Mixed-Integer Nonlinear Optimization: Applications, Algorithms, and Computation III

Sven Leyffer

Mathematics & Computer Science Division
Argonne National Laboratory

Graduate School in
Systems, Optimization, Control and Networks
Université catholique de Louvain
February 2013

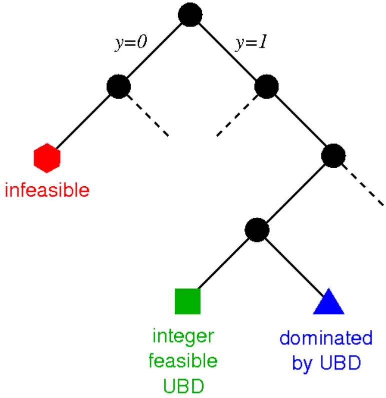
Outline

Recall: Nonlinear Branch-and-Bound

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

Solve continuous relaxation (NLP) ($0 \le x_I \le 1$)
...solution value provides lower bound

- Branch on x_i non-integral
- Solve NLPs & branch until
  1. Node infeasible: ●
  2. Node integer feasible: □
     $\Rightarrow$ get upper bound (U)
  3. Lower bound $\ge U$: ▲

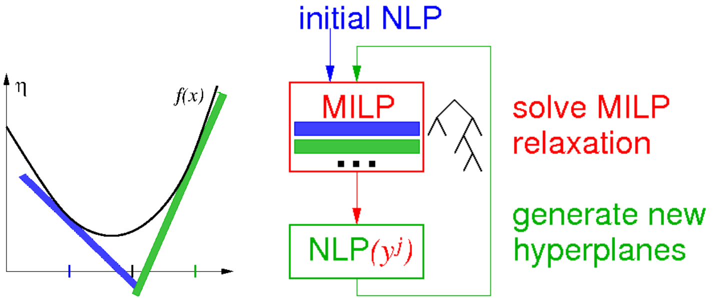


Search until no unexplored nodes

Snag: Solve thousands of NLPs ...

Recall: Outer Approximation

Alternate between solve NLP(x_I) and MILP relaxation



MILP $\Rightarrow$ lower bound; NLP $\Rightarrow$ upper bound

Snag: Solve multiple MILPs ...

Outline

Single-Tree Methods

Goal: perform only a single MILP tree-search per MINLP

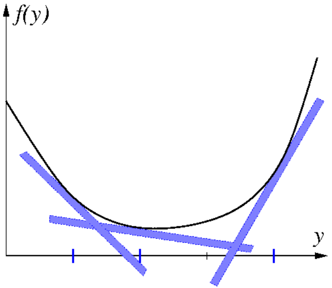
- Branch-and-Bound is s single-tree method
  ... but can be too expensive per node
- Avoid re-solving MILP master for OA, Benders, and ECP
  ... instead update master (MILP) data
- Can be interpreted as branch-and-cut approach
  ... but cuts are very simple
- Solve MILP with full set of linearizations $\mathcal{X}$ and apply delayed constraint generation technique of "formulation constraints" $\mathcal{X}^k \subset \mathcal{X}$.
- At integer points, separate cuts by solving an NLP

... basis for state-of-the-art convex MINLP solvers

LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

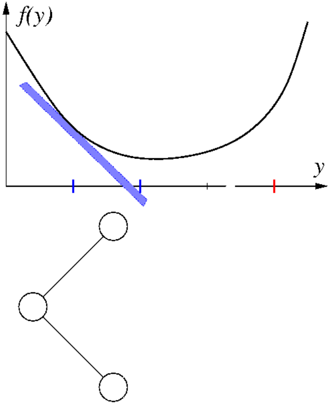
- Form MILP outer approximation



LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

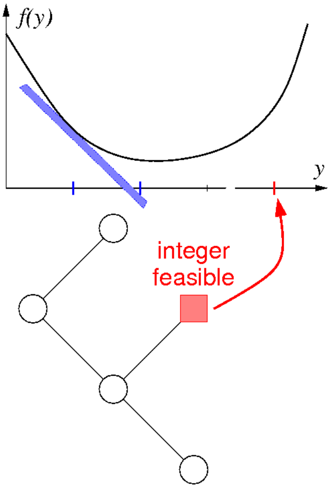
- Form MILP outer approximation
- Take initial MILP tree



LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

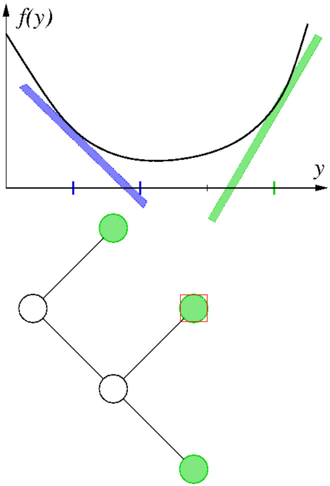
- Form MILP outer approximation
- Take initial MILP tree
- interrupt MILP, when new integral $x_I^{(j)}$ found
  $\Rightarrow$ solve NLP($x_I^{(j)}$) get $x^{(j)}$



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
  $\Rightarrow$ solve NLP($x_I^{(j)}$) get $x^{(j)}$
- linearize f, c about $x^{(j)}$
  $\Rightarrow$ add linearization to tree

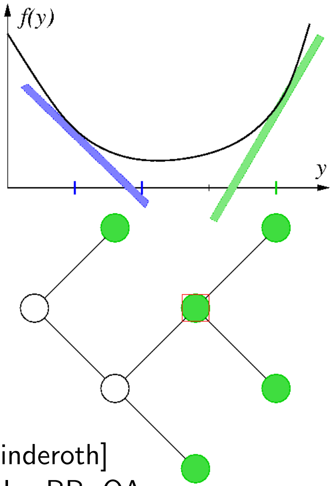


LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

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

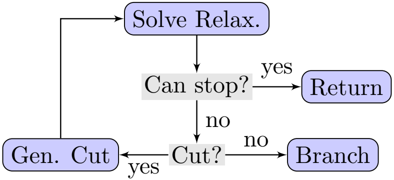
Suppose we need a branch-and-cut solver.

Node Relaxer

Obtain linear relaxation in root node.

Only CxLinHandler

Node Processor



CxLinHandler
IntVarHandler

Brancher

Pick a fractional variable.

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 ⇒ 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
    ⇒ 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

Presolve for MINLP

Presolve plays key role in MILP solvers

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

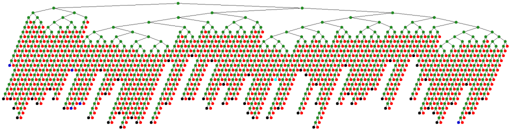
... creates tighter LP/NLP relaxations $\Rightarrow$ smaller trees!

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

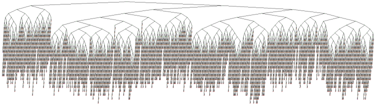
What Could Go Wrong in MINLP?

Syn20M04M : a synthesis design problem in chemical engineering

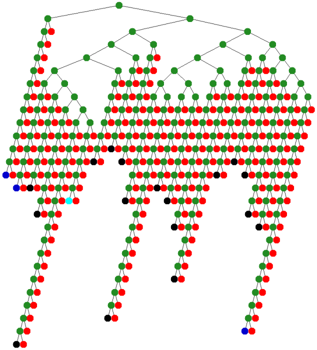
Problem size: 160 Integer Variables, 56 Nonlinear constraints



1000+ nodes after solving for 75s



5000+ nodes after solving for 200s



250+ nodes after solving for 45s

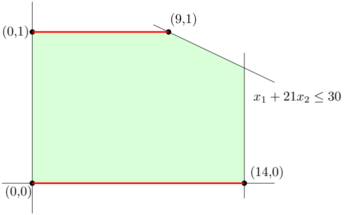
Solver	CPU	Nodes
Bonmin	>2h	>149k
MINLPBB	>2h	>150k
Minotaur	>2h	>264k

Improving Coefficients: An Example

$$
\begin{aligned}
(1) \quad x_1 + 21x_2 &\leq 30 \\
0 \leq x_1 &\leq 14 \\
x_2 &\in \{0, 1\}
\end{aligned}
$$

Improving Coefficients: An Example

$$(1) \quad x_1 + 21x_2 \le 30$$
$$0 \le x_1 \le 14$$
$$x_2 \in \{0, 1\}$$



If $x_2 = 0$	If $x_2 = 1$
$x_1 + 0 \le 30$	$x_1 \le 9$
(1) is loose.	(1) is tight.

Improving Coefficients: An Example

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

$0 \leq x_1 \leq 14$

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

If $x_2 = 0$

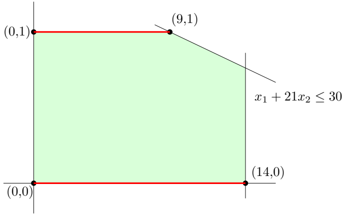
$x_1 + 0 \leq 30$

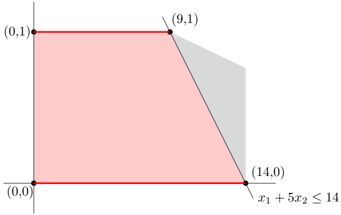
(1) is loose.

If $x_2 = 1$

$x_1 \leq 9$

(1) is tight.





Improving Coefficients: An Example

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

$0 \leq x_1 \leq 14$

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

If $x_2 = 0$

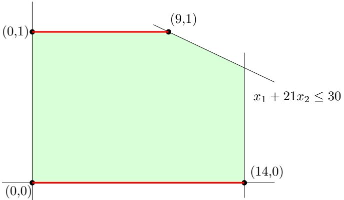
$x_1 + 0 \leq 30$

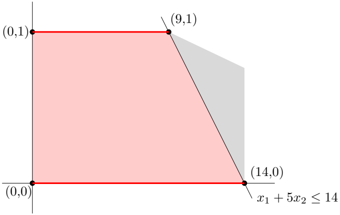
(1) is loose.

If $x_2 = 1$

$x_1 \leq 9$

(1) is tight.





Reformulation:

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

$0 \leq x_1 \leq 14$

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

If $x_2 = 0$

$x_1 + 0 \leq 14$

(2) is tight.

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

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

Improving Coefficients: Linear to Nonlinear

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

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

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

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

Improving Coefficients: Linear to Nonlinear

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

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

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

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

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

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

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

Improving Coefficients: Using Implications

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

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

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

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

Improving Coefficients: Using Implications

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

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

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

- $x_0 = 0 \Rightarrow x_1 = x_2, \ldots = x_k = 0$. (Implications)
- If $c(0, \ldots, 0) < M$, 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):

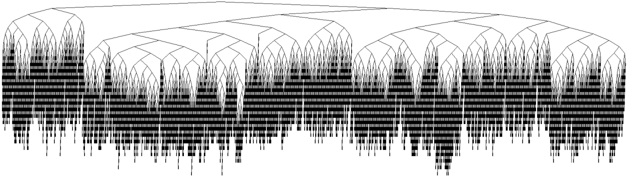
- Tighten bounds on variables and constraints.
- Fix/remove variables.
- Identify and remove redundant constraints.
- Check duplicacy.

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

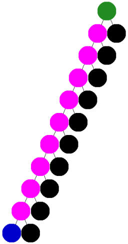
Presolve for MINLP: Computational Results

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

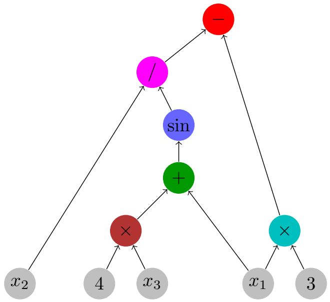


Full Presolve

Why does no one else do this?

Why Does No One else Do It? . . . Better AD!

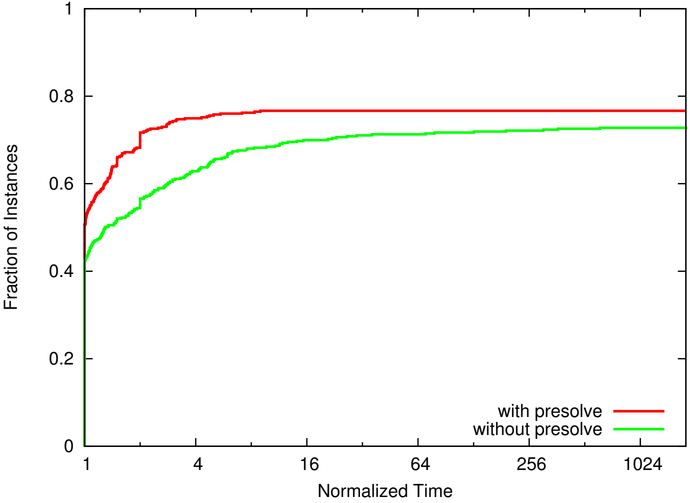
- NLP solvers need 1^{st} and 2^{nd} derivatives
- Rely on modeling software: AMPL, GAMS
  $\Rightarrow$ cannot modify functions during solve
- Minotaur has routines to
  - create computational graphs,
  - evaluate 1^{st} and 2^{nd} derivatives,
  - tighten and propagate bounds,
  - modify graphs.
- Simple modification routines:
  - Fix and delete variables.
  - Substitute variables.
  - Extract subgraphs.



$$f = \frac{x_2}{\sin(4 \times x_3 + x_1)} - 3 \times x_1$$

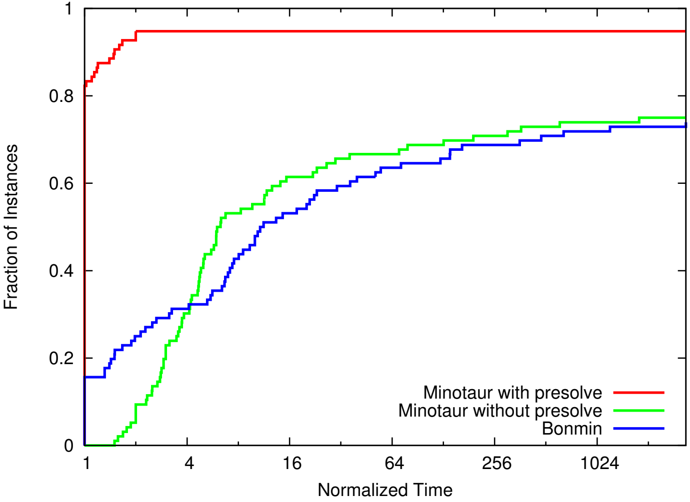
Scope for more improvements

Presolve for MINLP: Results



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

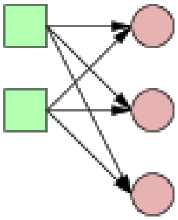
Presolve for MINLP: Results



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

Presolve for MINLP: Constraint Disaggregation

[Wolsey, 1998] uncapacitated facility location



- Set of customers $i = 1, \dots, m$
- Set of facilities $j = 1, \dots, n$
- Which facilities should we open ($x_j \in \{0, 1\}, \; j = 1, \dots, n$)
- $y_{ij} = 1$ if facility j serves customer i

Every customer served by one facility:

$$\sum_{j=1}^{n} y_{ij} = 1, \; \forall i = 1, \dots, m, \quad \text{and} \quad \sum_{i=1}^{m} y_{ij} \leq m x_j, \; \forall j = 1, \dots, n,$$

Equivalent tighter formulation is (disagregated constraints):

$$\sum_{j=1}^{n} y_{ij} = 1, \; \forall i = 1, \dots, m, \quad \text{and} \quad y_{ij} \leq x_j, \; \forall i = 1, \dots, m, \; j = 1, \dots, n.$$

... modern MIP solvers detect this automatically

Presolve for MINLP: Constraint Disaggregation

Nonlinear disaggregation [Tawarmalani and Sahinidis, 2005]

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

$g : \mathbb{R}^n \to \mathbb{R}^p$ smooth convex;
$h : \mathbb{R}^p \to \mathbb{R}$ smooth, convex, and nondecreasing
$\Rightarrow c(x)$ smooth convex

Like group partial separability [Griewank and Toint, 1984]

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

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

Lemma

S is projection of S_d onto x.

Presolve for MINLP: Constraint Disaggregation

Consider

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

and

$$S_d := \left\{(x, y) \in \mathbb{R}^n \times \mathbb{R}^p : h(y) \le 0,\ y \ge g(x)\right\}.$$

Theorem

Any outer approximation of S_d is stronger than OA of S

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

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

$$S_d^{oa} := \left\{(x, y) : h^{(l)} + \nabla h^{(l)^T}(y - g(x^{(l)})) \le 0,\right.$$

$$\left. y \ge g^{(l)} + \nabla g^{(l)^T}(x - x^{(l)}),\ \forall x^{(l)} \in \mathcal{X}^k\right\},$$

[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 $h_j : \mathbb{R} \to \mathbb{R}$ are smooth and convex

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

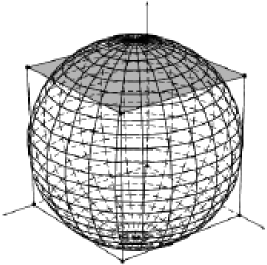
$$\left\{ (x, y) : \sum_{j=1}^{q} y_j \leq 0, \quad \text{and} \quad y_j \geq h_j(a_j^T x + b_j) \right\}$$

can be shown to be tighter

Recall: Worst Case Example of OA

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

$$\begin{aligned} \underset{y}{\text{minimize}} \quad & 0 \\ \text{subject to} \quad & \sum_{i=1}^{n} \left(x_i - \frac{1}{2}\right)^2 \leq \frac{n-1}{4} \\ & x \in \{0,1\}^n \end{aligned}$$



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

Disaggregate $\sum \left(x_i - \frac{1}{2}\right)^2 \leq \frac{n-1}{4}$ 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 $e = (1, \ldots, 1)$
- OA of disaggregated constraint is

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

- 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

Outline

Mixed-Integer Nonlinear Optimization

Mixed-Integer Nonlinear Program (MINLP)

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) \leq 0 \\ & x \in X \\ & x_i \in \mathbb{Z} \text{ for all } i \in I \end{aligned}$$

Assumptions:

- A1 X is a bounded polyhedral set.
- A2 f and c are twice continuously differentiable convex functions.
- A3 MINLP satisfies a constraint qualification.

Look at another class of branch-and-cut methods ...

Overview of Branch-and-Cut Methods

Extend nonlinear branch-and-bound

1. Solve NLP(l, u) at each node of tree
    - Generate a cut to eliminate fractional solution & re-solve
    - Only branch if solution fractional after some rounds of cuts
2. Generation of good cuts is key [Stubbs and Mehrotra, 1999]
3. Hope that tree is smaller than BnB
4. Goal: get formulation closer to convex hull

Recall Nonlinear Branch-and-Bound

Solve NLP relaxation

$$\underset{x}{\text{minimize}} \; f(x) \quad \text{subject to } c(x) \le 0, \; x \in X$$

- If $x_i \in \mathbb{Z} \; \forall \; i \in I$, then solved MINLP
- If relaxation is infeasible, then MINLP infeasible

... otherwise search tree whose nodes are NLPs:

$$\left\{ \begin{array}{ll} \underset{x}{\text{minimize}} & f(x), \\ \text{subject to} & c(x) \le 0, \\ & x \in X, \\ & l_i \le x_i \le u_i, \; \forall i \in I. \end{array} \right. \qquad (\text{NLP}(l, u))$$

NLP relaxation is $\text{NLP}(-\infty, \infty)$

Recall Nonlinear Branch-and-Bound

Branch-and-bound for MINLP

Choose tol $\epsilon > 0$, set $U = \infty$, add (NLP$(-\infty, \infty)$) to heap $\mathcal{H}$.

while $\mathcal{H} \neq \emptyset$ **do**

  Remove (NLP(l, u)) from heap: $\mathcal{H} = \mathcal{H} - \{$ NLP(l, u) $\}$.

  Solve (NLP(l, u)) $\Rightarrow$ solution $x^{(l,u)}$.

  if *(NLP(l, u)) is infeasible* **then**

    Prune node: infeasible

  else if $f(x^{(l,u)}) > U$ **then**

    Prune node; dominated by bound U

  else if $x_I^{(l,u)}$ *integral* **then**

    Update incumbent : $U = f(x^{(l,u)})$, $x^* = x^{(l,u)}$.

  else

    BranchOnVariable$(x_i^{(l,u)}, l, u, \mathcal{H})$

Generic Nonlinear Branch-and-Cut

Branch-and-cut for MINLP

Choose a tol $\epsilon > 0$, set $U = \infty$, add (NLP($-\infty, \infty$)) to heap $\mathcal{H}$.

while $\mathcal{H} \neq \emptyset$ **do**

Remove (NLP(l, u)) from heap: $\mathcal{H} = \mathcal{H} - \{$ NLP(l, u) $\}$.

repeat

Solve (NLP(l, u)) $\Rightarrow$ solution $x^{(l,u)}$.

if *(NLP(l, u)) is infeasible* **then**

Prune node: infeasible

else if $f(x^{(l,u)}) > U$ **then**

Prune node; dominated by bound U

else if $x_I^{(l,u)}$ *integral* **then**

Update incumbent: $U = f(x^{(l,u)})$, $x^* = x^{(l,u)}$ & prune.

else GenerateCuts($x^{(l,u)}, j$) ... details later

until *no new cuts generated or node pruned*

if *(NLP(l, u)) not pruned & not incumbent* **then**

BranchOnVariable($x_j^{(l,u)}, l, u, \mathcal{H}$)

Cut Generation Overview

Algorithm 1: Solve separation problem to generate subgradient cut

Subroutine: GenerateCuts $(x^{(l,u)}, j)$

// *Generate a valid inequality that cuts off* $x_j^{(l,u)} \notin \{0,1\}$

Solve separation (NLP) problem in $x^{(l,u)}$ for valid cut.

Add valid inequality to $(\text{NLP}(l,u))$.

GenerateCuts: valid inequality to eliminate fractional solution

- Given fractional solution $x^{(l,u)}$ with $x_j^{(l,u)} \notin \{0,1\}$.
- Let $\mathcal{F}(l,u)$ mixed-integer feasible set of node $\text{NLP}(l,u)$.
- Find cut $\pi^T x \leq \pi_0$ such that
  - $\pi^T x \leq \pi_0$ for all $x \in \mathcal{F}(l,u)$
  - $\pi^T x^{(l,u)} > \pi_0$, i.e. $x^{(l,u)}$ violates the cut
- Solve a separation problem (e.g. an NLP) for cut $\pi^T x \leq \pi_0$

... lifting cuts makes them valid throughout the tree.

Branch-and-Cut Challenges

Computational Considerations of Branch-and-Cut

- Cut-generation problem may be hard to solve
- Adds burden of additional NLP solves to BnB
  - Can solve LP instead of NLP, e.g. from OA
- Must add cut-management to solver
- Lifting cuts may help to make them valid in whole tree
- NLPs still don't hot-start

[Stubbs and Mehrotra, 1999] generate cuts only at root node

Outline

Mixed-Integer Rounding (MIR) for OA-MILP

Goal: Strengthen MILP relaxations of LP/NLP-based BnB
... iteratively add cuts to remove fractional LP solutions

Start by considering MIR cuts for "easy set"

$$S := \{(x_1, x_2) \in \mathbb{R} \times \mathbb{Z} \mid x_2 \le b + x_1,\ x_1 \ge 0\},$$

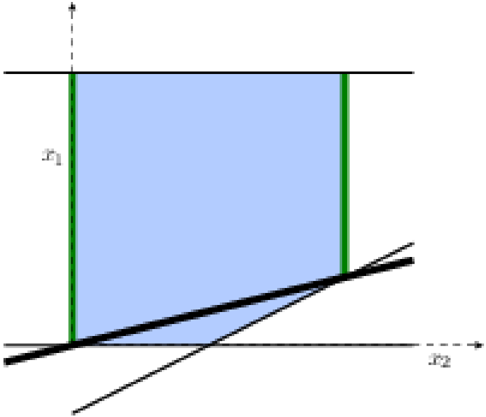
where $R = \{1\}$ and $I = \{2\}$.
Let $f_0 = b - \lfloor b \rfloor$, then cut

$$x_2 \le \lfloor b \rfloor + \frac{x_1}{1 - f_0}$$

is valid for S; look at two cases:

1. $x_2 \le \lfloor b \rfloor$
2. $x_2 \ge \lfloor b \rfloor + 1$.

Example of Simple MIR Cut



MIR cut: $x_2 \leq 2x_1$ derived from $x_2 \leq \frac{1}{2} + x_1$.

General MIR Cuts

For general MILP consider set

$$X := \{(x_R^+, x_R^-, x_I) \in \mathbb{R}_+^2 \times \mathbb{Z}_+^p \mid a_I^T x_I + x_R^+ \leq b + x_R^- \}.$$

... selected constraint row of MILP or one-row relaxation of subset

- Continuous variables aggregated in x_R^+ and x_R^- depending on sign of coefficient in a_R.
- Obtain following valid inequality:

$$\sum_{i \in I} \left(\lfloor a_i \rfloor + \frac{\max\{f_i - f_0, 0\}}{1 - f_0} \right) x_i \leq \lfloor b \rfloor + \frac{x_R^-}{1 - f_0},$$

$f_i = a_i - \lfloor a_i \rfloor$ for $i \in I$ and $f_0 = b - \lfloor b \rfloor$ fractional parts a and b.

Gomory Cuts and MIR Cuts

Gomory cuts originally from [Gomory, 1958, Gomory, 1960] for ILP
MILP Gomory cut given by

$$\sum_{i \in I_1} f_i x_i + \sum_{i \in I_2} \frac{f_0(1-f_i)}{f_i} x_i + x_R^+ + \frac{f_0}{1-f_0} x_R^- \geq f_0$$

where $I_1 = \{i \in I \mid f_i \leq f_0\}$ and $I_2 = I \setminus I_1$.
... is instance of MIR cut. Consider set

$$X = \{(x_R, x_0, x_I) \in \mathbb{R}^2_+ \times \mathbb{Z}_+ \times \mathbb{Z}^p \mid x_0 + a_I^T x_I + x_R^+ - x_R^- = b\},$$

generate a MIR inequality, and eliminate x_I^0.
In MINLP Gomory & MIR cuts generated from MILP relaxations
... [Akrotirianakis et al., 2001] report modest improvement

Outline

Perspective Formulations

MINLPs use binary indicator variables, x_b, to model nonpositivity of $x_c \in \mathbb{R}$

Model as variable upper bound

$$0 \leq x_c \leq u_c x_b, \quad x_b \in \{0,1\}$$

$\Rightarrow$ if $x_c > 0$, then $x_b = 1$

Perspective reformulation applies, if x_b also in convex $c(x) \leq 0$

- Significantly improve reformulation
- Pioneered by [Frangioni and Gentile, 2006];
  ... strengthen relaxation using perspective cuts

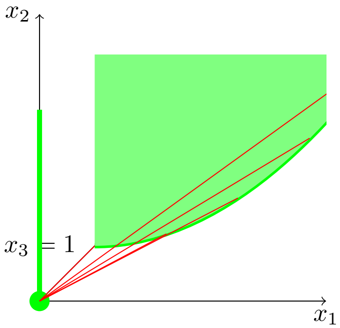
Example of Perspective Formulation

Consider MINLP set with three variables:

$$S = \left\{ (x_1, x_2, x_3) \in \mathbb{R}^2 \times \{0,1\} \ : \ x_2 \geq x_1^2, \quad u x_3 \geq x_1 \geq 0 \right\}.$$

Can show that $S = S^0 \cup S^1$, where

$$S^0 = \left\{ (0, x_2, 0) \in \mathbb{R}^3 \ : \ x_2 \geq 0 \right\},$$
$$S^1 = \left\{ (x_1, x_2, 1) \in \mathbb{R}^3 \ : \ x_2 \geq x_1^2, \ u \geq x_1 \geq 0 \right\}.$$



Example of Perspective Formulation

Geometry of convex hull of S:
Lines connecting origin ($x_3 = 0$) to parabola $x_2 = x_1^2$ at $x_3 = 1$

Define convex hull of S as conv(S)

$$:= \left\{(x_1, x_2, x_3) \in \mathbb{R}^3 : x_2 x_3 \geq x_1^2,\ u x_3 \geq x_1 \geq 0, 1 \geq x_3 \geq 0, x_2 \geq 0\right\}$$

where $x_2 x_3 \geq x_1^2$ is defined in terms of perspective function

$$\mathcal{P}_f(x, z) := \begin{cases} 0 & \text{if } z = 0, \\ z f(x/z) & \text{if } z > 0. \end{cases}$$

Epigraph of $\mathcal{P}_f(x, z)$: cone pointed at origin with lower shape $f(x)$

$x_b \in \{0, 1\}$ indicator forces $x_c = 0$, or $c(x_c) \leq 0$ if $x_b = 1$ write

$x_b c(x_c / x_b)$ ...is tighter convex formulation

Generalization of Perspective Cuts

[Günlük and Linderoth, 2012] consider more general problem

$$(P) \qquad \min_{(x,z,\eta)\in\mathbb{R}^n\times\{0,1\}\times\mathbb{R}} \left\{\eta \mid \eta \geq f(x) + cz, Ax \leq bz\right\}.$$

where

1. $X = \{x \mid Ax \leq b\}$ is bounded
2. $f(x)$ is convex and finite on X, and $f(0) = 0$

Theorem (Perspective Cut)

For any $\bar{x} \in X$ and subgradient $s \in \partial f(\bar{x})$, the inequality

$$\eta \geq f(\bar{x}) + c + s^T(x - \bar{x}) + (c + f(\bar{x}) - s^T\bar{x}))(z - 1)$$

is valid cut for (P)

Stronger Relaxations [Günlük and Linderoth, 2012]

- z_R: Value of NLP relaxation
- z_{GLW}: Value of NLP relaxation after GLW cuts
- z_P: Value of perspective relaxation
- z^*: Optimal solution value

Separable Quadratic Facility Location Problems

$\|M\|$	$\|N\|$	z_R	z_{GLW}	z_P	z^*
10	30	140.6	326.4	346.5	348.7
15	50	141.3	312.2	380.0	384.1
20	65	122.5	248.7	288.9	289.3
25	80	121.3	260.1	314.8	315.8
30	100	128.0	327.0	391.7	393.2

$\Rightarrow$ Tighter relaxation gives faster solves!

Nonlinear Perspective of the Perspective

Potential Pitfalls of Perspective of $h(x) \leq 0$:

- $y h(x/y) \leq 0$ ... division by zero?
- function, gradients & Hessian may not be defined at 0
- in practice get IEEE exception messages from AMPL

Example: Stochastic Service System Design

$$
\begin{array}{ll}
\underset{x,y,z}{\text{minimize}} & \frac{v}{100} + (y - \frac{1}{4})^2 + (z - \frac{1}{2})^2 \\
\text{subject to} & z - \frac{v}{1+v} \leq 0 \\
& 0 \leq z \leq y, \quad v \geq 0, \quad y \in \{0, 1\}
\end{array}
$$

Perspective of nonlinear constraint:

$$
y \left(z/y - \frac{v/y}{1 + v/y} \right) \leq 0 \quad \Leftrightarrow \quad z - \frac{v}{1 + v/y} \leq 0
$$

... not defined at $y = 0$ even after cancellation.

Nonlinear Perspective of the Perspective

Study re-formulations:

$$
\begin{aligned}
z - \tfrac{v}{1+v/y} &\leq 0 && \text{perspective} \\
zy + zv - vy &\leq 0 && \text{smooth} \\
\sqrt{4v^2 + (y+z)^2} - 2v + y - z &\leq 0 && \text{2nd-order cone}
\end{aligned}
$$

- 2nd-order cone requires SOC solver $\Rightarrow$ no general NLPs!
- IPOPT, SNOPT et al. fail for smooth formulation:
  - "Smooth formulation is nonconvex $\Rightarrow$ NLP solvers fail"
  - BONMIN fails to solve MINLPs using smooth formulation
- BB solvers fail on perspective formulation:
  ... IEEE exception $\forall$ nodes with $y = 0$

Nonlinear Perspective on the Perspective

Nonconvex formulation: $c_1(v, y, z) = zy + zv - vy \leq 0$

- Feasible set is convex $\Rightarrow$ unique minimizer
- NLP solvers converge to unique minimum ... just very slowly!
- Look at gradient:

$$\nabla c_1 = \begin{pmatrix} z - y \\ z - v \\ y + v \end{pmatrix}$$

$\Rightarrow \nabla c_1(0) = 0^T$

$\Rightarrow c_1$ violates MFCQ at 0

- Slow convergence & failure is due to failure of MFCQ

... more next!

Gradients & Constraint Qualifications (CQ)

Let $\mathcal{F} := \{c(x) \geq 0\}$ feasible set

CQs ensure that linearizations describe $\mathcal{F}$ locally!

- LPs always satisfy a CQ
- Ensure validity of first-order (gradient/KKT) conditions
- Solvers that rely on linearization techniques work well

Mangasarian-Fromowitz Constraint Qualification (MFCQ)

1. The gradients of equality constraints linearly independent
2. For all active $\mathcal{A}$ inequality constraints $\mathcal{A}(x) := \{i : c_i(x) = 0\}$:
   $\exists s : \nabla c_i^T s < 0, \; \forall i \in \mathcal{A}$ ... strictly feasible direction

MFCQ violated by $\nabla c_1 = 0$, because $0^T s < 0$ can never hold!
... causes slow convergence of *any NLP solver*

Numerical Experience with the Bad the Perspective

Bad perspective of uncapacitated facility location problem:

$$\begin{aligned} & \underset{x,y,z}{\text{minimize}} \quad z + y \\ & \text{subject to } x^2 - zy \le 0 \quad 0 \le x \le z, \; z \in \{0,1\} \end{aligned}$$

```
Major Minor TrustRad RegParam StepNorm Constrnts  Objective Optimal Phase Step
------------------------------------------------------------------------------
    0     0       10       10        0       0.5       1.01       0     2
    1     1       10       10    0.625         0      0.385       0     2  SQP
    2     1       10       10    0.188         0     0.1875       0     2  SQP
[ ... ]
   28     1       10       10 2.79e-09         0  2.794e-09       0     2  SQP
   29     1       10       10   1.4e-09         0  1.397e-09       0     2  SQP
   30     1       10       10 6.98e-10         0  6.985e-10       2     2  SQP

ASTROS Version 2.0.2 (20100913): Solution Summary
===============================================
  Major iters     =         30 ; Minor iters     =         30 ;
  KKT-residual    =     0.4286 ; Complementarity =  1.996e-10 ;
  Final step-norm =  6.985e-10 ; Final TR-radius =         10 ;
---------------------------------------------------------------
ASTROS  Version 2.0.2 (20100913): Step got too small
```

Linear rate of convergence ... similar for MINOS, FilterSQP, ...

Remedy: Limiting Gradients for the Perspective

Goal: Compute limiting gradients for perspective as $y \to 0$

Perspective of SSSD example

$$\begin{aligned} & z - \frac{v}{1+v/y} \le 0 \\ & 0 \le z \le y \\ & v \ge 0,\ y \in \{0,1\}. \end{aligned}$$

$$\nabla c_p = \begin{pmatrix} \dfrac{-1}{(1+v/y)^2} \\ \\ \dfrac{-v^2/y^2}{(1+v/y)^2} \\ \\ 1 \end{pmatrix}$$

Objective implies
$v = z/(1-z)$ active.

Observation: $y \to 0$ implies $z \to 0$, and $v = z/(1-z) \to 0$.

$$\nabla c_p(0) \in \text{conv} \left\{ \begin{pmatrix} -1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -\frac{1}{4} \\ -\frac{1}{4} \\ 1 \end{pmatrix} \right\}$$

... similar derivation possible for gradients of SOC formulation!

Nonlinear Perspective of the Perspective

NLP solvers for perspective constraints

- Perspective violates linear independence CQ (LICQ)
  ... OK for robust NLP solvers (work with basis)
- Limiting gradients exist & satisfy MFCQ at 0
- Hessian blows up near $y = 0$: $\nabla^2 c_p = \mathcal{O}(y^{-1})$ typically
  ... OK because null-space is empty near $y = 0$ (LICQ fails)

Modify NLP solvers & make them aware of structure

1. Use limiting gradients near 0
2. Set Hessian $\nabla^2 c_p = [0]$ near 0

$\Rightarrow$ robust & fast local convergence (proof similar to MPECs?)

Exact Smoothing of the Perspective

Changing NLP solvers is hard ... modify the perspective:

$$\underset{x,y,z}{\text{minimize}} \quad z + y$$
$$\text{subject to } \frac{x^2}{z} - y \leq 0, \quad 0 \leq x \leq z, \; z \in \{0,1\}$$

For $\tau > 0$ (e.g. $\tau = 0.1$), replace perspective by:

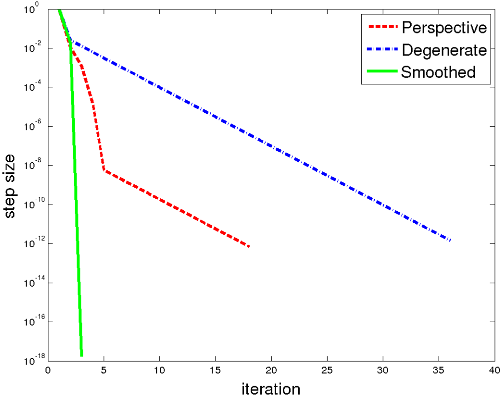
$$c_s(x,y,z) = \begin{cases} \frac{x^2}{z} - y & \text{if } z \geq \tau \\ 2x + x - y - z & \text{otherwise,} \end{cases}$$

continuously differentiable (across line $x = z = \tau$).

... readily implemented in AMPL & converges rapidly!

Nonlinear Perspective of the Perspective

Another example



... work in progress

Outline

Disjunctive Branch-and-Cut

[Stubbs and Mehrotra, 1999] for convex, binary MINLP:

$$\underset{\eta, x}{\text{minimize}}\ \eta \quad \text{s.t. } \eta \geq f(x),\ c(x) \leq 0,\ x \in X,\ x_i \in \{0,1\}\ \forall\, i \in I$$

Node in BnB tree with solution x', and $0 < x'_j < 1$ for $j \in I$
Relaxation: $\mathcal{C} = \{x \in X \mid f(x) \leq \eta,\ c(x) \leq 0,\ 0 \leq x_I \leq 1\}$
Let $I_0, I_1 \subseteq I$ index sets of 0-1 vars fixed to zero or one

Goal: Generate a valid inequality tat cuts off x'
Consider two disjoint sets ("feasible sets after branching on x_j")

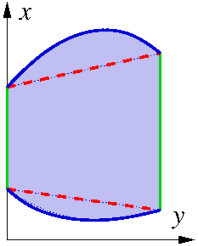
$$\mathcal{C}_j^0 = \{x \in \mathcal{C} \mid x_j = 0,\ 0 \leq x_i \leq 1\ \forall i \in I, i \neq j\},$$
$$\mathcal{C}_j^1 = \{x \in \mathcal{C} \mid x_j = 1,\ 0 \leq x_i \leq 1\ \forall i \in I, i \neq j\}.$$

... and find description of convex hull: $\tilde{M}_j(\mathcal{C}) = \text{conv}(\mathcal{C}_j^0 \cup \mathcal{C}_j^1)$

Disjunctive Cuts for MINLP

Extension of disjunctive cuts from MILP, [Balas, 1979]
Continuous relaxation

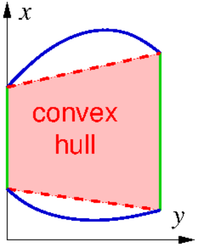
- $\mathcal{C} := \{x | c(x) \leq 0,\ 0 \leq x_I \leq 1,\ 0 \leq x_C \leq U\}$



Disjunctive Cuts for MINLP

Extension of disjunctive cuts from MILP, [Balas, 1979]
Continuous relaxation

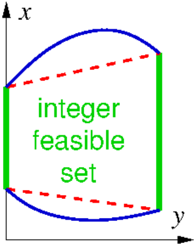
- $\mathcal{C} := \{x | c(x) \le 0,\ 0 \le x_I \le 1,\ 0 \le x_C \le U\}$
- $\mathcal{C} := \text{conv}(\{x \in \mathcal{C} \mid x_I \in \{0,1\}^p\})$



Disjunctive Cuts for MINLP

Extension of disjunctive cuts from MILP, [Balas, 1979]
Continuous relaxation

- $\mathcal{C} := \{x | c(x) \leq 0,\ 0 \leq x_I \leq 1,\ 0 \leq x_C \leq U\}$
- $\mathcal{C} := \text{conv}(\{x \in \mathcal{C} \mid x_I \in \{0,1\}^p\})$
- $\mathcal{C}_j^{0/1} := \{x \in \mathcal{C} | x_j = 0/1\}$



$$\text{let } \mathcal{M}_j(C) := \left\{ \begin{array}{l} z = \lambda_0 u_0 + \lambda_1 u_1 \\ \lambda_0 + \lambda_1 = 1,\ \lambda_0, \lambda_1 \geq 0 \\ u_0 \in \mathcal{C}_j^0,\ u_1 \in \mathcal{C}_j^1 \end{array} \right\}$$

$\Rightarrow \mathcal{P}_j(\mathcal{C}) :=$ projection of $\mathcal{M}_j(\mathcal{C})$ onto z

$\Rightarrow \mathcal{P}_j(\mathcal{C}) = \text{conv}\left(\mathcal{C} \cap x_j \in \{0,1\}\right)$ and $\mathcal{P}_{1\ldots p}(\mathcal{C}) = \mathcal{C}$

Disjunctive Cuts

Snag: Description of convex hull is nonconvex:

$$\text{let } \mathcal{M}_j(\mathcal{C}) := \left\{ \begin{array}{l} z = \lambda_0 u_0 + \lambda_1 u_1 \\ \lambda_0 + \lambda_1 = 1,\ \lambda_0, \lambda_1 \geq 0 \\ u_0 \in \mathcal{C}_j^0,\ u_1 \in \mathcal{C}_j^1 \end{array} \right\}$$

$\Rightarrow$ need global optimization solvers for separation problem

$\Rightarrow$ prohibitive; instead use convex formulation: $\tilde{M}_j(\mathcal{C})$

Disjunctive Cuts

Can describe $\tilde{M}_j(\mathcal{C})$ with perspective $\mathcal{P}_{c_i}$

$$\tilde{M}_j(\mathcal{C}) = \left\{ (x_F, v_0, v_1, \lambda_0, \lambda_1) \left| \begin{array}{ll} v_0 + v_1 = x_F, & v_{0j} = 0, \;\; v_{1j} = \lambda_1 \\ \lambda_0 + \lambda_1 = 1, & \lambda_0, \lambda_1 \geq 0 \\ \lambda_0 c_i(v_0/\lambda_0) \leq 0, 1 \leq \;\; i \leq m & \\ \lambda_1 c_i(v_1/\lambda_1) \leq 0, 1 \leq \;\; i \leq m & \end{array} \right. \right\},$$

Obtain a convex separation NLP ...

Disjunctive Cuts: Separation NLP

Goal: Find $\hat{x}$ closest to fractional solution x' in convex hull

$$\text{BC-SEP}(x', j) \begin{cases} \underset{x, v_0, v_1, \lambda_0, \lambda_1}{\text{minimize}} & ||x - x'||, \\ \text{subject to} & (x, v_0, v_1, \lambda_0, \lambda_1) \in \tilde{M}_j(\mathcal{C}) \\ & x_i = 0, \ \forall i \in I_0 \\ & x_i = 1, \ \forall i \in I_1. \end{cases}$$

optimal solution $\hat{x}$ with multipliers π_F for equality $v_0 + v_1 = x_F$

Theorem

Optimal dual solution of (BC-SEP(x', j)), then following cut is valid and eliminates x':

$$\pi_F^T x_F \leq \pi_F^T \hat{x}_F$$

Disjunctive Cuts: Example

Consider following MINLP example

$$\begin{cases} \underset{x_1,x_2}{\text{minimize}} & x_1 \\ \text{subject to} & (x_1 - \frac{1}{2})^2 + (x_2 - \frac{3}{4})^2 \le 1 \\ & -2 \le x_1 \le 2 \\ & x_2 \in \{0,1\} \end{cases}$$

$\Rightarrow$ solution of NLP relaxation: $x' = (x'_1, x'_2) = (-\frac{1}{2}, \frac{3}{4})$

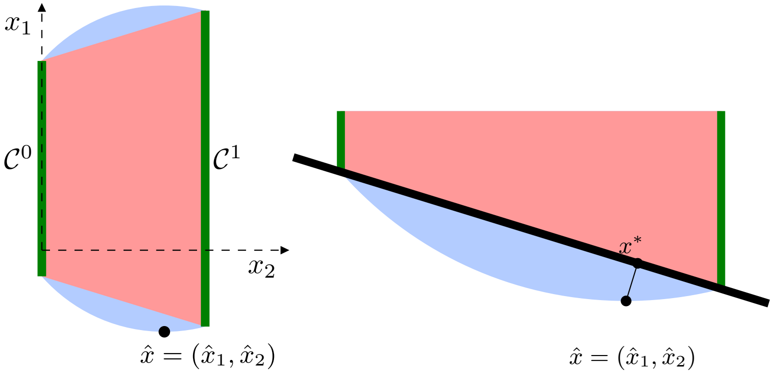
Solve $(x_1 - \frac{1}{2})^2 + (x_2 - \frac{3}{4})^2 \le 1$ for x_1, given $x_2 = 0$ and $x_2 = 1$:

$$\mathcal{C}^0 = \left\{ (x_1, 0) \in \mathbb{R} \times \{0,1\} \;\middle|\; 2 - \sqrt{7} \le 4x_1 \le 2 + \sqrt{7} \right\},$$

$$\mathcal{C}^1 = \left\{ (x_1, 1) \in \mathbb{R} \times \{0,1\} \;\middle|\; 2 - \sqrt{15} \le 4x_1 \le 2 + \sqrt{15} \right\}.$$

Solving (BC-SEP$(x', 2)$), we find the cut $x_1 + 0.3x_2 \ge -0.166$

Disjunctive Cuts: Example



Convex hull, relaxation, and disjunctive cut

Lifting Disjunctive Cuts

Cuts are only valid for sub-tree rooted at relaxation
To obtain globally valid cut

$$\pi^T x \leq \pi^T \hat{x}$$

assign

$$\pi_i = \min\{e_i^T H_0^T \mu_0, e_i^T H_1^T \mu_1\}, \; i \notin F$$

where e_i is i^{th} unit vector, F set of "free" variables and

- $\mu_0 = (\mu_{0F}, 0)$ and μ_{0F} multiplier of perspective $\mathcal{P}_c(v_0, \lambda_0) \leq 0$
- $\mu_1 = (\mu_{1F}, 0)$ and μ_{1F} multiplier of perspective $\mathcal{P}_c(v_1, \lambda_1) \leq 0$
- H_0, H_1 matrices of subgradient rows $\partial_v \mathcal{P}_{c_i}(v_j, \lambda_j)^T$, for $j = 0, 1$

Preferred norm for cut generation, (BC-SEP(x', j)), is ℓ_∞-norm

Outline

Implementation of Disjunctive Cuts

NLP (BC-SEP(x', j)) is not easy to solve:

- NLP has twice number of variables as original problem
- Perspective functions not differentiable at origin
- Hessian of perspective blows up near origin

$\Rightarrow$ NLP slow (and solvers may fail)

Suggest LP-based separation [Kılınç et al., 2010]

- Consider outer approximation relaxations of MINLP
- Iteratively tighten the outer approximation

$\Rightarrow$ faster and more robust cut generation

Implementation of Disjunctive Cuts

Let $\mathcal{B} \supset \mathcal{C} = \{x \in X \mid f(x) \leq \eta,\ c(x) \leq 0,\ 0 \leq x_I \leq 1\}$

Instead of $\mathcal{C}_j^0$ and $\mathcal{C}_j^1$ we consider

$$\mathcal{B}_j^0 = \{x \in \mathcal{B}^0 \mid x_j = 0\}, \quad \mathcal{B}_j^1 = \{x \in \mathcal{B}^0 \mid x_j = 1\}$$

valid inequalities for $\mathrm{conv}(\mathcal{B}_j^0 \cup \mathcal{B}_j^1)$ are also valid for $\mathrm{conv}(\mathcal{C}_j^0 \cup \mathcal{C}_j^1)$

Create linear (OA) sets $\mathcal{B}_j^0, \mathcal{B}_j^1$ iteratively (t):

$$\begin{aligned}\mathcal{B}_j^0(t) = \{x \in \mathbb{R}^n \mid x_j = 0,\ & f' + \nabla f'^T (x - x') \leq \eta, \\ & c' + \nabla c'^T (x - x') \leq 0,\ \forall x' \in \mathcal{K}_j^0(t)\},\end{aligned}$$

where $\mathcal{K}_j^0(t)$ set of linearization points; $\mathcal{B}_j^1(t)$ defined similarly

- $\mathcal{K}_j^0(t)$ augmented by solution of linear separation, x_t'
- Use "friendly points", $x_t' = \lambda x_{t0}' + (1 - \lambda) x_{t1}'$ for $\lambda \in [0, 1]$

$\Rightarrow$ converges to solution of (BC-SEP(x', j)); but slowly (?)

Outline

Summary and Exercises

Key points

- Single-tree methods are state-of-the-art
- Presolve for MINLP important ... need computational graph
- Branch-and-cut approaches being developed for MINLP

Solution to exercises ...

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.

Akrotirianakis, I., Maros, I., and Rustem, B. (2001).
An outer approximation based branch-and-cut algorithm for convex 0-1 MINLP problems.
Optimization Methods and Software, 16:21–47.

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

Balas, E. (1979).
Disjunctive programming.
In *Annals of Discrete Mathematics 5: Discrete Optimization*, pages 3–51. North Holland.

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.

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

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

Drewes, S. and Ulbrich, S. (2012).
Subgradient based outer approximation for mixed integer second order cone programming.
In *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 41–59. Springer, New York.
ISBN 978-1-4614-1926-6.

Frangioni, A. and Gentile, C. (2006).
Perspective cuts for a class of convex 0-1 mixed integer programs.
Mathematical Programming, 106:225–236.

Gomory, R. E. (1958).
Outline of an algorithm for integer solutions to linear programs.
Bulletin of the American Mathematical Monthly, 64:275–278.

Gomory, R. E. (1960).
An algorithm for the mixed integer problem.
Technical Report RM-2597, The RAND Corporation.

Griewank, A. and Toint, P. L. (1984).
On the exsistence of convex decompositions of partially separable functions.
Mathematical Programming, 28:25–49.

Günlük, O. and Linderoth, J. T. (2012).
Perspective reformulation and applications.

In *IMA Volumes*, volume 154, pages 61–92.

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.

Kılınç, M., Linderoth, J., and Luedtke, J. (2010).
Effective separation of disjunctive cuts for convex mixed integer nonlinear programs.
Technical Report 1681, Computer Sciences Department, University of Wisconsin-Madison.

Savelsbergh, M. W. P. (1994).
Preprocessing and probing techniques for mixed integer programming problems.
ORSA Journal on Computing, 6:445–454.

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

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

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