A Branch-and-Refine Method for Nonconvex Mixed-Integer Optimization

Sven Leyffer[2] Annick Sartenaer[1] Emilie Wanufelle[1]

[1]University of Namur, Belgium

[2]Argonne National Laboratory, USA

IMA Workshop, Minneapolis, November 18, 2008

Overview

Application & Motivation
- Tertiary Voltage Control
- Problem Characteristics

Branch-and-Refine for Nonconvex MINLPs
- Special Ordered Set (SOS) Approximations
- Decomposition of Nonlinear Functions
- Piecewise Polyhedral Envelopes
- Branch-and-Refine

Theoretical and Numerical Results
- Comparison to Underestimators & SOS-Branching
- Numerical Experience
- Conclusions & Future Research

Overview

- Application & Motivation
  - Tertiary Voltage Control
  - Problem Characteristics
- Branch-and-Refine for Nonconvex MINLPs
  - Special Ordered Set (SOS) Approximations
  - Decomposition of Nonlinear Functions
  - Piecewise Polyhedral Envelopes
  - Branch-and-Refine
- Theoretical and Numerical Results
  - Comparison to Underestimators & SOS-Branching
  - Numerical Experience
  - Conclusions & Future Research

Tertiary Voltage Control Problem

- Provided by Tractebel Engineering
- Optimal Power Flow (OPF) problem: find the best operating conditions of an existing system
- Focus on the Tertiary Voltage Control (TVC) problem (but applicable to more general OPF problems)
- In alternating current, power is a complex number:
  - real part = real power (P)
  - imaginary part = reactive power (Q)
- Reactive power transmission causes voltage drops and losses ⇒ need a regulation of the reactive power produced by each generator

TVC

$$\left\{\begin{array}{lll}
\text{min} & \sum_{k\in N_G} w_k(Q_k - obj_k)^2, & \\
\text{s.t.} & P_i - P_{i_c} - \sum\limits_{ik\in S_i^s} P_{ik} - \sum\limits_{ik\in S_i^e} P_{ik} - \sum\limits_{ik\in T_i^s} P_{ik} - \sum\limits_{ik\in T_i^e} P_{ik} = 0, & \forall i \in N, \\
 & Q_i - Q_{i_c} + a_i {\nu_i}^2 Q_{i_0} - \sum\limits_{ik\in S_i^s} Q_{ik} - \sum\limits_{ik\in S_i^e} Q_{ik} - \sum\limits_{ik\in T_i^s} Q_{ik} - \sum\limits_{ik\in T_i^e} Q_{ik} = 0, & \forall i \in N, \\
 & \sum\limits_{ik\in B^*} Q_{ik} = K, & \\
 & \nu_{min_i} \leq \nu_i \leq \nu_{max_i}, \qquad a_i \text{ binary,} & \forall i \in N, \\
 & P_{min_i} \leq P_i \leq P_{max_i}, & \forall i \in N_G, \\
 & Q_{min_i} \leq Q_i \leq Q_{max_i}, & \forall i \in N_G, \\
 & r_{min_{ik}} \leq r_{ik} \leq r_{max_{ik}}, \qquad r_{ik} \in E_{disc} \text{ discrete,} & \forall ik \in T,
\end{array}\right.$$

where:

$$
\begin{aligned}
P_{ik} &= {\nu_i}^2 (y_{ik} \cos(\zeta_{ik}) + g_{ik}) - \nu_i \nu_k y_{ik} \cos(\zeta_{ik} + \theta_i - \theta_k), && \forall ik \in S_i^e, \\
Q_{ik} &= {\nu_i}^2 (y_{ik} \sin(\zeta_{ik}) - h_{ik}) - \nu_i \nu_k y_{ik} \sin(\zeta_{ik} + \theta_i - \theta_k), && \forall ik \in S_i^e, \\
\\
P_{ik} &= {\nu_i}^2 {r_{ik}}^2 y_{ik} \cos(\zeta_{ik}) - \nu_i \nu_k r_{ik} y_{ik} \cos(\zeta_{ik} + \theta_i - \theta_k), && \forall ik \in T_i^e, \\
Q_{ik} &= {\nu_i}^2 {r_{ik}}^2 y_{ik} \sin(\zeta_{ik}) - \nu_i \nu_k r_{ik} y_{ik} \sin(\zeta_{ik} + \theta_i - \theta_k), && \forall ik \in T_i^e, \\
\\
P_{ki} &= {\nu_k}^2 (y_{ik} \cos(\zeta_{ik}) + g_{ik}) - \nu_i \nu_k y_{ik} \cos(\zeta_{ik} + \theta_k - \theta_i), && \forall ki \in S_i^s, \\
Q_{ki} &= {\nu_k}^2 (y_{ik} \sin(\zeta_{ik}) - h_{ik}) - \nu_i \nu_k y_{ik} \sin(\zeta_{ik} + \theta_k - \theta_i), && \forall ki \in S_i^s, \\
\\
P_{ki} &= {\nu_k}^2 (y_{ik} \cos(\zeta_{ik}) + y_{0_{ik}} \cos(\zeta_{0_{ik}})) - \nu_i \nu_k r_{ik} y_{ik} \cos(\zeta_{ik} + \theta_k - \theta_i), && \forall ki \in T_i^s, \\
Q_{ki} &= {\nu_k}^2 (y_{ik} \sin(\zeta_{ik}) + y_{0_{ik}} \sin(\zeta_{0_{ik}})) - \nu_i \nu_k r_{ik} y_{ik} \sin(\zeta_{ik} + \theta_k - \theta_i), && \forall ki \in T_i^s.
\end{aligned}
$$

→ highly nonlinear, nonconvex

Problem Characteristics

- Nonconvex equality constraints with $\sin(w)$, $\cos(w)$, w^2, $w_1 w_2$
- Bounds and network constraints
- Discrete variables:
  - binary: $a_i \quad (i \in N)$ → variables on/off
  - discrete: $r_{ik} \in E_{disc} \quad (ik \in T)$ → $r_{ik} = \frac{N_{P_{ik}}}{N_{S_{ik}}}$

→ <u>Nonconvex MINLP</u>:

$$(P) \left\{ \begin{array}{ll} \underset{x,y}{\text{minimize}} & g_0(x,y), \\ \text{subject to} & g_i(x,y) = 0, \ i = 1,..,m, \\ & (x,y) \in P, \ y \in \mathbb{Z}^t \end{array} \right.$$

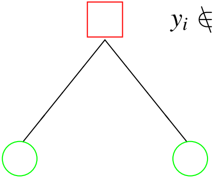
Branch-and-bound for MINLP (binary variables)

Drop the binary restrictions.
Solve the continuous NLP.

Continuous relaxation

Branch-and-bound for MINLP (binary variables)

Choose a variable $\notin \{0,1\}$. Split the problem into 2 new subproblems.



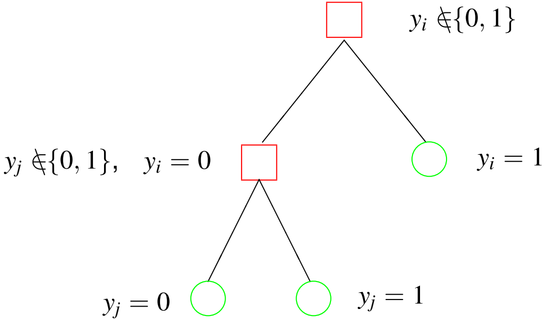
$y_i \notin \{0,1\}$

$y_i = 0$

$y_i = 1$

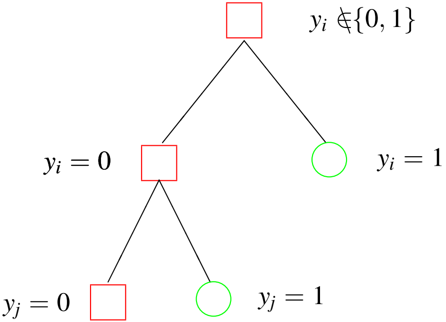
Branch-and-bound for MINLP (binary variables)

Choose a subproblem and solve it.
Choose a variable $\notin \{0, 1\}$.
Split the problem into 2 new subproblems.



Branch-and-bound for MINLP (binary variables)

The NLP is infeasible, cut the node.



$y_i \notin \{0, 1\}$

$y_i = 0$

$y_i = 1$

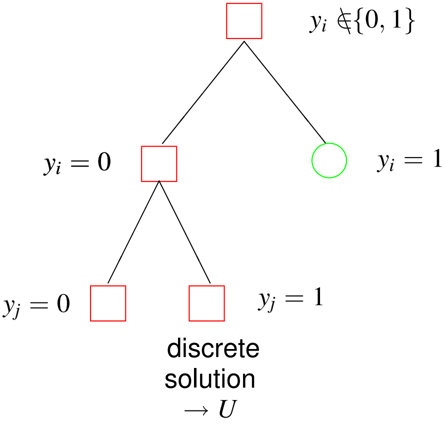
$y_j = 0$

$y_j = 1$

infeasible
problem

Branch-and-bound for MINLP (binary variables)

If x_{NLP} is feasible for MINLP, store U and cut the node.



$y_i \notin \{0, 1\}$

$y_i = 0$

$y_i = 1$

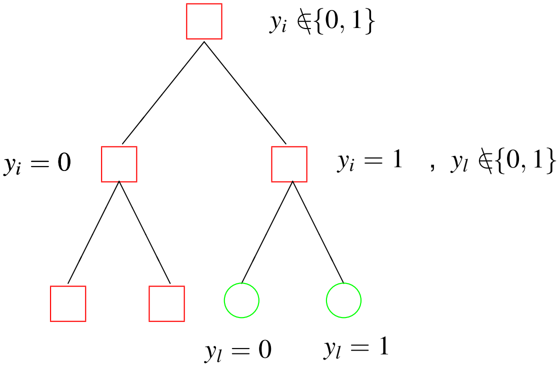
$y_j = 0$

$y_j = 1$

discrete solution $\rightarrow U$

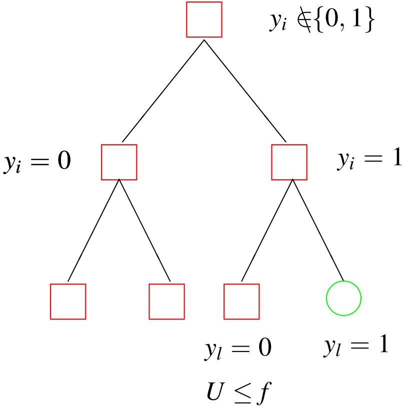
Branch-and-bound for MINLP (binary variables)

Choose a subproblem and solve it.
Choose a variable $\notin \{0, 1\}$.
Split the problem into 2 new subproblems.



Branch-and-bound for MINLP (binary variables)

The optimum value of NLP is larger than the current upper bound on MINLP. Cut the node.



$y_i \notin \{0,1\}$

$y_i = 0$

$y_i = 1$

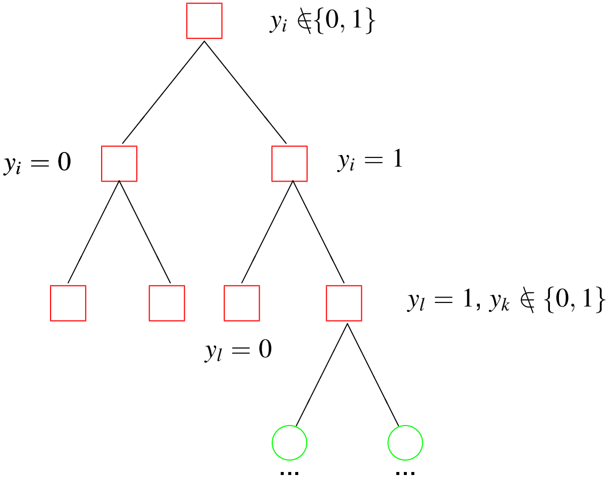
$y_l = 0$

$y_l = 1$

$U \leq f$

Branch-and-bound for MINLP (binary variables)

Pursue like this until the whole tree has been explored.



$y_i \notin \{0,1\}$

$y_i = 0$

$y_i = 1$

$y_l = 1,\ y_k \notin \{0,1\}$

$y_l = 0$

...

...

⟶ Branch-and-bound for nonconvex problems:

- no guarantee to converge
- the continuous nonlinear relaxations are expensive to solve

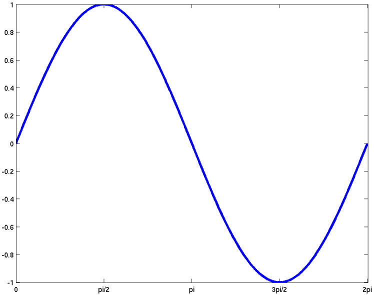
Our goal

Find the **global solution** of the TVC problem
by solving a sequence of appropriate **linear** problems

Overview

Linear Approximation of a Nonconvex Function

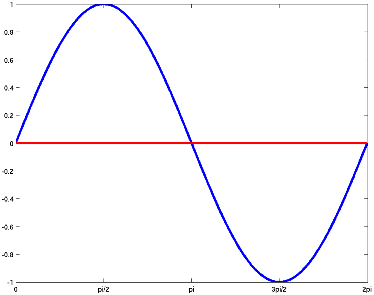
e.g.: sin(x)



Linear Approximation of a Nonconvex Function

e.g.: sin(x)

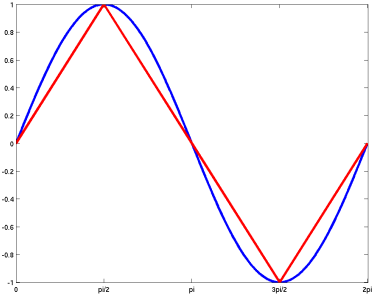
→ not accurate



Linear Approximation of a Nonconvex Function

e.g.: sin(x)

→ piecewise linear approximation



Special Ordered Set (SOS) Approximations (1 dimension)

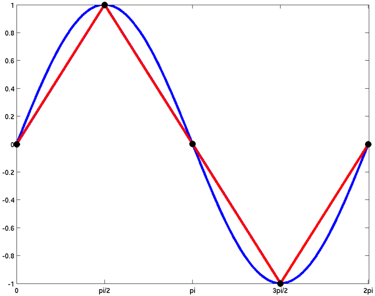
Choose n breakpoints x_i and set

$$f(x) \approx \tilde{f}(x) = \sum_{i=1}^{n} \lambda_i f(x_i)$$

with

$$x = \sum_{i=1}^{n} \lambda_i x_i$$

$$\sum_{i=1}^{n} \lambda_i = 1, \ \lambda_i \geq 0, \quad i = 1 : n$$



Refs: Beale, Tomlin

Special Ordered Set (SOS) Approximations (1 dimension)

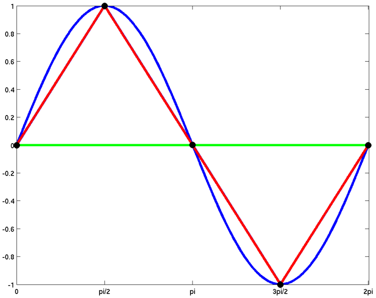
Choose n breakpoints x_i and set

$$f(x) \approx \tilde{f}(x) = \sum_{i=1}^{n} \lambda_i f(x_i)$$

with

$$x = \sum_{i=1}^{n} \lambda_i x_i$$

$$\sum_{i=1}^{n} \lambda_i = 1, \ \lambda_i \geq 0, \quad i = 1 : n$$



Refs: Beale, Tomlin

Special Ordered Set (SOS) Approximations (1 dimension)

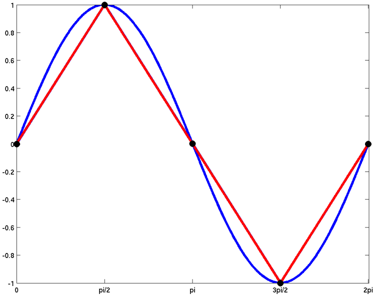
Choose n breakpoints x_i and set

$$f(x) \approx \tilde{f}(x) = \sum_{i=1}^{n} \lambda_i f(x_i)$$

with

$$x = \sum_{i=1}^{n} \lambda_i x_i$$

$$\sum_{i=1}^{n} \lambda_i = 1, \ \lambda_i \geq 0, \quad i = 1 : n$$



SOS condition:
At most 2 λ_i can be nonzero
and these λ_i must be adjacent

Refs: Beale, Tomlin

SOS approximation method

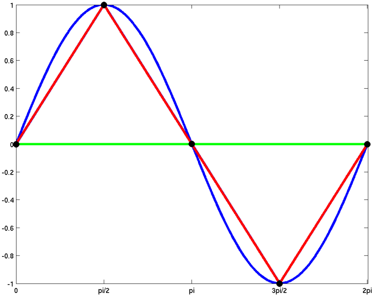
- Construction of a linear approximation problem subject to SOS conditions satisfied via a branch-and-bound process
- Branching on the SOS variables λ_i
- Used successfully by Martin *et al.* to solve nonconvex MINLP problems arising in gas network management

Remark: in our case

Relaxation of the SOS condition

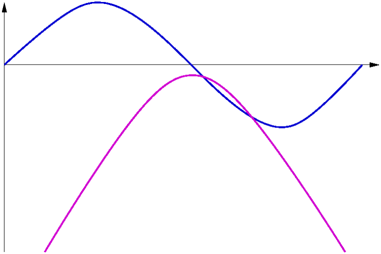
(No branching on SOS variables)

$\longrightarrow$ “SOS-like” approx.



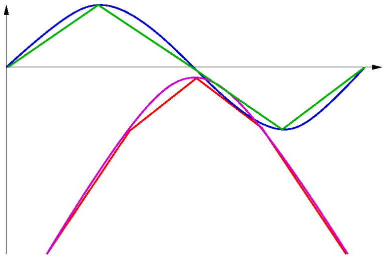
Infeasible SOS Approximations

- Solution of an approximation problem
- Physical constraints must be satisfied
- Equality constraints
- Solution has little chance to be feasible for our problem



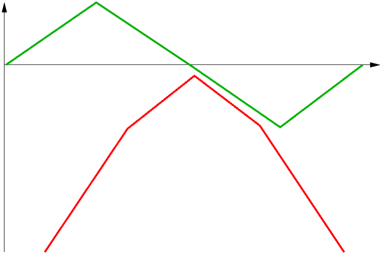
Infeasible SOS Approximations

- Solution of an approximation problem
- Physical constraints must be satisfied
- Equality constraints
- Solution has little chance to be feasible for our problem

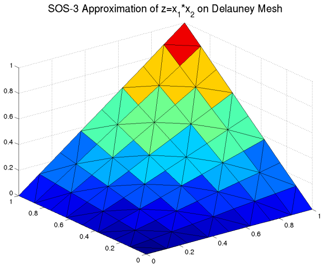


Infeasible SOS Approximations

- Solution of an approximation problem
- Physical constraints must be satisfied
- Equality constraints
- Solution has little chance to be feasible for our problem



Exponential Complexity of SOS Approximations



- approximate $h(x,y)$ for $(x,y) \in \mathbb{R}^n$
- p breakpoints in each dimension
- p^n SOS-variables λ_i

e.g., expression for real power has $n = 8$ variables → impractical

Decomposition of Nonlinear Functions

Idea: decompose $h(x, y)$ into simpler functions:

$$\begin{array}{lcll} w_j & = & x_j & j = 1, \ldots, s, \\ w_{s+j} & = & y_j & j = 1, \ldots, t, \\ w_{s+t+j} & = & h_j(w_{j_1}\{, w_{j_2}\}) & j = 1, \ldots, K, \end{array}$$

where h_j are univariate or bivariate and $j_1, j_2 < s + t + j$

$$\longrightarrow \quad h(x, y) = w_{s+t+K}$$

Only 3 kinds of nonlinear h_j functions for the TVC problem:

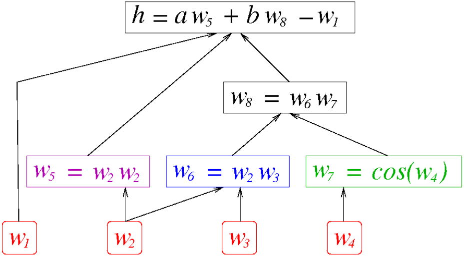
- square functions: x^2
- trigonometric functions: $\sin(x)$, $\cos(x)$
- bilinear functions: xy

Example

Consider, for some constants a and b:

$$h(x_1, x_2, x_3, x_4) = ax_2^2 + bx_2x_3\cos(x_4) - x_1$$

- $w_j = x_j \quad j = 1:4$
- $w_5 = w_2^2$
- $w_6 = w_2w_3$
- $w_7 = \cos(w_4)$
- $w_8 = w_6w_7$
- $h = aw_5 + bw_8 - w_1$



$h = a\,w_5 + b\,w_8 - w_1$

$w_8 = w_6\,w_7$

$w_5 = w_2\,w_2$

$w_6 = w_2\,w_3$

$w_7 = cos(w_4)$

w_1

w_2

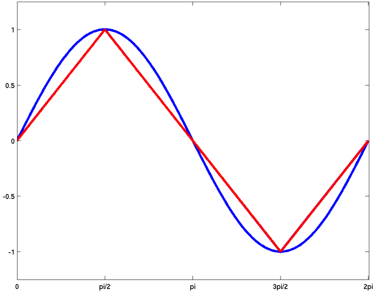
w_3

w_4

Remark: decomposition not unique, e.g., $w_6 = \cos(w_4)$, etc.

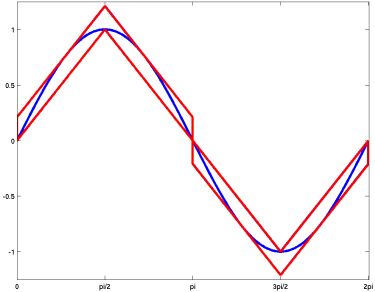
Piecewise Polyhedral Envelopes

Idea: Outer approximation by piecewise polyhedral envelopes



Piecewise Polyhedral Envelopes

Idea: Outer approximation by piecewise polyhedral envelopes



Determination of an Envelope

For each nonlinear function h (e.g., x^2, $\sin(x)$, $\cos(x)$, xy), determine the expression of the maximum approximation errors done by its SOS approximation:

$$\begin{aligned} \epsilon_{L,\text{max}} &= \max_x(\tilde{h}(x)_{\text{SOS}} - h(x), 0), \\ \epsilon_{U,\text{max}} &= \max_x(h(x) - \tilde{h}(x)_{\text{SOS}}, 0), \end{aligned}$$

and approach $h(x)$ by $w_{h(x)}$ where:

$$\tilde{h}(x) - \epsilon_{L,\text{max}} \leq w_{h(x)} \leq \tilde{h}(x) + \epsilon_{U,\text{max}}$$

Approximation Errors

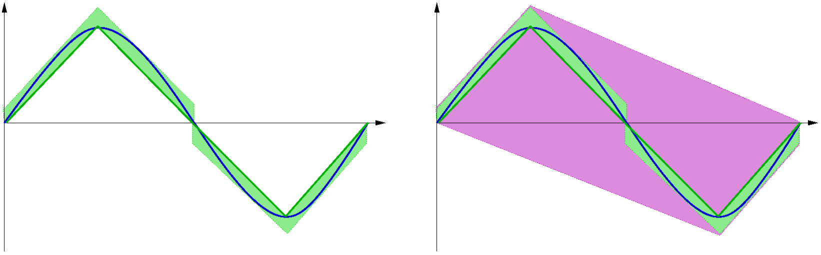
Obtain maximum overestimation and underestimation errors of SOS approximation on $[x^k, x^{k+1}]$:

→ Pre-computed bounds on $[x_k, x_{k+1}]$, e.g., for x^2:

$$\epsilon_{x^2,L_k} = \frac{(x_{k+1}-x_k)^2}{4} \quad \text{and} \quad \epsilon_{x^2,U_k} = 0$$

(See Emilie's thesis for other functions)

Illustration with or without SOS Condition



With SOS condition

Without SOS condition

Piecewise Polyhedral Envelopes Surprise

Theorem: Every (x, y, xy) with $l_x \leq x \leq u_x$ and $l_y \leq y \leq u_y$ is unique convex combination of $(l_x, l_y, l_x l_y)$, $(l_x, u_y, l_x u_y)$, $(u_x, l_y, u_x l_y)$ and $(u_x, u_y, u_x u_y)$, i.e. $\exists\ \lambda_i \geq 0,\ i = 1, \ldots, 4$:

$$\begin{pmatrix} x \\ y \\ xy \\ 1 \end{pmatrix} = \begin{bmatrix} l_x & l_x & u_x & u_x \\ l_y & u_y & l_y & u_y \\ l_x l_y & l_x u_y & u_x l_y & u_x u_y \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix}$$

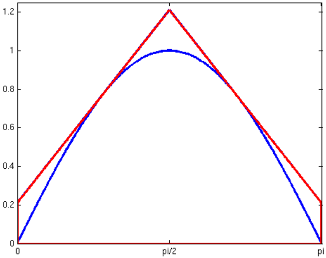
$\longrightarrow$ No need of envelope for xy !

Refinement of Approximations

- "Piecewise envelope" problem : (much) larger feasible domain than that of the original problem

  → Need to refine the approximations

- Use of branch-and-bound to reduce the approximation intervals (the tighter the domain, the better the approximations)
- Use the same number of breakpoints → $\boxed{Branch - and - Refine}$

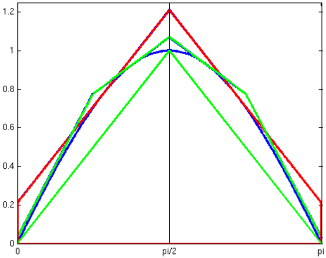


Refinement of Approximations

- "Piecewise envelope" problem : (much) larger feasible domain than that of the original problem

  → Need to refine the approximations

- Use of branch-and-bound to reduce the approximation intervals (the tighter the domain, the better the approximations)
- Use the same number of breakpoints → $\boxed{Branch - and - Refine}$



Refinement of Approximations

- “Piecewise envelope” problem : (much) larger feasible domain than that of the original problem

  → Need to refine the approximations

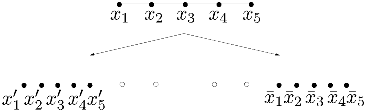
- Use of branch-and-bound to reduce the approximation intervals (the tighter the domain, the better the approximations)
- Use the same number of breakpoints → $\boxed{Branch - and - Refine}$
- Ideal framework for discrete problems
- Convergence to the global solution (within some accuracy) is guaranteed under mild assumptions

Branch-and-Refine

- Solve piecewise envelope problem
- Branch on continous variables or integer variables
- Refine and tighten the envelope while going down the tree
- Exploit exactness of bilinear terms

Branch-and-Refine: Branching

1 dimension



2 dimension

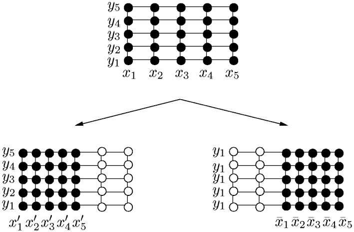


Illustration of branching and refinement

Branch-and-Refine: Fathoming Rules

Assume at node k: $x \in X_k$ and $y \in Y_k$ (continuous domains)

- LP(X_k, Y_k): LP relaxation of piecewise envelope problem
- NLP(X_k, Y_k):

$$\left\{ \begin{array}{llll} z_{\text{NLP}_k} & := & \underset{x,y}{\text{minimize}} & g_0(x,y) \\ & & \text{subject to} & g_i(x,y) = 0, \; i = 1,..,m \\ & & & x \in X_k, \; y \in Y_k \end{array} \right.$$

Fathoming Rules:

1. Infeasible LP(X_k, Y_k) relaxation
2. NLP(X_k, Y_k) solution same as LP(X_k, Y_k) relaxation
3. LP(X_k, Y_k) relaxation dominated by incumbent

Branch-and-Refine: Algorithm

```
set $U = \infty$, $k = 1$ & put LP($X_k, Y_k$) on stack
while stack is not empty
    solve LP($X_k, Y_k$) $\longrightarrow$ solution $(x^k, y^k)$
    if LP($X_k, Y_k$) infeasible or $z_{\mathsf{LP}_k} \geq U - \epsilon$ then
        fathom node (Case 1. or 3.)
    else
        solve NLP($X_k, Y_k$) $\longrightarrow$ solution $(\hat{x}^k, \hat{y}^k)$
        if $z_{\mathsf{NLP}_k} < U - \epsilon$ & $\hat{y}^k$ integer then
            update $U := z_{\mathsf{NLP}_k}$ & incumbent $(x^*, y^*) := (\hat{x}^k, \hat{y}^k)$
        if $|z_{\mathsf{NLP}_k} - z_{\mathsf{LP}_k}| \leq \epsilon$ then
            fathom node (Case 2.)
        else
            branch (creating two new LPs) and refine
    k = k+1
```

Overview

Application & Motivation
- Tertiary Voltage Control
- Problem Characteristics

Branch-and-Refine for Nonconvex MINLPs
- Special Ordered Set (SOS) Approximations
- Decomposition of Nonlinear Functions
- Piecewise Polyhedral Envelopes
- Branch-and-Refine

Theoretical and Numerical Results
- Comparison to Underestimators & SOS-Branching
- Numerical Experience
- Conclusions & Future Research

Comparison to Caratzoulas & Floudas

[Caratzoulas & Floudas, 2004] construct convex underestimators:

$$U_{\sin}(x) = -15.72 \sin\left(\frac{1}{6}(x + 2\pi)\right) + 13.61,$$

and

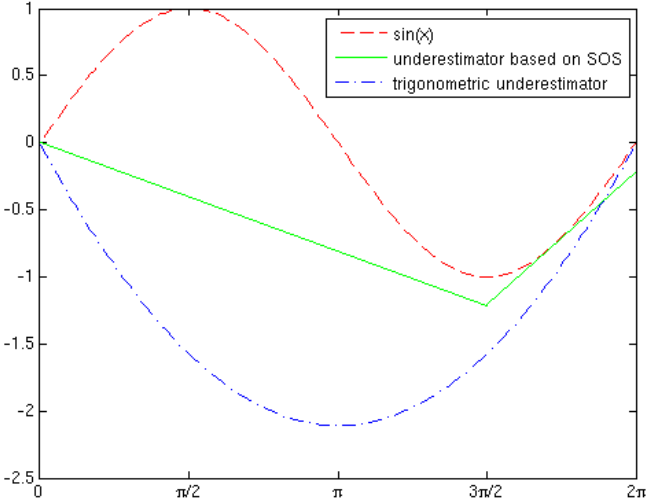
$$U_{\cos}(x) = -16.99 \sin\left(\frac{1}{6}(x + 2\pi)\right) + 15.72.$$

Theorem: Let $A_{SOS}^{\sin}$, $A_{SOS}^{\cos}$ be SOS underestimation errors and $A_{trig}^{\sin}$, $A_{trig}^{\cos}$ be underestimation errors of [C&F, 2004]. Then

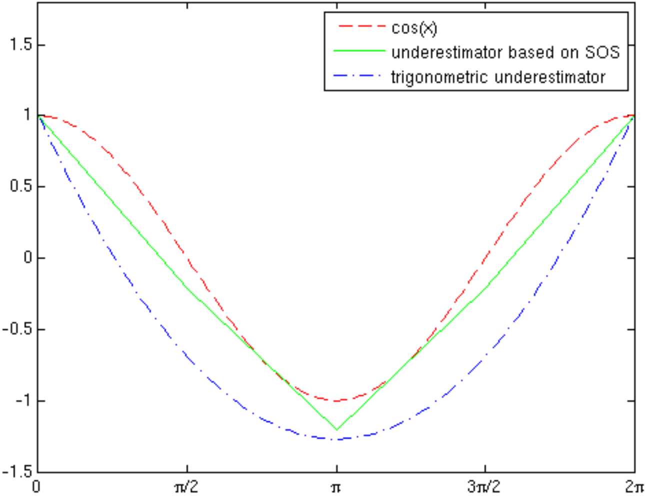
$$A_{SOS}^{\sin} = 0.451\, A_{trig}^{\sin} \quad \text{and} \quad A_{SOS}^{\cos} = 0.313\, A_{trig}^{\cos}$$

$\longrightarrow$ piecewise polyhedra are tighter on $[0, 2\pi]$.

Comparison to Caratzoulas & Floudas: $\sin(x)$

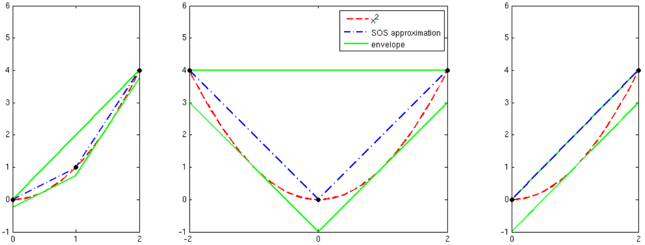


Comparison to Caratzoulas & Floudas: $\cos(x)$



Comparison to SOS-Branching

Example: x^2



$\longrightarrow$ Better to refine than to branch on SOS variables (factor $11/12$)

Test Problems (Generic)

prob	#var	#cons	#var OA	#cons OA	#sets λ	#disc
pb0	4	2	44	32	6	1
pb1	4	2	44	32	6	1
pb2	6	2	41	30	5	1
pb3	6	2	41	30	5	1
pb4	12	4	97	71	11	2
pb5	12	4	97	71	11	2
pb6	12	4	143	97	19	3
pb7	12	4	143	97	19	3
pb8	12	4	119	77	14	2
pb9	12	4	119	77	14	2
pb10	10	4	111	72	13	2
pb11	10	4	111	72	13	2
pb12	24	8	275	187	40	6
pb13	24	8	275	187	40	6

Test Problems (Tertiary Voltage Control)

prob	#var	#cons	#var OA	#cons OA	#sets λ	#disc
TVC1	16	9	269	200	39	6
TVC2	18	9	275	204	40	6
TVC3	27	15	422	315	61	9
TVC4	27	15	422	315	61	9
TVC5	37	21	602	449	87	13
TVC6	38	21	635	472	92	14

→ moderately sized problems

Comparison with Other Softwares Available on NEOS

- Nonlinear global optimization solvers available on NEOS:

  BARON: not directly applicable due to $\sin(x), \cos(x)$

  Possibility: replace each trigonometric function by its Taylor approximation of degree 7 (Bussieck)

  $\rightarrow$ but not accurate

- Nonlinear local optimization solvers available on NEOS:

  \+ cheaper than our method since local methods

  \- no guarantee to converge to the global optimum

Comparison with Local Methods

For NLP:

	Branch-and-Refine	IPOPT	KNITRO	FilterSQP
# solved	20	17	17	12
# global solution	20	14	13	8

For MINLP:

	Branch-and-Refine	Bonmin	MINLP_BB
# solved	20	15	11
# global solution	20	11	9

Implementation Details & Tricks

- LPs solved with CPLEX
- decomposition (hand-coded by Emilie)
  - exploit common sub-expressions
  - can be automated, similar to automatic differentiation (AD)
- NLPs solved with FilterSQP (AD for gradients/Hessians)
- propagate & strengthen bounds through computational graph
- pre-solve (LP) to reduce range of variables (like BARON)
  - adaptive presolve is best: trail-off factor
- (generalized) pseudo-cost branching
- (generalized) best-estimate node selection

Numerical Results (# LPs solved)

prob	basic	+presolve	+var-select	+node-select
pb0	63	63	68	68
pb1	133	131	79	68
pb2	2115	3237	194	260
pb3	135	197	121	97
pb4	15389	11388	120	120
pb5	3009	257	145	145
pb6	65800	6145	348	292
pb7	377	1353	1235	1121
pb8	fail	198817	263	241
pb9	62149	33668	442	442
pb10	113846	51816	205	197
pb11	3806	7349	558	258
pb12	fail	33407	1503	1056
pb13	fail	8093	17388	3885

Numerical Results (# LPs solved)

prob	basic	+presolve	+var-select	+node-select
TVC1	108861	40446	7756	8031
TVC2	fail	72270	5792	5547
TVC3	62045	861	627	627
TVC4	fail	38792	1396	1582
TVC5	fail	7369	5619	4338
TVC6	fail	12131	6096	5503

Conclusions & Future Research

Branch-and-Refine

- Three key ingredients:
  1. decompose functions into 1D and 2D components
  2. construct piecewise polyhedral envelope
  3. branch on (continuous or integer) variables, not on SOS ones
- favorable theoretical properties
- encouraging numerical results

Future Work

- exploit expression tree ... use AD tricks for better OA
- what decomposition is best ... non-unique
- avoid λ variables ... work with OA directly
- avoid SOS approx. for convex functions ... NLP subproblems
- efficient implementation & support for `AMPL`, `GAMS`