2^{\mathsf{U}}]$
- if $x_1^{\mathsf{L}} x_2^{\mathsf{L}} < \alpha$
  (i.e. $(x_1^{\mathsf{L}}, x_2^{\mathsf{L}})$ infeasible)

$$\Rightarrow \begin{cases} x_1 \geq \alpha / x_2^{\mathsf{U}} \\ x_2 \geq \alpha / x_1^{\mathsf{U}} \end{cases}$$



# OBBR and FBBR work well together

OBBR and FBBR are somewhat complementary:

- OBBR uses the (whole) LP relaxation
- FBBR uses (single expressions of) the initial MINLP only

# OBBR and FBBR work well together

OBBR and FBBR are somewhat complementary:

- OBBR uses the (whole) LP relaxation
- FBBR uses (single expressions of) the initial MINLP only

Although OBBR is generally more efficient, it does *not* dominate FBBR, which uses different information.

- OBBR would be useless in other contexts (e.g. LP)
- in MINLP, it feeds better bounds to FBBR, which in turn improves others

# Uses of BR techniques

Bound reduction is a basic tool that can be used in various contexts. A few examples:

- Aggressive BR ($\approx$ probing)
- Strong/reliability branching
- Disjunctive cut generation

# Aggressive bound reduction (probing)

Suppose a feasible solution $x^{\mathrm{nl}}$ with value $z^{\mathrm{nl}}$ is given; we want to focus the search around it.
$\Rightarrow$ exclude portions of bounding box around it by proving

- they are infeasible, **or**
- they have a lower bound above $z^{\mathrm{nl}}$

# Aggressive bound reduction (probing)

Suppose a feasible solution $x^{\text{nl}}$ with value $z^{\text{nl}}$ is given; we want to focus the search around it.
$\Rightarrow$ exclude portions of bounding box around it by proving

- they are infeasible, **or**
- they have a lower bound above $z^{\text{nl}}$

- Apply FBBR to subproblems
  $S_1 : [l, u_{\text{F}}]$ and $S_2 : [l_{\text{F}}, u]$

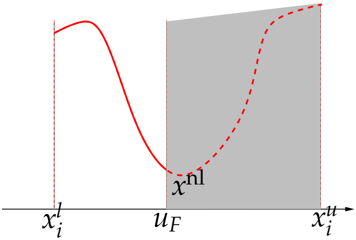
# Aggressive bound reduction (probing)

Suppose a feasible solution $x^{\mathrm{nl}}$ with value $z^{\mathrm{nl}}$ is given; we want to focus the search around it.

$\Rightarrow$ exclude portions of bounding box around it by proving

- they are infeasible, **or**
- they have a lower bound above $z^{\mathrm{nl}}$

- Apply FBBR to subproblems $S_1 : [l, u_{\mathrm{F}}]$ and $S_2 : [l_{\mathrm{F}}, u]$
- If $S_1$ (resp. $S_2$) is pruned either on bounds or on infeasibility, $[u_{\mathrm{F}}, u]$ (resp. $[l, l_{\mathrm{F}}]$) is a valid reduction

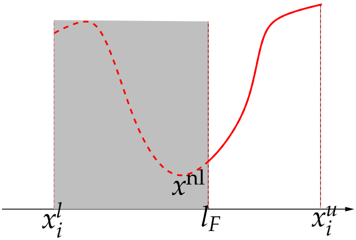


# Aggressive bound reduction (probing)

Suppose a feasible solution $x^{\mathrm{nl}}$ with value $z^{\mathrm{nl}}$ is given; we want to focus the search around it.
$\Rightarrow$ exclude portions of bounding box around it by proving

- they are infeasible, **or**
- they have a lower bound above $z^{\mathrm{nl}}$

- Apply FBBR to subproblems $S_1 : [l, u_{\mathrm{F}}]$ and $S_2 : [l_{\mathrm{F}}, u]$
- If $S_1$ (resp. $S_2$) is pruned either on bounds or on infeasibility, $[u_{\mathrm{F}}, u]$ (resp. $[l, l_{\mathrm{F}}]$) is a valid reduction



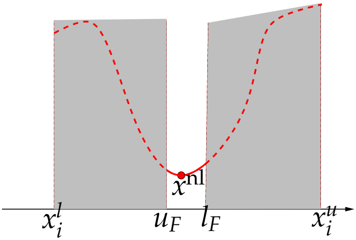
# Aggressive bound reduction (probing)

Suppose a feasible solution $x^{\mathrm{nl}}$ with value $z^{\mathrm{nl}}$ is given; we want to focus the search around it.

$\Rightarrow$ exclude portions of bounding box around it by proving

- they are infeasible, **or**
- they have a lower bound above $z^{\mathrm{nl}}$

- Apply FBBR to subproblems $S_1 : [l, u_{\mathrm{F}}]$ and $S_2 : [l_{\mathrm{F}}, u]$
- If $S_1$ (resp. $S_2$) is pruned either on bounds or on infeasibility, $[u_{\mathrm{F}}, u]$ (resp. $[l, l_{\mathrm{F}}]$) is a valid reduction



# Disjunctive cuts for MINLP

A *disjunction* partitions the solution set. Examples:

MILP: if $x_i$ is constrained to be integer, a disjunction is

$$x_i \leq 4 \quad \vee \quad x_i \geq 5$$

MINLP: if $x_i$ is a continuous variable, a disjunction is

$$x_i < 3.23 \quad \vee \quad x_i \geq 3.23$$

From a set $X$ of solutions, a disjunction creates two feasible sets:

$$X_< = \{x \in X : x_i < 3.23\} \quad \text{and} \quad X_\geq = \{x \in X : x_i \geq 3.23\}$$

A *disjunctive cut*[5] for an MINLP is a valid ineq. for $X_< \cup X_\geq$.

[5]E. Balas, *Op. Res.* 19:19-39, 1971.

# Disjunctive cuts for MINLP: an example

Consider a continuous nonconvex NLP:

$$(P) \quad \min \quad x^2$$
$$x^4 \geq 1$$

Feasible set $(-\infty, -1] \cup [+1, +\infty)$, two global minima $\{-1, +1\}$.

# Disjunctive cuts for MINLP: an example

Consider a continuous nonconvex NLP:

$$(P) \quad \min \quad x^2$$
$$x^4 \geq 1$$

Feasible set $(-\infty, -1] \cup [+1, +\infty)$, two global minima $\{-1, +1\}$.

Reformulation:

$$(P^R) \quad \min \quad w$$
$$y \geq 1$$
$$w = x^2$$
$$y = x^4$$

# Disjunctive cuts for MINLP: an example

Consider a continuous nonconvex NLP:

$$(P) \quad \min \quad x^2$$
$$x^4 \geq 1$$

Feasible set $(-\infty, -1] \cup [+1, +\infty)$, two global minima $\{-1, +1\}$.

Reformulation:

$$(P^R) \quad \min \quad w$$
$$y \geq 1$$
$$w = x^2$$
$$y = x^4$$

Convex relaxation:

$$(P^C) \quad \min \quad w$$
$$y \geq 1$$
$$w \geq x^2$$
$$y \geq x^4$$

# Disjunctive cuts for MINLP: an example

Consider a continuous nonconvex NLP:

$$(P) \quad \min \quad x^2 \qquad x^4 \geq 1$$

Feasible set $(-\infty, -1] \cup [+1, +\infty)$, two global minima $\{-1, +1\}$.

Reformulation:

$$(P^R) \quad \min \quad w \qquad y \geq 1, \quad w = x^2, \quad y = x^4$$

Convex relaxation:

$$(P^C) \quad \min \quad w \qquad y \geq 1, \quad w \geq x^2, \quad y \geq x^4$$

Optimum of $P^C$: $(x, w, y) = (0, 0, 1)$, infeasible for $P^R$ (and $P$).

# Example: a simple nonconvex problem (2)

Apply disjunction $x < 0 \vee x \geq 0$. The relaxations are

$$\begin{array}{ll} (P^C_{\mathrm{L}}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x < 0\} \\ (P^C_{\mathrm{R}}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x \geq 0\} \end{array}$$

[6] After one call to Bound Reduction…

# Example: a simple nonconvex problem (2)

Apply disjunction $x < 0 \vee x \geq 0$. The relaxations are[6]

$$
\begin{array}{ll}
(P_{\mathrm{L}}^{C}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x \leq -1\} \\
(P_{\mathrm{R}}^{C}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x \geq +1\}
\end{array}
$$

$\Rightarrow$ two B&B nodes, both find a global optimum ($-1$ and $+1$)

[6] After one call to Bound Reduction…

# Example: a simple nonconvex problem (2)

Apply disjunction $x < 0 \vee x \geq 0$. The relaxations are[6]

$$\begin{array}{ll} (P^C_{\mathrm{L}}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x \leq -1\} \\ (P^C_{\mathrm{R}}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x \geq +1\} \end{array}$$

$\Rightarrow$ two B&B nodes, both find a global optimum ($-1$ and $+1$)

- However, the **disjunctive cut** $w \geq 1$, valid for $P^C_{\mathrm{L}} \cup P^C_{\mathrm{R}}$, solves the problem at root node

[6] After one call to Bound Reduction…

# Example: a simple nonconvex problem (2)

Apply disjunction $x < 0 \vee x \geq 0$. The relaxations are[6]

$$\begin{array}{ll} (P_{\mathrm{L}}^{C}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x \leq -1\} \\ (P_{\mathrm{R}}^{C}) & \min\{w : y \geq 1, w \geq x^2, y \geq x^4, x \geq +1\} \end{array}$$

$\Rightarrow$ two B&B nodes, both find a global optimum ($-1$ and $+1$)

- However, the **disjunctive cut** $w \geq 1$, valid for $P_{\mathrm{L}}^{C} \cup P_{\mathrm{R}}^{C}$, solves the problem at root node
- Can be complicated arbitrarily:

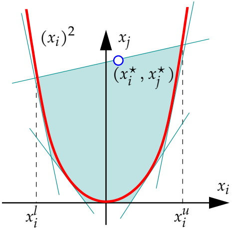
$$\begin{array}{ll} \min & \sum_{i=1}^{n} x_i^2 \\ & x_i^4 \geq 1 \qquad \forall i = 1, 2 \ldots, n \end{array}$$

[6] After one call to Bound Reduction…

# MINLP disjunctive cuts

If the (linear) relaxation of the MINLP is

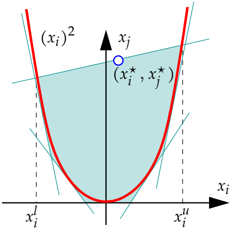
$$(P^C) \quad \min\{x_{n+q} : Ax \geq a_0\}$$



# MINLP disjunctive cuts

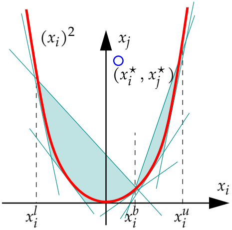
If the (linear) relaxation of the MINLP is

$$(P^C) \quad \min\{x_{n+q} : Ax \geq a_0\}$$




and the relaxations arising from the disjunction are (after bound reduction)

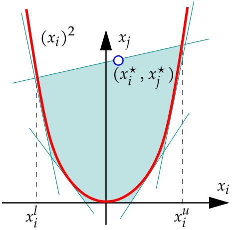
$$(P^C_{\mathrm{L}}) \quad \min\{x_{n+q} : Ax \geq a, Bx \geq b\}$$
$$(P^C_{\mathrm{R}}) \quad \min\{x_{n+q} : Ax \geq a, Cx \geq c\}$$



# MINLP disjunctive cuts

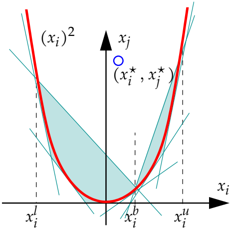
If the (linear) relaxation of the MINLP is

$$(P^C) \quad \min\{x_{n+q} : Ax \geq a_0\}$$




and the relaxations arising from the disjunction are (after bound reduction)

$$(P^C_{\mathrm{L}}) \quad \min\{x_{n+q} : Ax \geq a, Bx \geq b\}$$
$$(P^C_{\mathrm{R}}) \quad \min\{x_{n+q} : Ax \geq a, Cx \geq c\}$$




then a cut $\boxed{\alpha x \geq \beta}$ is valid if

$$\begin{aligned} &\alpha \geq uA + u'B, \quad \beta = ua + u'b, \\ &\alpha \geq vA + v'C, \quad \beta = va + v'c, \\ &u, u', v, v' \geq 0. \end{aligned}$$

# The Cut Generating LP (CGLP)

In order to get a "deepest" cut, i.e., one that is **most violated** by the current solution $x^\star$ of the relaxation, solve[7]

$$
\begin{array}{llllll}
(CGLP) & \min & \alpha x^\star - \beta & & & \\
 & & \alpha & \geq uA & +u'B & \\
 & & \alpha & \geq vA & & +v'C \\
 & & \beta & = ua & +u'b & \\
 & & \beta & = va & & +v'c \\
 & & u, u', v, v' & \geq 0 & & \\
 & & ||(u, u', v, v')||_1 & \leq 1 & &
\end{array}
$$

Optimal $\alpha$ and $\beta$ provide an inequality that is **valid** for both "sides" of the disjunction.

[7] See e.g. Balas, Ceria, Cornuéjols, *Math. Prog.* 89:1-44, 1998.

# Strong branching

Strong branching[8]: for each branching candidate $x_i$,

- simulate br. rule $x_i \leq x_i^b$, and apply BR
- solve $LP_{\leq} \rightarrow$ new lower bound $z^{i,\leq}$

[8] Applegate et al., "The TSP, a computational study".
[9] Benichou et al, "Experiments in MIP", *MathProg* '71
[10] Achterberg et al., "Branching rules revisited", *OR letters* 2005

# Strong branching

Strong branching[8]: for each branching candidate $x_i$,

- simulate br. rule $x_i \leq x_i^b$, and apply BR
- solve $LP_{\leq} \rightarrow$ new lower bound $z^{i,\leq}$
- simulate br. rule $x_i \geq x_i^b$, and apply BR
- solve $LP_{\geq} \rightarrow$ new lower bound $z^{i,\geq}$

[8] Applegate et al., "The TSP, a computational study".
[9] Benichou et al, "Experiments in MIP", *MathProg* '71
[10] Achterberg et al., "Branching rules revisited", *OR letters* 2005

# Strong branching

Strong branching[8]: for each branching candidate $x_i$,

- simulate br. rule $x_i \leq x_i^b$, and apply BR
- solve $LP_{\leq} \rightarrow$ new lower bound $z^{i,\leq}$
- simulate br. rule $x_i \geq x_i^b$, and apply BR
- solve $LP_{\geq} \rightarrow$ new lower bound $z^{i,\geq}$
- Branch on variables with maximum $\alpha \min\{z^{i,\leq}, z^{i,\geq}\} + (1-\alpha)\max\{z^{i,\leq}, z^{i,\geq}\}$, with $0 < \alpha < 1$

Computationally expensive…
$\Rightarrow$Pseudocosts[9], Reliability Branching[10]: statistics at initial nodes are used to *estimate* it at later nodes

[8] Applegate et al., "The TSP, a computational study".
[9] Benichou et al, "Experiments in MIP", *MathProg* '71
[10] Achterberg et al., "Branching rules revisited", *OR letters* 2005

# Computational tests

How does BR affect the performance of a MINLP solver?

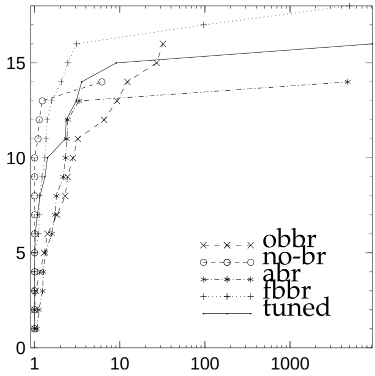
We have tested various combinations of BR techniques:

- **no BR**: the simplest branch&bound with no reduction
- **FBBR**: only Feasibility-based BR
- **OBBR**: only Optimality-based BR
- **ABR**: Aggressive BR (probing) and FBBR
- **tuned**: ABR and OBBR applied at select nodes, FBBR at all nodes of the branch&bound tree

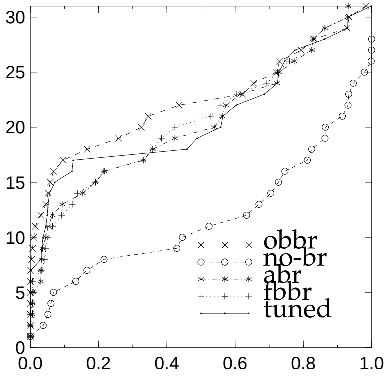
Testbed: 90 instances from publicly available instance libraries (`globallib`, `minlplib`, `macminlp`) – see B., Lee, Liberti, Margot, Wächter, *opt-online*, July 2008 for more complete results

# Experimental results – MINLP instances

Performance profiles for CPU time and “remaining gap” — i.e., gap between lower and upper bound that is still there after one hour of execution time.




CPU time

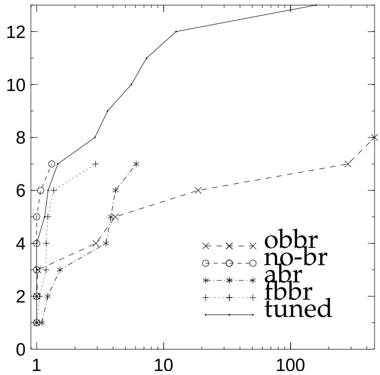



Remaining gap

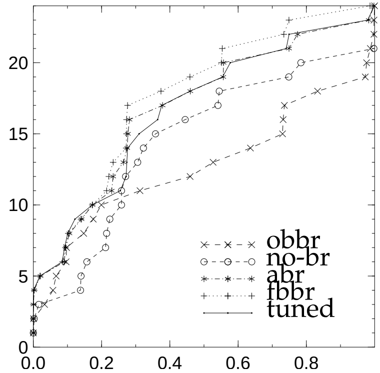
(A higher line corresponds to a better algorithm)

# Experimental results – (continuous) NLP instances

Performance profiles for CPU time and "remaining gap" — i.e., gap between lower and upper bound that is still there after one hour of execution time.




CPU time




Remaining gap

(A higher line corresponds to a better algorithm)

# Use in Strong branching and Disjunctive cuts

Time limit: one hour

| | | | | time (lower bound) | | | |
|---|---|---|---|---|---|---|---|
| Name | var | int | con | none | BR | SB+DC | all |
| du-opt | 20 | 13 | 8 | 4.9 | 130.4 | (3.55) | 677.8 |
| ex1252 | 39 | 15 | 43 | (0.0) | (72912) | (12666) | 54.2 |
| synheat | 53 | 12 | 61 | (66260) | (74922) | (100500) | 366.49 |
| ex1244 | 86 | 21 | 110 | (54014) | 19.4 | (61368.4) | 103.1 |
| multist. | 185 | 18 | 265 | – | (-2.4921) | 0.74 | 213.72 |
| ibell3a | 182 | 60 | 104 | 511 | 394 | (877143) | 529 |
| trimlon12 | 168 | 168 | 72 | (10.2) | (19.5) | (16.7) | (23.2) |
| sched47 | 233 | 140 | 138 | (-1e+8) | (-6e+6) | (-2e+7) | (-3e+5) |
| aich1 | 818 | 23 | 987 | (-113.2) | (-110.6) | (-103.5) | (-103.0) |

**none**: no BR, no SB, no DC **BR**: bound reduction only
**SB+DC**: Strong Br., Disj. cuts **all**: BR and SB and DC

# Use in Strong branching and Disjunctive cuts

Time limit: one hour

| | | | | time (lower bound) | | | |
|---|---|---|---|---|---|---|---|
| Name | var | int | con | none | BR | SB+DC | all |
| nvs19 | 8 | 8 | 8 | 12.0 | 11.5 | (-2116) | 306.9 |
| nvs23 | 9 | 9 | 9 | 31.5 | 32.14 | (-3483) | 1297.96 |
| nous1 | 48 | 2 | 41 | (-.5) | (1.1) | (3.8) | (6.7) |
| nous2 | 48 | 2 | 41 | (-.47) | 3.9 | 1746.5 | 5.9 |
| foulds3 | 168 | 0 | 48 | (-79.5) | (-73.3) | (-87.4) | (-63.3) |
| qshare1b | 221 | 0 | 111 | 853 | 796 | (719015) | (720126) |
| qetamacr | 543 | 0 | 334 | (65437) | (64922) | (64076) | (63980) |

**none**: no BR, no SB, no DC

**BR**: bound reduction only

**SB+DC**: Strong Br., Disj. cuts

**all**: BR and SB and DC