Convex Mixed-Integer Nonlinear Optimization I

Summer School on Optimization of Dynamical Systems

Sven Leyffer and Jeff Linderoth

Argonne National Laboratory

September 3-7, 2018

Outline

1. Problem Definition and Assumptions
2. Nonlinear Branch-and-Bound
3. Multi-Tree Methods
4. Single-Tree Methods

Mixed-Integer Nonlinear Optimization

Mixed-Integer Nonlinear Program (MINLP)

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

Basic Assumptions for Convex MINLP

- A1 $\mathcal{X}$ is a bounded polyhedral set.
- A2 f and c twice continuously differentiable convex
- A3 MINLP satisfies a constraint qualification.

A2 (convexity) most restrictive (show how to relax later)
A3 is technical (MFCQ would have been sufficient)

Overview of Basic Methods

Two broad classes of method

1. Single-tree methods; e.g.
    - Nonlinear branch-and-bound
    - LP/NLP-based branch-and-bound
    - Nonlinear branch-and-cut

   ... build and search a single tree
2. Multi-tree methods; e.g.
    - Outer approximation
    - Benders decomposition
    - Extended cutting plane method

   ... alternate between NLP and MILP solves

Multi-tree methods only evaluate functions at integer points

Concentrate on methods for convex problems today.

Can mix different methods & techniques.

Overview of Components of Methods

All MINLP solvers built on following components ...

Relaxation

- Used to compute a lower bound on the optimum
- Obtained by enlarging feasible set; e.g. ignore constraints
- Typically much easier to solve than MINLP

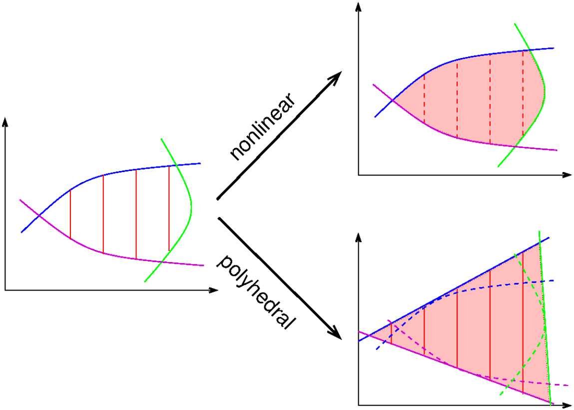
Constraint Enforcement

- Exclude solutions from relaxations not feasible in MINLP
- Refine or tighten of relaxation; e.g. add valid inequalities

Upper Bounds

- Obtained from any feasible point; e.g. solve NLP for fixed $x_{\mathcal{I}}$

Outline of Relaxations



Nonlinear and polyhedral relaxation

Theorem (Relaxation Property)

If solution of relaxation is feasible, then it is optimal.

Relaxations of Integrality

Definition (Relaxation)

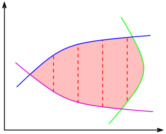
Optimization problem $\min\{\breve{f}(x) : x \in \mathcal{R}\}$ is a relaxation of $\min\{f(x) : x \in \mathcal{F}\}$, iff $\mathcal{R} \supset \mathcal{F}$ and $\breve{f}(x) \leq f(x)$ for all $x \in \mathcal{F}$.

Goal: relaxation easy to solve globally, e.g. MILP or NLP

Relaxing Integrality

- Relax Integrality $x_i \in \mathbb{Z}$ to $x_i \in \mathbb{R}$ for all $i \in \mathcal{I}$
- Gives *nonlinear relaxation* of MINLP, or NLP:

$$\begin{cases} \underset{x}{\text{minimize}} & f(x), \\ \text{subject to} & c(x) \leq 0, \\ & x \in \mathcal{X}, \quad \text{continuous} \end{cases}$$



- Used in branch-and-bound algorithms

Relaxations of Nonlinear Convex Constraints

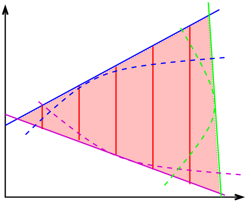
Relaxing Convex Constraints

- Convex $0 \geq c(x)$ and $\eta \geq f(x)f$ relaxed by supporting hyperplanes

$$\eta \geq f^{(k)} + \nabla f^{(k)^T}(x - x^{(k)})$$
$$0 \geq c^{(k)} + \nabla c^{(k)^T}(x - x^{(k)})$$

  for a set of points $x^{(k)},\ k = 1, \ldots, K$.

- Obtain polyhedral relaxation of convex constraints.
- Used in the outer approximation methods.



Constraint Enforcement

Goal: Given solution of relaxation, $\hat{x}$, not feasible in MINLP, exclude it from further consideration to ensure convergence

Three constraint enforcement strategies

1. Relaxation refinement: tighten the relaxation
2. Branching: disjunction to exclude set of non-integer points
3. Spatial branching: divide region into sub-regions

Strategies can be combined ...

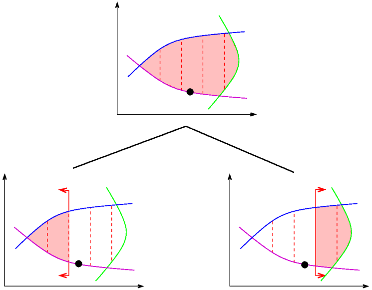
Constraint Enforcement: Branching

Eliminate current $\hat{x}$ solution by branch on integer variables:

1. Select fractional $\hat{x}_i$ for some $i \in \mathcal{I}$
2. Create two new relaxations by adding

$$x_i \leq \lfloor \hat{x}_i \rfloor \quad \text{and} \quad x_i \geq \lceil \hat{x}_i \rceil \qquad \text{respectively}$$

... solution to MINLP lies in one of the new relaxations.



... creates branch-and-bound tree

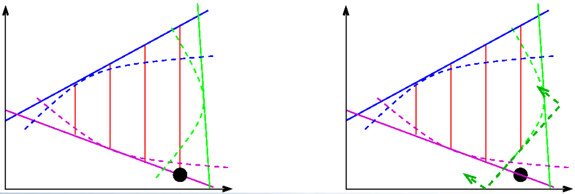
Constraint Enforcement: Refinement

Tighten the relaxation to remove current solution $\hat{x}$ of relaxation

- Add a valid inequality to relaxation, i.e. an inequality that is satisfied by all feasible solutions of MINLP
- Valid inequality is called a cut if it excludes $\hat{x}$
- Example: $c(x) \leq 0$ convex, and $\exists i : c_i(\hat{x}) > 0$, then

$$0 \geq \hat{c}_i + \nabla \hat{c}^T (x - \hat{x})$$

  cuts off $\hat{x}$. Proof: Exercise.
- Used in Benders decomposition and outer approximation.
- MILP: cuts are basis for branch-and-cut techniques.



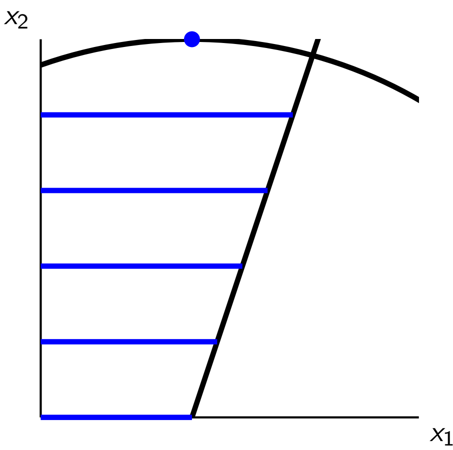
Outline

1. Problem Definition and Assumptions
2. Nonlinear Branch-and-Bound
3. Multi-Tree Methods
4. Single-Tree Methods

Natural Relaxation (Convex MINLP)

- For a convex MINLP

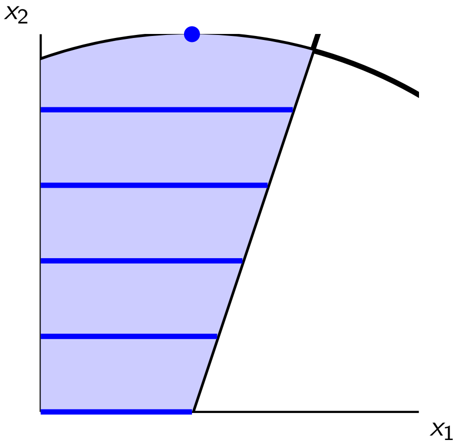
$$
\begin{aligned}
(x_1 - 2)^2 + (x_2 + 1)^2 &\leq 36 \\
3x_1 - x_2 &\leq 6 \\
0 \leq x_1, x_2 &\leq 5 \\
x_2 &\in \mathbb{Z}
\end{aligned}
$$



Natural Relaxation (Convex MINLP)

- For a convex MINLP

$$\begin{aligned}(x_1 - 2)^2 + (x_2 + 1)^2 &\leq 36 \\ 3x_1 - x_2 &\leq 6 \\ 0 \leq x_1, x_2 &\leq 5 \\ x_2 &\in \mathbb{R}\end{aligned}$$

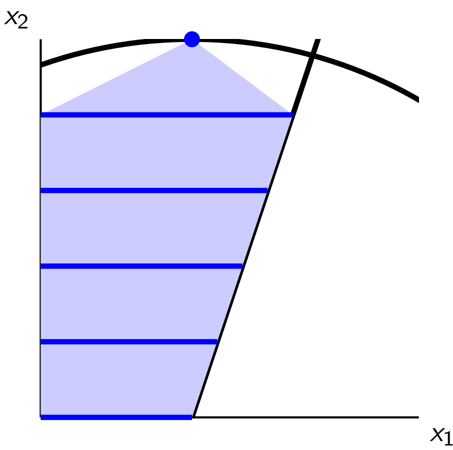


- Dropping integrality results in a convex, nonlinear relaxation

Natural Relaxation (Convex MINLP)

- For a convex MINLP

$$
\begin{aligned}
(x_1 - 2)^2 + (x_2 + 1)^2 &\leq 36 \\
3x_1 - x_2 &\leq 6 \\
0 \leq x_1, x_2 &\leq 5 \\
x_2 &\in \mathbb{R}
\end{aligned}
$$



- Dropping integrality results in a convex, nonlinear relaxation
- Ideal relaxation is convex hull of feasible points
- Optimizing linear function over this convex set solves the problem!

Nonlinear Convex Continuous Relaxation

Relaxing integrality gives convex NLP

Nonlinear Relaxation

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) \le 0 \\ & x \in \mathcal{X} \\ & x_i \in \mathbb{R} \text{ for all } i \in \mathcal{I} \end{aligned} \qquad (\text{NLP}_{relax})$$

- Convex optimization problem $\Rightarrow$ unique minimum
- NLP solvers guaranteed to find global minimum

Branching

- Solution x' of (NLP_{relax}) feasible but not integral:
  - Find a nonintegral variable, say $x'_i, i \in I$.
  - Introduce two child nodes with bounds $(l^-, u^-) = (l^+, u^+) = (l, u)$ and setting:

$$u_i^- := \lfloor x'_i \rfloor, \quad \text{and} \quad l_i^+ := \lceil x'_i \rceil$$

$\Rightarrow$ two problems $\text{NLP}_{(l^-,u^-)}$, $\text{NLP}_{(l^+,u^+)}$ (down/up branch)

Node NLPs

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c(x) \le 0 \\ & x \in \mathcal{X} \\ & l_i \le x_i \le u_i \end{aligned} \qquad (\text{NLP}_{(l,u)})$$

Branching

- Solution x' of (NLP_{relax}) feasible but not integral:
  - Find a nonintegral variable, say x'_i, $i \in I$.
  - Introduce two child nodes with bounds $(l^-, u^-) = (l^+, u^+) = (l, u)$ and setting:

$$u_i^- := \lfloor x'_i \rfloor, \quad \text{and} \quad l_i^+ := \lceil x'_i \rceil$$

$\Rightarrow$ two problems $\text{NLP}_{(l^-,u^-)}$, $\text{NLP}_{(l^+,u^+)}$ (down/up branch)

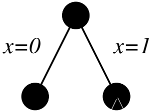
Pruning Rules

1. ($\text{NLP}_{(l,u)}$) infeasible $\Rightarrow$ NLPs in subtree also infeasible
2. Integer feasible solution $x^{(l,u)}$ of ($\text{NLP}_{(l,u)}$):
   - If $f(x^{(l,u)}) < U$, then new $x^* = x^{(l,u)}$ and $U = f^{(l,u)}$.
   - prune node no better solution in subtree
3. Optimal value of ($\text{NLP}_{(l,u)}$), $f(x^{(l,u)}) \geq U$
   $\Rightarrow$ prune node: no better integer solution in subtree

Nonlinear Branch and Bound

Solve relaxed NLP ($0 \leq x \leq 1$) . . . solution gives lower bound

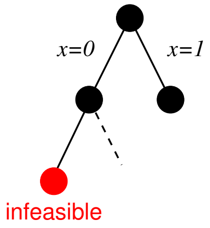
1. Solve NLPs & branch on x_i until



Nonlinear Branch and Bound

Solve relaxed NLP ($0 \leq x \leq 1$) ... solution gives lower bound

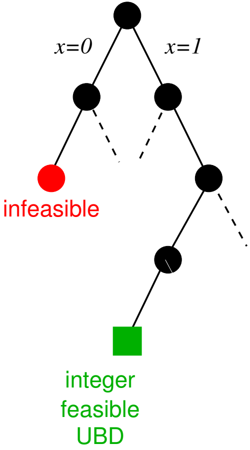
1. Solve NLPs & branch on x_i until
2. Node infeasible: ●



Nonlinear Branch and Bound

Solve relaxed NLP ($0 \leq x \leq 1$) . . . solution gives lower bound

1. Solve NLPs & branch on x_i until
2. Node infeasible: ●
3. Node integer feasible: □
   $\Rightarrow$ get upper bound (U)

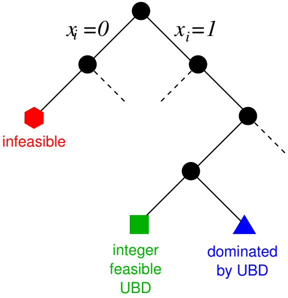


Nonlinear Branch and Bound

Solve relaxed NLP ($0 \leq x \leq 1$) ...solution gives lower bound

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

Search until no unexplored nodes



It Works Theorem

Assume that:

- X bounded polyhedral set;
- NLP solver returns global min.

$\Rightarrow$ BnB terminates at optimal solution

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 $x_{\mathcal{I}}^{(l,u)}$ *integral* **then**

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

  else

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

  end

end

Advanced Nonlinear BnB

Basic BnB will work, but needs improvements:

- Selection of branching variables ... ideally minimize tree to search
  ... estimate importance of variables from change in objective bounds
- Node selection strategies
  ... depth-first to find incumbent quickly
- Inexact NLP solves & hot-starts
  ... possible to search tree using QP (or LP solves)
- Cutting planes & branch-and-cut .... more later
- Software design & modern solvers, e.g. MINOTAUR
- Presolve & reformulations ⇒ better models

Presolve for Mixed-Integer Linear Optimization [Savelsbergh, 1994]

Outline

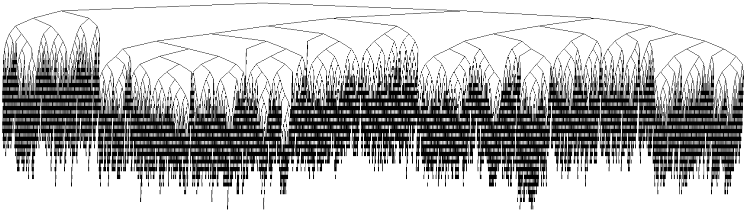
1 Problem Definition and Assumptions

2 Nonlinear Branch-and-Bound

3 Multi-Tree Methods

4 Single-Tree Methods

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!

Outer Approximation

Mixed-Integer Nonlinear Program (MINLP)

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

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

$$\text{NLP}(x_{\mathcal{I}}^{(j)}) \begin{cases} \underset{x}{\text{minimize}} & f(x) \\ \text{subject to } c(x) \leq 0 \\ \qquad\qquad x \in \mathcal{X} & \text{and } x_{\mathcal{I}} = x_{\mathcal{I}}^{(j)}, \end{cases}$$

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

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

Outer Approximation

Mixed-Integer Nonlinear Program (MINLP)

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

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

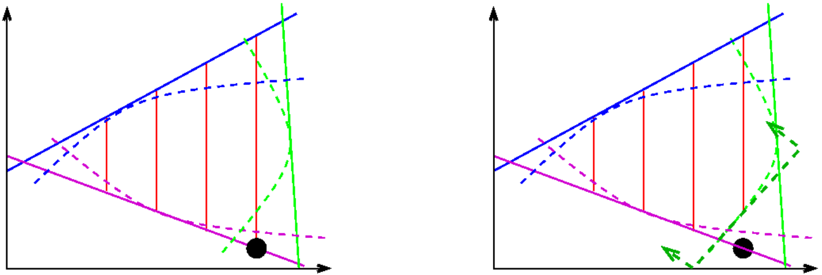
$$\mathrm{F}(x_{\mathcal{I}}^{(j)}) \begin{cases} \underset{x}{\text{minimize}} & \displaystyle\sum_{i \in J^{\perp}} w_i c_i^{+}(x) \\ \text{subject to } c_i(x) \leq 0,\ i \in J \\ \qquad x \in \mathcal{X} \quad \text{and } x_{\mathcal{I}} = x_{\mathcal{I}}^{(j)}, \end{cases}$$

where $w_i > 0$ are weights and solution is $x^{(j)}$.

$(\mathrm{F}(x_{\mathcal{I}}^{(j)}))$ generalize minimum norm solution

... provides certificate that $(\mathrm{NLP}(x_{\mathcal{I}}^{(j)}))$ infeasible

Outer Approximation



Separate infeasible points by

- Solving NLP for fixed integers to generate cut
- Collect cuts in MILP master problem

Outer Approximation

Convexity of f and c implies that

Lemma (Supporting Hyperplane)

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

$$\eta \geq f^{(j)} + \nabla f^{(j)^T}(x - x^{(j)})$$

and

$$0 \geq c^{(j)} + \nabla c^{(j)^T}(x - x^{(j)}),$$

are outer approximations (OA) of the feasible set of η-MINLP.

Lemma (Feasibility Cuts – Exercise Tomorrow)

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)

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

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

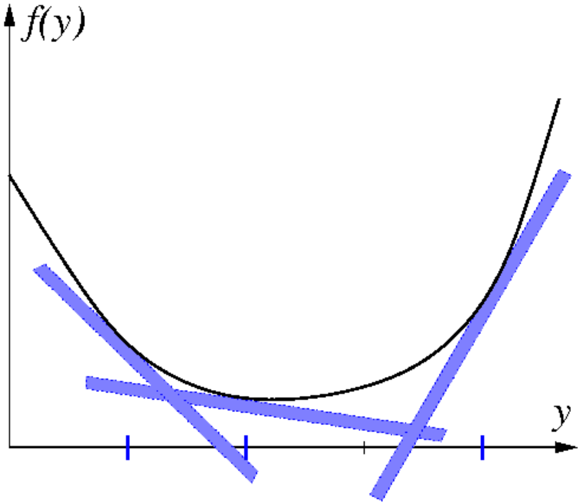
$$\mathcal{F} := \left\{ x^{(j)} \in \mathcal{X} : x^{(j)} \text{ solves } (\text{NLP}(x_{\mathcal{I}}^{(j)})) \quad \text{or } (\text{F}(x_{\mathcal{I}}^{(j)})) \right\}.$$

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

Construct equivalent OA-MILP (outer approximation MILP)

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

Outer Approximation in Less Than 1000 Words



… collecting all hyperplanes impractical!

Outer Approximation Algorithm

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

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

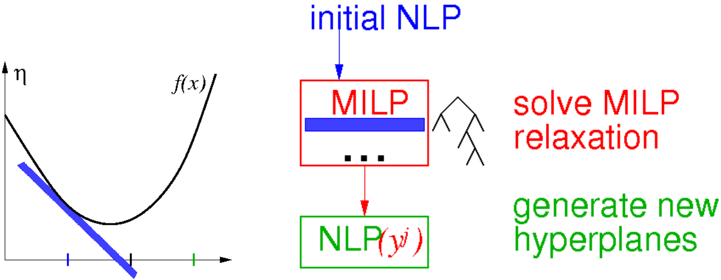
Define relaxation of OA-MILP, using $\mathcal{F}^k \subset \mathcal{F}$, with $\mathcal{F}^0 = \{0\}$

$$M(\mathcal{F}^k) \begin{cases} \underset{\eta,x}{\text{minimize}} & \eta, \\ \text{subject to} & \eta \le U^k - \epsilon \\ & \eta \ge f^{(j)} + \nabla f^{(j)^T}(x - x^{(j)}), \ \forall x^{(j)} \in \mathcal{F}^k \\ & 0 \ge c^{(j)} + \nabla c^{(j)^T}(x - x^{(j)}), \ \forall x^{(j)} \in \mathcal{F}^k \\ & x \in \mathcal{X}, \\ & x_i \in \mathbb{Z}, \ \forall i \in \mathcal{I}. \end{cases}$$

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

Outer Approximation Algorithm

Alternate between solve NLP(y_j) and MILP relaxation

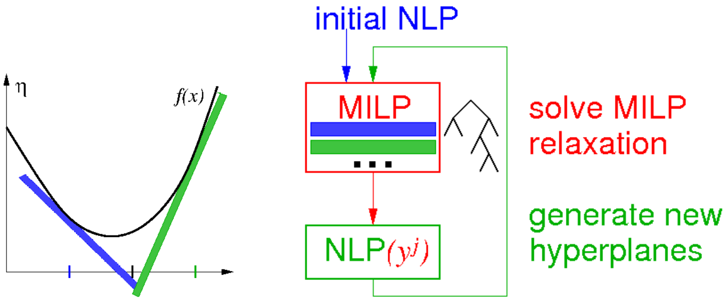


MILP $\Rightarrow$ lower bound; NLP $\Rightarrow$ upper bound

... convergence follows from convexity & finiteness

Outer Approximation Algorithm

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 tol $\epsilon > 0$, set $U^{-1} = \infty$, set $k = 0$, and $\mathcal{F}^{-1} = \emptyset$. ;

repeat

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

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

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

  else

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

  end

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

  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 (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

Benders Decomposition (Exercise ... add ECP???)

Can derive Benders cut from outer approximation:

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

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

- Using KKT conditions wrt continuous variables x_C:
  $0 = \nabla_C \mathcal{L}^{(j)} = \nabla_C f + \nabla_C c \lambda^{(j)}$ & $\lambda^{(j)^T} c^{(j)} = 0$
  ... eliminates continuous variables, x_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 (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
     $\Rightarrow$ $\nexists$ feasible descend 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 (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 ...

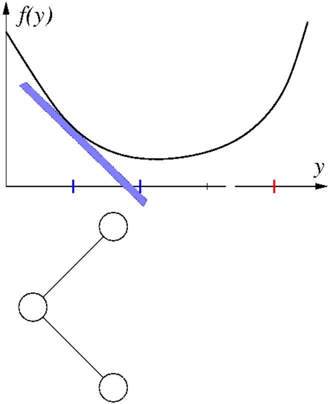
Outline

1. Problem Definition and Assumptions
2. Nonlinear Branch-and-Bound
3. Multi-Tree Methods
4. Single-Tree Methods

LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

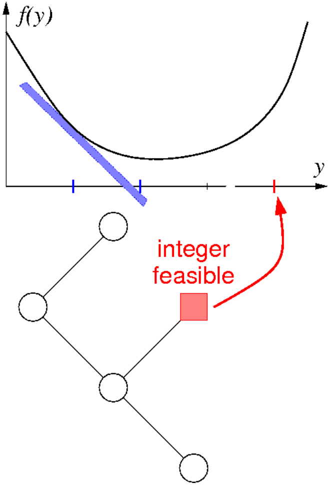
- Start solving master MILP ... using MILP branch-and-cut



LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

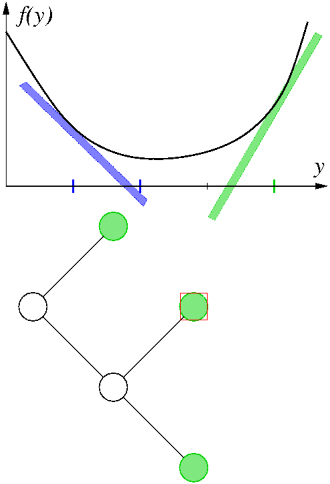
- Start solving master MILP ... using MILP branch-and-cut
- If $x_I^{(j)}$ integral, then interrupt MILP; solve NLP($x_I^{(j)}$) get $x^{(j)}$



LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

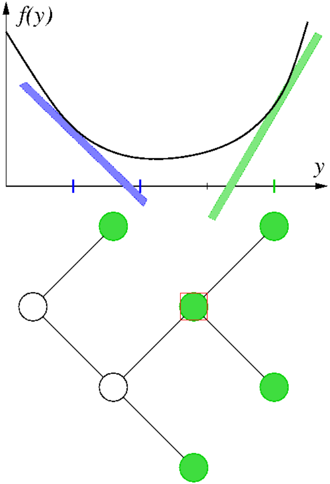
- Start solving master MILP ... using MILP branch-and-cut
- If $x_I^{(j)}$ integral, then interrupt MILP; solve NLP($x_I^{(j)}$) get $x^{(j)}$
- Linearize f, c about $x^{(j)}$ $\Rightarrow$ add linearization to tree



LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

- Start solving master MILP ...
  using MILP branch-and-cut
- If $x_I^{(j)}$ integral, then interrupt MILP; solve NLP($x_I^{(j)}$) get $x^{(j)}$
- Linearize f, c about $x^{(j)}$
  $\Rightarrow$ add linearization to tree
- Continue MILP tree-search

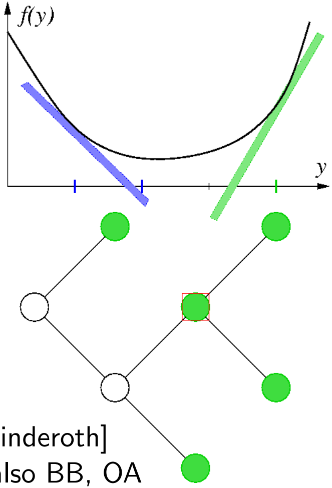


LP/NLP-Based Branch-and-Bound

Aim: avoid solving expensive MILPs

- Start solving master MILP ... using MILP branch-and-cut
- If $x_I^{(j)}$ integral, then interrupt MILP; 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

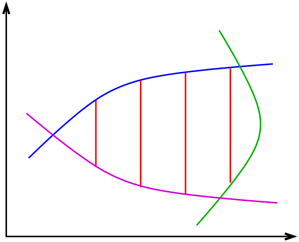


Software:
FilMINT: FilterSQP + MINTO [L & Linderoth]
BONMIN: IPOPT + CBC [IBM/CMU] also BB, OA

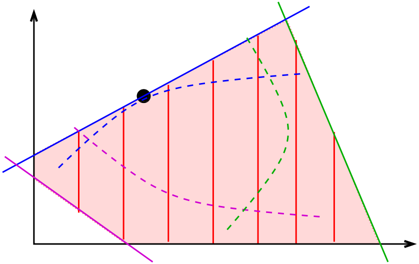
LP/NLP Branch and Bound

LP/NLP-based branch-and-bound

- Branch-and-cut algorithm with cuts from NLP solves
- Create MILP relaxation of MINLP



$0 \geq g(x)$



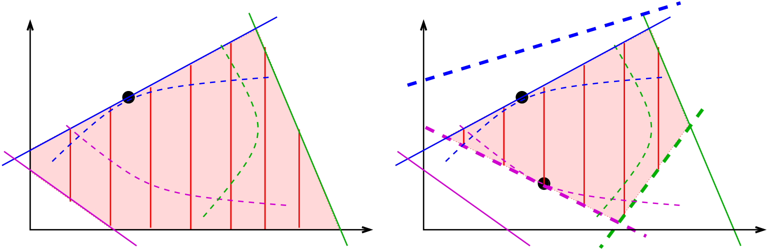
$0 \geq g^{(k)} + \nabla g^{(k)^T}(x - x^{(k)})$

- Search MILP-tree $\Rightarrow$ faster re-solves
- Interrupt MILP tree-search to create new linearizations

LP/NLP Branch and Bound

LP/NLP-based branch-and-bound

- Branch-and-cut algorithm with cuts from NLP solves
- Create MILP relaxation of MINLP & refine linearizations



$0 \geq g(x)$ $\qquad$ $0 \geq g^{(k)} + \nabla g^{(k)^T}(x - x^{(k)})$

- Search MILP-tree $\Rightarrow$ faster re-solves
- Interrupt MILP tree-search to create new linearizations

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.

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.

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

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