

Mixed-Integer Nonlinear Optimization: Convex MINLPs

GIAN Short Course on Optimization:
Applications, Algorithms, and Computation

Sven Leyffer

Argonne National Laboratory

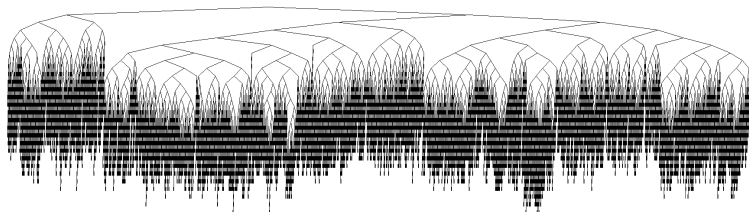
September 12-24, 2016

Outline

- 1 Multi-Tree Methods
- 2 Single-Tree Methods
- 3 Presolve for MINLP



Motivation MINLP Trees are Huge



Synthesis MINLP B&B Tree: 10000+ nodes after 360s

- Requires solution of thousands of NLPs
QP solves can be good alternative
- Can we have even faster solves at nodes?
Consider MILP solvers to search tree ...

Multi-Tree Methods

MILP solvers much better developed than MINLP

- LPs are easy to hot-start
- Decades of investment into software
- MILPs much easier; e.g. no need for constraint qualifications

⇒ developed methods that exploit this technology

Multi-Tree Methods

- Outer approximation [Duran and Grossmann, 1986]
- Benders decomposition [Geoffrion, 1972]
- Extended cutting plane method
[Westerlund and Pettersson, 1995]

... solve a sequence of MILP (and NLP) problems

Multi-tree methods evaluate functions “only” at integer points!



Multi-Tree Methods

Recall the η -MINLP formulation

$$\left\{ \begin{array}{l} \underset{\eta, x}{\text{minimize}} \quad \eta, \\ \text{subject to} \quad f(x) \leq \eta, \\ \quad \quad \quad c(x) \leq 0, \\ \quad \quad \quad x \in \mathcal{X}, \\ \quad \quad \quad x_i \in \mathbb{Z}, \forall i \in \mathcal{I}. \end{array} \right.$$

where we have “linearized” the objective: $\eta \geq f(x)$

Use η -MINLP in this section



Outer Approximation

Mixed-Integer Nonlinear Program (**MINLP**)

$$\underset{x}{\text{minimize}} \quad f(x) \quad \text{subject to} \quad c(x) \leq 0, \quad x \in \mathcal{X}, \quad x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}$$

NLP subproblem for fixed integers $x_{\mathcal{I}}^{(j)}$:

$$\text{NLP}(x_{\mathcal{I}}^{(j)}) \quad \left\{ \begin{array}{l} \underset{x}{\text{minimize}} \quad f(x) \\ \text{subject to} \quad c(x) \leq 0 \\ x \in \mathcal{X} \quad \text{and} \quad x_{\mathcal{I}} = x_{\mathcal{I}}^{(j)}, \end{array} \right.$$

with solution $x^{(j)}$.

If $(\text{NLP}(x_{\mathcal{I}}^{(j)}))$ infeasible then solve feasibility problem ...



Outer Approximation

Convexity of f and c implies that

Lemma (Supporting Hyperplane)

Linearization about solution $x^{(j)}$ of $(NLP(x_{\mathcal{I}}^{(j)}))$

$$(OA) \quad \eta \geq f^{(j)} + \nabla f^{(j)T} (x - x^{(j)}) \quad \text{and} \quad 0 \geq c^{(j)} + \nabla c^{(j)T} (x - x^{(j)}),$$

are outer approximations of the feasible set of η -MINLP.

Lemma (Feasibility Cuts)

If $(NLP(x_{\mathcal{I}}^{(j)}))$ infeasible, then (OA) cuts off $x_{\mathcal{I}} = x_{\mathcal{I}}^{(j)}$.



Outer Approximation

Mixed-Integer Nonlinear Program (η -MINLP)

$$\min_x \eta \quad \text{s.t. } \eta \geq f(x), \quad c(x) \leq 0, \quad x \in \mathcal{X}, \quad x_i \in \mathbb{Z} \quad \forall i \in \mathcal{I}$$

Define index set of all possible feasible integers, \mathcal{F}

$$\mathcal{F} := \left\{ x^{(j)} \in \mathcal{X} : x^{(j)} \text{ solves } (\text{NLP}(x_{\mathcal{I}}^{(j)})) \text{ or } (\text{F}(x_{\mathcal{I}}^{(j)})) \right\}.$$

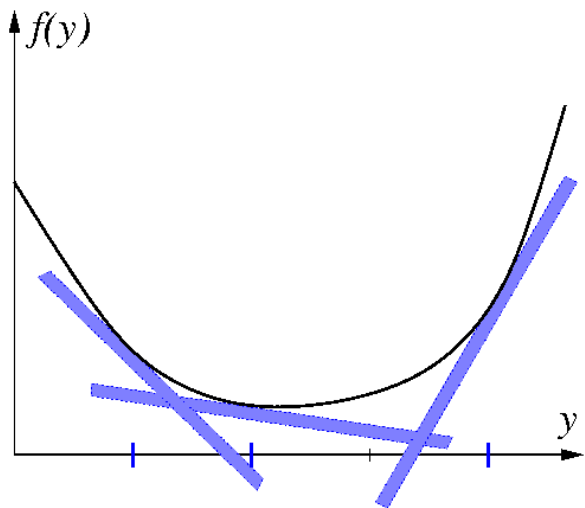
... boundedness of \mathcal{X} implies $|\mathcal{F}| < \infty$

Construct **equivalent OA-MILP** (outer approximation MILP)

$$\left\{ \begin{array}{l} \underset{\eta, x}{\text{minimize}} \quad \eta, \\ \text{subject to} \quad \eta \geq f^{(j)} + \nabla f^{(j)T} (x - x^{(j)}), \quad \forall x^{(j)} \in \mathcal{F} \\ \quad \quad \quad 0 \geq c^{(j)} + \nabla c^{(j)T} (x - x^{(j)}), \quad \forall x^{(j)} \in \mathcal{F} \\ \quad \quad \quad x \in \mathcal{X}, \\ \quad \quad \quad x_i \in \mathbb{Z}, \quad \forall i \in \mathcal{I}. \end{array} \right.$$



Outer Approximation in Less Than 1000 Words



Outer Approximation Algorithm

Solving OA-MILP clearly not sensible; define upper bound as

$$U^k := \min_{j \leq k} \left\{ f^{(j)} \mid (\text{NLP}(x_{\mathcal{I}}^{(j)})) \text{ is feasible} \right\}.$$

Define relaxation of OA-MILP, using $\mathcal{F}^k \subset \mathcal{F}$, with $\mathcal{F}^0 = \{0\}$

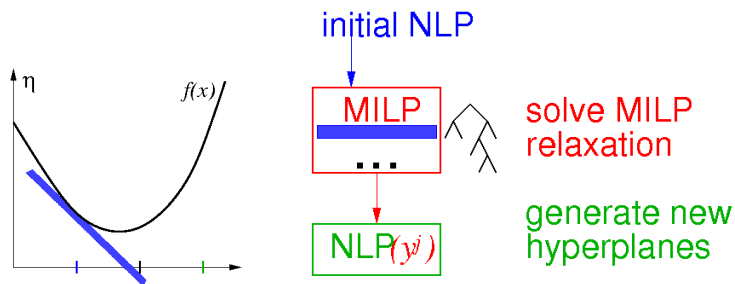
$$M(\mathcal{F}^k) \left\{ \begin{array}{l} \underset{\eta, x}{\text{minimize}} \quad \eta, \\ \text{subject to} \quad \eta \leq U^k - \epsilon \\ \quad \eta \geq f^{(j)} + \nabla f^{(j)T} (x - x^{(j)}), \quad \forall x^{(j)} \in \mathcal{F}^k \\ \quad 0 \geq c^{(j)} + \nabla c^{(j)T} (x - x^{(j)}), \quad \forall x^{(j)} \in \mathcal{F}^k \\ \quad x \in \mathcal{X}, \\ \quad x_i \in \mathbb{Z}, \quad \forall i \in \mathcal{I}. \end{array} \right.$$

... build up better OA \mathcal{F}^k iteratively for $k = 0, 1, \dots$



Outer Approximation Algorithm

Alternate between solve $\text{NLP}(y_j)$ and MILP relaxation



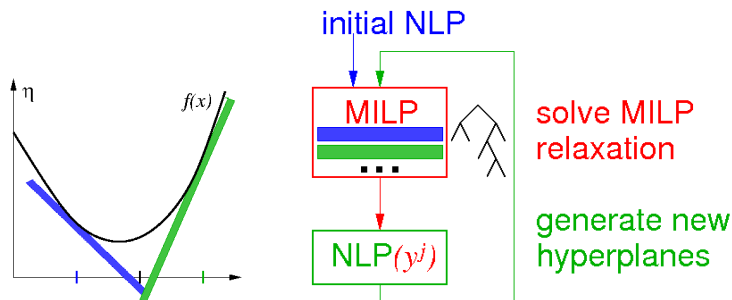
MILP \Rightarrow lower bound; NLP \Rightarrow upper bound

... convergence follows from convexity & finiteness



Outer Approximation Algorithm

Alternate between solve NLP(y_j) and MILP relaxation



MILP \Rightarrow lower bound; NLP \Rightarrow upper bound

... convergence follows from convexity & finiteness



Outer Approximation Algorithm

Outer approximation ;

Given $x^{(0)}$, choose $\text{tol } \epsilon > 0$, set $U^{-1} = \infty$, set $k = 0$, and $\mathcal{F}^{-1} = \emptyset$. ;

repeat

 Solve $(NLP(x_{\mathcal{I}}^{(j)}))$ or $(F(x_{\mathcal{I}}^{(j)}))$; solution $x^{(j)}$.;

if $(NLP(x_{\mathcal{I}}^{(j)}))$ feasible & $f^{(j)} < U^{k-1}$ **then**

 | Update best point: $x^* = x^{(j)}$ and $U^k = f^{(j)}$.;

else

 | Set $U^k = U^{k-1}$.;

end

 Linearize f and c about $x^{(j)}$ and set $\mathcal{F}^k = \mathcal{F}^{k-1} \cup \{j\}$. ;

 Solve $(M(\mathcal{F}^k))$, let solution be $x^{(k+1)}$ & set $k = k + 1$. ;

until MILP $(M(\mathcal{F}^k))$ is infeasible;



Outer Approximation Algorithm

Theorem (Convergence of Outer Approximation)

Let Assumptions A1-A3 hold, then outer approximation terminates finitely at optimal solution of MINLP or indicates it is infeasible.

Outline of Proof.

- Optimality of $x^{(j)}$ in $(\text{NLP}(x_{\mathcal{I}}^{(j)}))$
 $\Rightarrow \eta \geq f^{(j)}$ for feasible point of $(M(\mathcal{F}^k))$
... ensures finiteness, since \mathcal{X} compact
- Convexity \Rightarrow linearizations are supporting hyperplanes
... ensures optimality



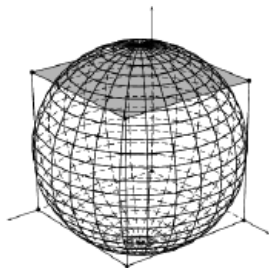
Worst Case Example of Outer Approximation

[Hijazi et al., 2010] construct **infeasible** MINLP:

minimize 0
 y

$$\text{subject to } \sum_{i=1}^n \left(y_i - \frac{1}{2} \right)^2 \leq \frac{n-1}{4}$$
$$y \in \{0, 1\}^n$$

Intersection of ball of radius $\frac{\sqrt{n-1}}{2}$
with unit hypercube.



Lemma

*OA cannot cut more than one vertex of the hypercube
MILP master problem feasible for any $k < 2^n$ OA cuts*

Theorem

OA visits all 2^n vertices

Benders Decomposition

Can derive Benders cut from outer approximation:

- Take **optimal multipliers** $\lambda^{(j)}$ of $(\text{NLP}(x_{\mathcal{I}}^{(j)}))$
- Sum outer approximations

$$\begin{array}{r} \eta \geq f^{(j)} + \nabla f^{(j)T} (x - x^{(j)}) \\ + \lambda^{(j)T} (0 \geq c^{(j)} + \nabla c^{(j)T} (x - x^{(j)})) \\ \hline \eta \geq f^{(j)} + \nabla_{\mathcal{I}} \mathcal{L}^{(j)T} (x_{\mathcal{I}} - x_{\mathcal{I}}^{(j)}) \end{array}$$

- Using KKT conditions wrt continuous variables $x_{\mathcal{C}}$:
 $0 = \nabla_{\mathcal{C}} \mathcal{L}^{(j)} = \nabla_{\mathcal{C}} f + \nabla_{\mathcal{C}} c \lambda^{(j)}$ & $\lambda^{(j)T} c^{(j)} = 0$
... eliminates continuous variables, $x_{\mathcal{C}}$

Benders cut only involves integer variables $x_{\mathcal{I}}$.

Can write cut as $\eta \geq f^{(j)} + \mu^{(j)T} (x_{\mathcal{I}} - x_{\mathcal{I}}^{(j)})$,
where $\mu^{(j)}$ multiplier of $x = x_{\mathcal{I}}^{(j)}$ in $(\text{NLP}(x_{\mathcal{I}}^{(j)}))$



Benders Decomposition

For MINLPs with convex problems functions f , c , we can show:

- 1 Benders cuts are weaker than outer approximation
 - Benders cuts are linear combination of OA
- 2 Outer Approximation & Benders converge finitely
 - Functions f , c convex \Rightarrow OA cuts are outer approximations
 - OA cut derived at optimal solution to NLP subproblem
 - \Rightarrow ∇ feasible descent directions
 - ... every OA cut corresponds to first-order condition
 - Cannot visit same integer $x_{\mathcal{I}}^{(j)}$ more than once

\Rightarrow terminate finitely at optimal solution

Readily extended to situations where $(\text{NLP}(x_{\mathcal{I}}^{(j)}))$ not feasible.



Summary of Multi-Tree Methods

Three Classes of Multi-Tree Methods (did not discuss ECP)

- 1 Outer approximation based on first-order expansion
- 2 Benders decomposition linear combination of OA cuts
- 3 Extended cutting plane method: avoids NLP solves

Common Properties of Multi-Tree Methods

- Only need to solve final MILP to optimality
... can terminate MILP early ... adding more NLPs
- Can add cuts from incomplete NLP solves
- Worst-case example for OA also applies for Benders and ECP
- **No warm-starts for MILP ... expensive tree-search**

... motivates single-tree methods next ...



Outline

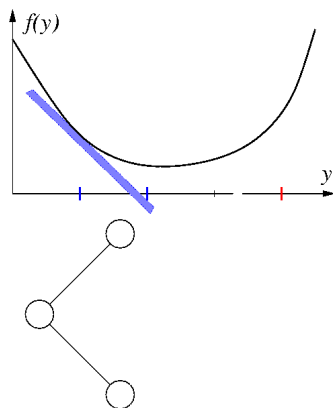
- 1 Multi-Tree Methods
- 2 Single-Tree Methods
- 3 Presolve for MINLP



LP/NLP-Based Branch-and-Bound

Aim: avoid **solving expensive MILPs**

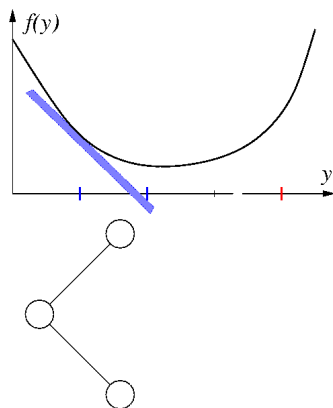
- Form MILP outer approximation



LP/NLP-Based Branch-and-Bound

Aim: avoid **solving expensive MILPs**

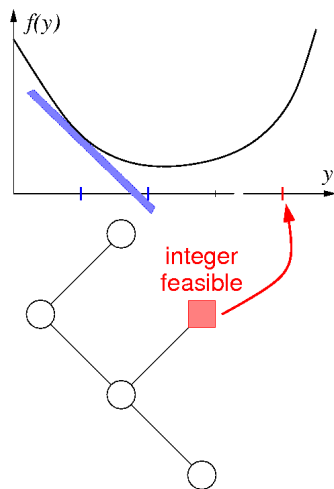
- Form MILP outer approximation
- Take initial MILP tree



LP/NLP-Based Branch-and-Bound

Aim: avoid **solving expensive MILPs**

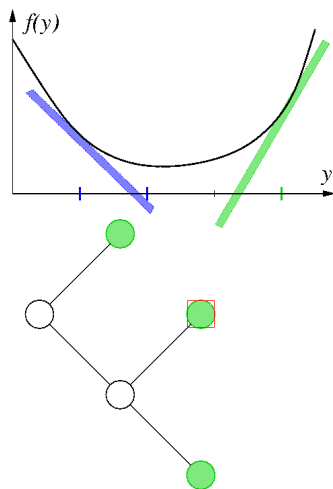
- Form MILP outer approximation
- Take initial MILP tree
- **interrupt MILP**, when new integral $x_I^{(j)}$ found
⇒ solve NLP($x_I^{(j)}$) get $x^{(j)}$



LP/NLP-Based Branch-and-Bound

Aim: avoid **solving expensive MILPs**

- Form MILP outer approximation
- Take initial MILP tree
- **interrupt MILP**, when new integral $x_I^{(j)}$ found
⇒ solve NLP($x_I^{(j)}$) get $x^{(j)}$
- linearize f, c about $x^{(j)}$
⇒ **add linearization to tree**



LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

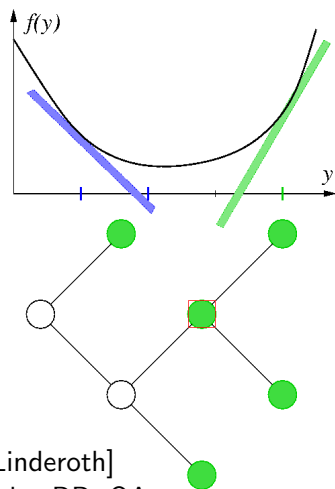
- Form MILP outer approximation
- Take initial MILP tree
- interrupt MILP, when new integral $x_I^{(j)}$ found
⇒ solve NLP($x_I^{(j)}$) get $x^{(j)}$
- linearize f , c about $x^{(j)}$
⇒ add linearization to tree
- continue MILP tree-search

... until lower bound \geq upper bound

Software:

FilMINT: FilterSQP + MINTO [L & Linderoth]

BONMIN: IPOPT + CBC [IBM/CMU] also BB, OA



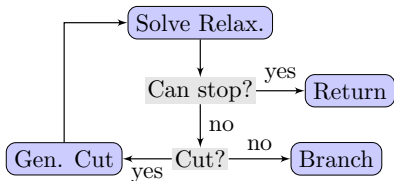
Branch-and-Cut in MINOTAUR

Suppose we need a **branch-and-cut** solver.

Node Relaxer

Obtain linear relaxation in root node.

Node Processor



Brancher

Pick a fractional variable.

Only
CxLinHandler

CxLinHandler
IntVarHandler

Only IntVarHandler

```
relax() {  
  // Solve NLP  
  // get Linearization at sol.  
}  
bool isFeasible() {  
  // check non-linear constraints
```

```
separate() {  
  // solve NLP  
  // get Linearization at sol.  
}  
cand* findBrCandidates() {  
  // empty
```



LP/NLP-Based Branch-and-Bound

Algorithmic refinements, e.g. [Abhishek et al., 2010]

- Advanced MILP search and **cut management techniques**
... remove “old” OA cuts from LP relaxation \Rightarrow faster LP
- Generate cuts at non-integer points: ECP cuts are cheap
... generate cuts early (near root) of tree
- Strong branching, adaptive node selection & **cut management**
 - Fewer nodes, if we add more cuts (e.g. ECP cuts)
 - More cuts make LP harder to solve
 \Rightarrow remove outdated/inactive cuts from LP relaxation
... balance OA accuracy with LP solvability
- Compress OA cuts into Benders cuts can be OK

Interpret as hybrid algorithm, [Bonami et al., 2008]

Benders and ECP versions are also possible.



Outline

- 1 Multi-Tree Methods
- 2 Single-Tree Methods
- 3 Presolve for MINLP



Presolve for MINLP

Presolve plays key role in MILP solvers

- Bound tightening techniques
- Checking for duplicate rows
- Fixing or removing variables
- Identifying redundant constraints

... creates tighter LP/NLP relaxations \Rightarrow smaller trees!

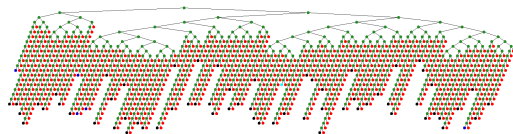
... some presolve in AMPL, but no nonlinear presolve



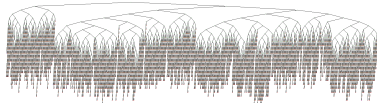
What Could Go Wrong in MINLP?

Syn20M04M: a synthesis design problem
in chemical engineering

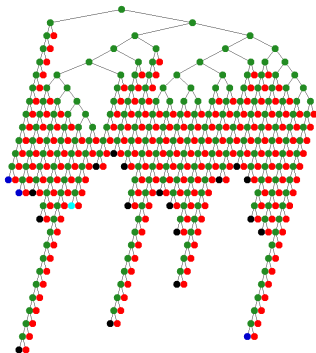
Problem size: 160 Integer Variables,
56 Nonlinear constraints



1000+ nodes after solving for 75s



5000+ nodes after solving for 200s



250+ nodes after solving for 45s

Solver	CPU	Nodes
Bonmin	>2h	>149k
MINLPBB	>2h	>150k
Minotaur	>2h	>264k

Improving Coefficients: An Example

$$(1) \quad x_1 + 21x_2 \leq 30$$

$$0 \leq x_1 \leq 14$$

$$x_2 \in \{0, 1\}$$

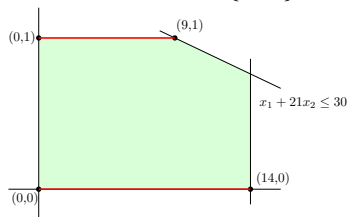


Improving Coefficients: An Example

$$(1) \quad x_1 + 21x_2 \leq 30$$

$$0 \leq x_1 \leq 14$$

$$x_2 \in \{0, 1\}$$



$$\boxed{\text{If } x_2 = 0}$$

$$x_1 + 0 \leq 30$$

(1) is loose.

$$\boxed{\text{If } x_2 = 1}$$

$$x_1 \leq 9$$

(1) is tight.

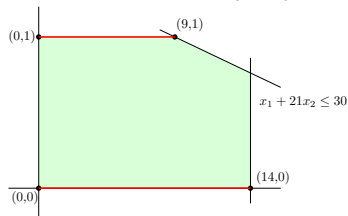


Improving Coefficients: An Example

$$(1) \quad x_1 + 21x_2 \leq 30$$

$$0 \leq x_1 \leq 14$$

$$x_2 \in \{0, 1\}$$



$$\boxed{\text{If } x_2 = 0}$$

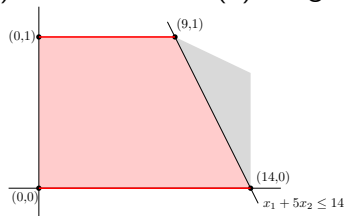
$$x_1 + 0 \leq 30$$

(1) is loose.

$$\boxed{\text{If } x_2 = 1}$$

$$x_1 \leq 9$$

(1) is tight.

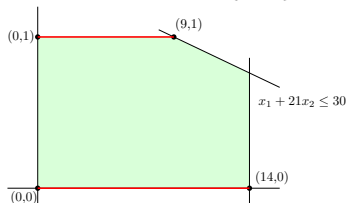


Improving Coefficients: An Example

$$(1) \quad x_1 + 21x_2 \leq 30$$

$$0 \leq x_1 \leq 14$$

$$x_2 \in \{0, 1\}$$



Reformulation:

$$(2) \quad x_1 + 5x_2 \leq 14$$

$$0 \leq x_1 \leq 14$$

$$x_2 \in \{0, 1\}$$

$$\text{If } x_2 = 0$$

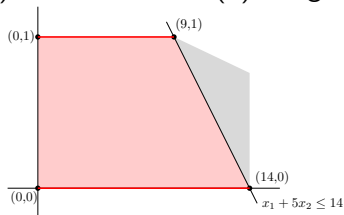
$$x_1 + 0 \leq 30$$

(1) is loose.

$$\text{If } x_2 = 1$$

$$x_1 \leq 9$$

(1) is tight.



$$\text{If } x_2 = 0$$

$$x_1 + 0 \leq 14$$

(2) is tight.

$$\text{If } x_2 = 1$$

$$x_1 \leq 9$$

(2) is tight.

(1) and (2) equivalent. But relaxation of (2) is tighter.



Improving Coefficients: Linear to Nonlinear

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0) \\ l_i &\leq x_i \leq u_i, \quad i = 1, \dots, k \\ x_0 &\in \{0, 1\}\end{aligned}$$



Improving Coefficients: Linear to Nonlinear

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0) \\l_i &\leq x_i \leq u_i, \quad i = 1, \dots, k \\x_0 &\in \{0, 1\}\end{aligned}$$

- If $c(x_1, x_2, \dots, x_k) \leq M(1 - 0)$, is loose, tighten it!

$$\begin{aligned}\text{Let } c^u &= \max_x c(x_1, \dots, x_k) && \text{(MAX-c)} \\ \text{s.t. } &l_i \leq x_i \leq u_i, \quad i = 1, \dots, k\end{aligned}$$

- If $c^u < M$, then tighten: $c(x_1, \dots, x_k) \leq c^u(1 - x_0)$



Improving Coefficients: Linear to Nonlinear

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0) \\ l_i &\leq x_i \leq u_i, \quad i = 1, \dots, k \\ x_0 &\in \{0, 1\}\end{aligned}$$

- If $c(x_1, x_2, \dots, x_k) \leq M(1 - 0)$, is loose, tighten it!

$$\begin{aligned}\text{Let } c^u &= \max_x c(x_1, \dots, x_k) && \text{(MAX-c)} \\ \text{s.t. } & l_i \leq x_i \leq u_i, \quad i = 1, \dots, k\end{aligned}$$

- If $c^u < M$, then tighten: $c(x_1, \dots, x_k) \leq c^u(1 - x_0)$
- (MAX-c) is a **nonconvex NLP** ... time-consuming
- Upper bound on (MAX-c) will also tighten
- Trade-off between time and quality of bound: Fast or Tight!

Improving Coefficients: Using Implications

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0), \\l_i &\leq x_i \leq u_i, \quad i = 1, \dots, k, \\x_0 &\in \{0, 1\}.\end{aligned}$$

- Often, x_0, x_i also occur in other constraints of MINLP. e.g.

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0) \\0 &\leq x_1 \leq M_1 x_0 \\0 &\leq x_2 \leq M_2 x_0 \\&\dots \\x_0 &\in \{0, 1\}\end{aligned}$$



Improving Coefficients: Using Implications

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0), \\ l_i &\leq x_i \leq u_i, \quad i = 1, \dots, k, \\ x_0 &\in \{0, 1\}.\end{aligned}$$

- Often, x_0, x_i also occur in other constraints of MINLP. e.g.

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0) \\ 0 &\leq x_1 \leq M_1 x_0 \\ 0 &\leq x_2 \leq M_2 x_0 \\ &\dots \\ x_0 &\in \{0, 1\}\end{aligned}$$

- $x_0 = 0 \Rightarrow x_1 = x_2, \dots = x_k = 0$. (**Implications**)
- If $c(0, \dots, 0) < M$, then we can tighten.



Improving Coefficients: Using Implications

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0), \\l_i &\leq x_i \leq u_i, \quad i = 1, \dots, k, \\x_0 &\in \{0, 1\}.\end{aligned}$$

- Often, x_0, x_i also occur in other constraints of MINLP. e.g.

$$\begin{aligned}c(x_1, x_2, \dots, x_k) &\leq M(1 - x_0) \\0 &\leq x_1 \leq M_1 x_0 \\0 &\leq x_2 \leq M_2 x_0 \\&\dots \\x_0 &\in \{0, 1\}\end{aligned}$$

- $x_0 = 0 \Rightarrow x_1 = x_2, \dots = x_k = 0$. (Implications)
- If $c(0, \dots, 0) < M$, then we can tighten.
- No need to solve (MAX-c). Fast and Tight.

Presolve for MINLP

Advanced functions of presolve (Reformulating):

- Improve coefficients.
- Disaggregate constraints.
- Derive implications and conflicts.

Basic functions of presolve (Housekeeping):

- Tighten bounds on variables and constraints.
- Fix/remove variables.
- Identify and remove redundant constraints.
- Check duplicacy.

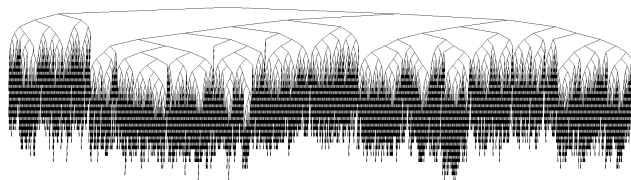
Popular in Mixed-Integer Linear Optimization [[Savelsbergh, 1994](#)]



Presolve for MINLP: Computational Results

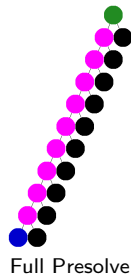
Syn20M04M from `egon.cheme.cmu.edu`

	No Presolve	Basic Presolve	Full Presolve
Variables:	420	328	292
Binary Vars:	160	144	144
Constraints:	1052	718	610
Nonlin. Constr:	56	56	56
Bonmin(sec):	>7200	NA	NA
Minotaur(sec):	>7200	>7200	2.3



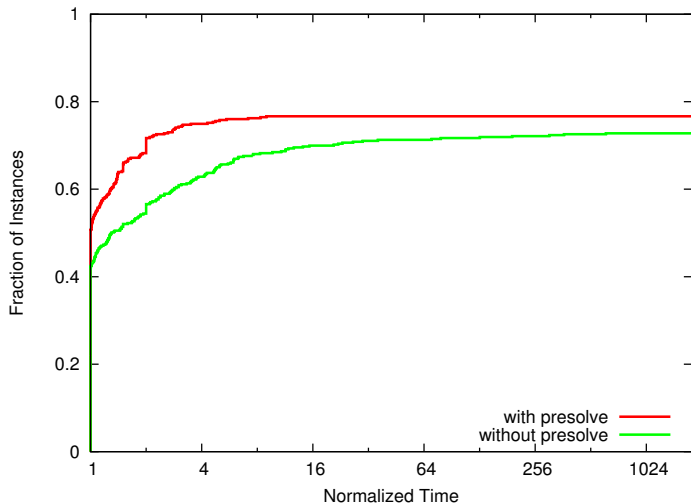
Minotaur, no presolve: 10000+ nodes after solving for 360s

Why does no one else do this?



Full Presolve

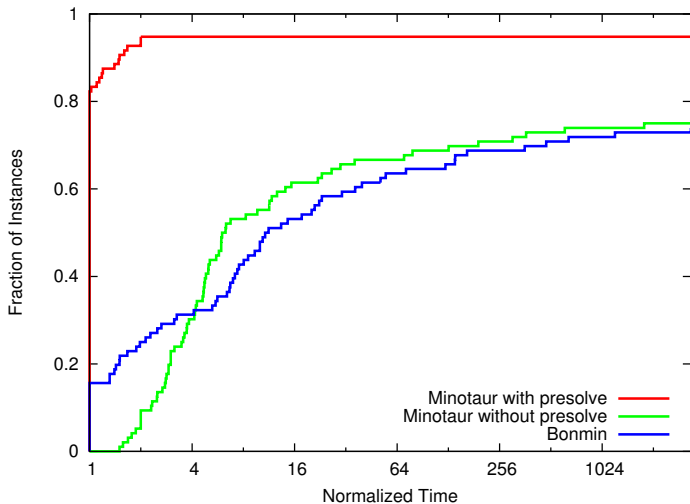
Presolve for MINLP: Results



Time taken in Branch-and-Bound on all 463 instances.



Presolve for MINLP: Results



Time for B&B on 96 RSyn-X and Syn-X instances.



Presolve for MINLP: Constraint Disaggregation

Nonlinear disaggregation [Tawarmalani and Sahinidis, 2005]

$$S := \{x \in \mathbb{R}^n : c(x) = h(g(x)) \leq 0\},$$

$g : \mathbb{R}^n \rightarrow \mathbb{R}^p$ smooth convex;

$h : \mathbb{R}^p \rightarrow \mathbb{R}$ smooth, convex, and **nondecreasing**

$\Rightarrow c(x)$ smooth convex

Disaggregated formulation: introduce $y = g(x) \in \mathbb{R}^p$

$$S_d := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : h(y) \leq 0, y \geq g(x)\}.$$

Lemma

S is projection of S_d onto x .



Presolve for MINLP: Constraint Disaggregation

Consider

$$S := \{x \in \mathbb{R}^n : c(x) = h(g(x)) \leq 0\},$$

and

$$S_d := \{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : h(y) \leq 0, y \geq g(x)\}.$$

Theorem

Any outer approximation of S_d is stronger than OA of S

Given $\mathcal{X}^k := \{x^{(1)}, \dots, x^{(k)}\}$ construct OA for S, S_d :

$$S^{oa} := \{x : c^{(l)} + \nabla c^{(l)T} (x - x^{(l)}) \leq 0, \forall x^{(l)} \in \mathcal{X}^k\}$$

$$S_d^{oa} := \{(x, y) : h^{(l)} + \nabla h^{(l)T} (y - g(x^{(l)})) \leq 0, \\ y \geq g^{(l)} + \nabla g^{(l)T} (x - x^{(l)}), \forall x^{(l)} \in \mathcal{X}^k\},$$

[Tawarmalani and Sahinidis, 2005] show S_d^{oa} stronger than S^{oa}

Presolve for MINLP: Constraint Disaggregation

[Hijazi et al., 2010] study

$$\left\{ x : c(x) := \sum_{j=1}^q h_j(a_j^T x + b_j) \leq 0 \right\}$$

where $h_j : \mathbb{R} \rightarrow \mathbb{R}$ are smooth and convex

Disaggregated formulation: introduce $y \in \mathbb{R}^q$

$$\left\{ (x, y) : \sum_{j=1}^q y_j \leq 0, \text{ and } y_j \geq h_j(a_j^T x + b_j) \right\}$$

can be shown to be tighter



Recall: Worst Case Example of OA

Apply disaggregation to [Hijazi et al., 2010] example:

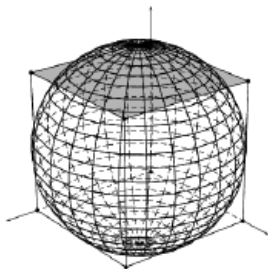
minimize 0
 y

$$\text{subject to } \sum_{i=1}^n \left(x_i - \frac{1}{2}\right)^2 \leq \frac{n-1}{4}$$
$$x \in \{0, 1\}^n$$

Intersection of ball of radius $\frac{\sqrt{n-1}}{2}$
with unit hypercube.

Disaggregate $\sum (x_i - \frac{1}{2})^2 \leq \frac{n-1}{4}$ as

$$\sum_{i=1}^n y_i \leq 0 \quad \text{and} \quad \left(x_i - \frac{1}{2}\right)^2 \leq y_i$$



Presolve for MINLP: Constraint Disaggregation

[Hijazi et al., 2010] disaggregation on worst-case example of OA

- Linearize around $x^{(1)} \in \{0, 1\}^n$ and complement $x^{(2)} := e - x^{(1)}$, where $e = (1, \dots, 1)$
- OA of disaggregated constraint is

$$\sum_{i=1}^n y_i, \quad \text{and} \quad x_i - \frac{3}{4} \leq y_i, \quad \text{and} \quad \frac{1}{4} - x_i \leq y_i,$$

- Using $x_i \in \{0, 1\}$ implies $z_i \geq 0$, implies $\sum z_i \geq \frac{n}{4} > \frac{n-1}{4}$
- \Rightarrow OA-MILP master of $x^{(1)}$ and $x^{(2)}$ is infeasible.
... terminate in two iterations



Teaching Points

Multi-Tree Methods

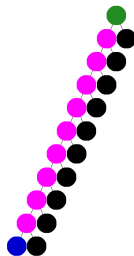
- Convexity \Rightarrow outer approximation to create MILP
- Iterate between MILP master and NLP
- Good when MINLP “almost” linear

Single-Tree or Hybrid Methods

- Integrate NLP solves within MILP tree-search
- Can be implemented effectively
 - ... cut management is needed

Presolve is Important

- Presolve can be critical for MINLP
- Simple tricks help
 - ... need access to computational graph





Abhishek, K., Leyffer, S., and Linderoth, J. T. (2010).

FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs.

INFORMS Journal on Computing, 22:555–567.

DOI:10.1287/ijoc.1090.0373.



Bonami, P., Biegler, L., Conn, A., Cornuéjols, G., Grossmann, I., Laird, C., Lee, J., Lodi, A., Margot, F., Sawaya, N., and Wächter, A. (2008).

An algorithmic framework for convex mixed integer nonlinear programs.

Discrete Optimization, 5(2):186–204.



Duran, M. A. and Grossmann, I. (1986).

An outer-approximation algorithm for a class of mixed-integer nonlinear programs.

Mathematical Programming, 36:307–339.



Geoffrion, A. M. (1972).

Generalized Benders decomposition.

Journal of Optimization Theory and Applications, 10(4):237–260.



Griewank, A. and Toint, P. L. (1984).

On the existence of convex decompositions of partially separable functions.

Mathematical Programming, 28:25–49.



Hijazi, H., Bonami, P., and Ouorou, A. (2010).

An outer-inner approximation for separable MINLPs.

Technical report, LIF, Faculté des Sciences de Luminy, Université de Marseille.



Savelsbergh, M. W. P. (1994).



Preprocessing and probing techniques for mixed integer programming problems.
ORSA Journal on Computing, 6:445–454.



Tawarmalani, M. and Sahinidis, N. V. (2005).
A polyhedral branch-and-cut approach to global optimization.
Mathematical Programming, 103(2):225–249.



Westerlund, T. and Pettersson, F. (1995).
A cutting plane method for solving convex MINLP problems.
Computers & Chemical Engineering, 19:s131–s136.



Wolsey, L. A. (1998).
Integer Programming.
John Wiley and Sons, New York.

