Development of an algorithm for solving mixed integer and nonconvex problems arising in electrical supply networks

E. Wanufelle[1] S. Leyffer[2] A. Sartenaer[1] Ph. Toint[1]

[1]FUNDP, University of Namur

[2]Argonne National Laboratory

One-day Symposium on Optimization and Engineering, CORE, May 24th 2006

Outline

The considered problem

The problem of tertiary voltage control (TVC)

- In alternating current: power is a complex number
  real part = real power
  imaginary part = reactive power
- reactive power transmission causes voltage drops and losses
  $\Rightarrow$ need a regulation of the reactive power produced by each generator of an electrical network
- under some physical laws
- problem and model provided by Tractebel Engineering

Modelling of the problem

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

Modelling of the problem (continued)

where

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

$\Rightarrow$ highly nonlinear, nonconvex

Use of discrete variables

- a_i: binary $\quad (i \in N)$
  $\longrightarrow$ variables on/off
- $r_{ik} \in E_{disc}$: discrete $\quad (ik \in T)$
  e.g.: $E_{disc} = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
  $\longrightarrow$ the transformer ratio can only be equal to some fixed values

$\Rightarrow$ Mixed Integer NonConvex Programming problem

Motivation

Current approach: heuristics:

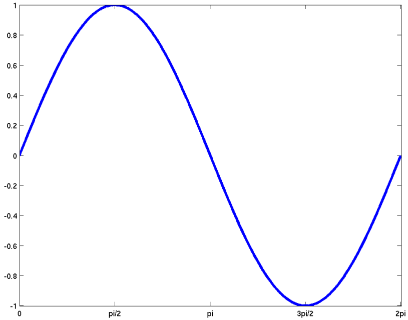
Successive solutions of relaxed nonlinear problems

$\Rightarrow$ wish to work with more reliable/robust methods

Idea: use an appropriate linear approximation of the problem

How can we approximate a nonlinear component by a linear function?

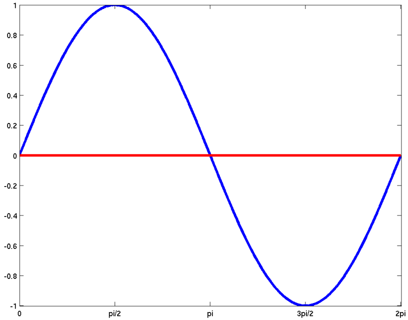
e.g.: sin



How can we approximate a nonlinear component by a linear function?

e.g.: sin

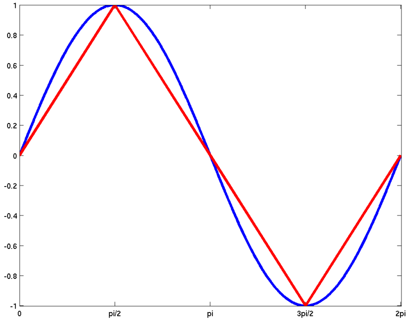
$\rightarrow$ not accurate



How can we approximate a nonlinear component by a linear function?

e.g.: sin

$\longrightarrow$ piecewise linear approximation



Approximation by special ordered sets

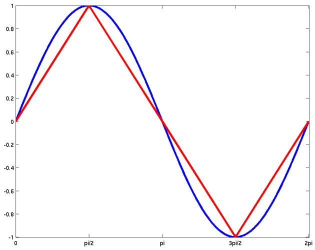
To approximate $f(x)$ by $\tilde{f}(x)$, we use

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

where x_i are breakpoints, $i = 1, n$

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

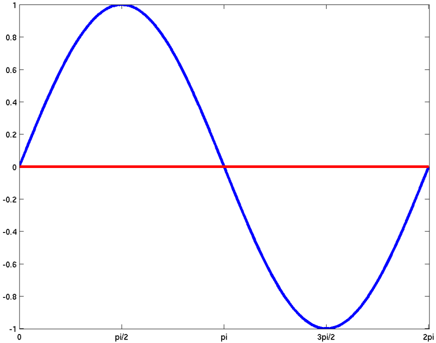
$$\sum_{i=1}^{n} \lambda_i = 1, \ \lambda_i \geq 0, \quad i = 1, n$$



Refs: Beale, Tomlin, Martin

SOS condition: motivation

If $\lambda_1 \neq 0, \ \lambda_5 \neq 0$
$\lambda_i = 0, \ i = 2, .., 4$



SOS formulation (1 dimension)

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

where x_i are breakpoints, $i = 1, .., n$

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

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

SOS type 2 condition:

At most 2 λ_i can be nonzero.

Moreover, these λ_i must be adjacent.

SOS formulation (1 dimension)

$$\left.\left.\begin{array}{l} f(x) \approx \tilde{f}(x) = \sum_{i=1}^{n} \lambda_i f(x_i) \\ \text{where} \quad x_i \text{ are breakpoints}, \quad i = 1, .., n \\ \qquad x = \sum_{i=1}^{n} \lambda_i x_i \\ \qquad \sum_{i=1}^{n} \lambda_i = 1, \; \lambda_i \geq 0, \quad i = 1, ..., n \end{array}\right\} (LP) \\ \begin{array}{l} \text{At most 2 } \lambda_i \text{ can be nonzero.} \\ \text{Moreover, these } \lambda_i \text{ must be adjacent.} \end{array} \right\} \begin{array}{c} (LP) \\ + \\ \text{branching} \end{array}$$

SOS formulation (2 dimensions)

$$f(x,y) \approx \tilde{f}(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} f(x_i, y_j)$$

where (x_i, y_j) are breakpoints, $i = 1,..,n,\ j = 1,..,m$

$$x = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} x_i$$

$$y = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} y_j$$

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_{ij} = 1,\ \lambda_{ij} \geq 0, \quad i = 1,...,n,\ j = 1,..,m$$

At most 3 λ_{ij} can be nonzero.
Moreover, these λ_{ij} must be adjacent on a triangle.

Illustration: xy

On $[-2:2] \times [-2:2]$:

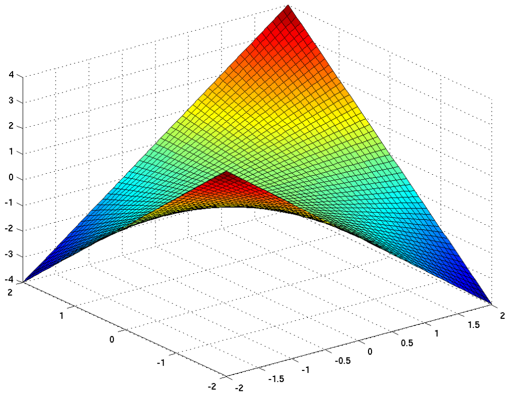


Illustration: xy

Approximation by SOS: 3 breakpoints are used in each dimension

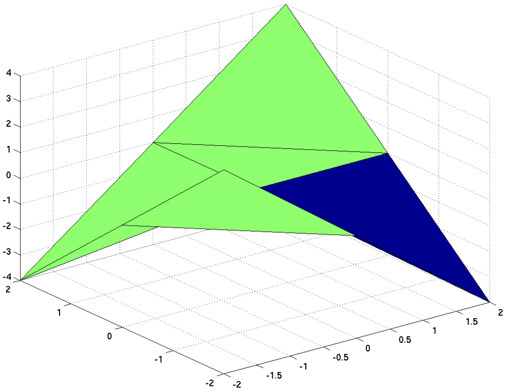
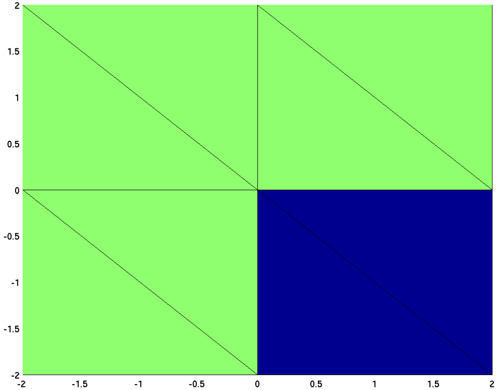


Illustration: xy

Dividing the feasible domain into triangles



Approximation by special ordered sets (3 dimensions and more)

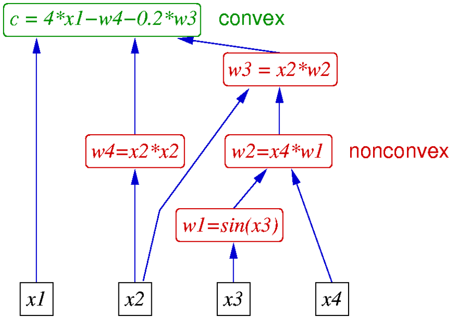
- the same reasoning could be used
- BUT introduction of a lot of variables into the problem: for k breakpoints in each dimension:

$$\begin{array}{ll} \text{1 dim :} & k \text{ var } \lambda \\ \text{2 dim :} & k^2 \text{ var } \lambda \\ \text{3 dim :} & k^3 \text{ var } \lambda \\ \ldots & \\ \text{n dim :} & k^n \text{ var } \lambda \end{array}$$

Idea: decompose problem into components of 1 or 2 variables

Decomposition of the problem

Computational graph for
$c = 4x_1 - {x_2}^2 - 0.2x_2x_4\sin(x_3)$



- Decomposition of the problem into nonlinear components of 1 or 2 variables
- Approximation of each of these nonlinear components by new variables
- Computational graph not unique

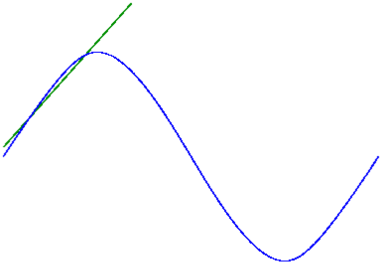
Main components of the problem

3 main kinds of nonlinear components:

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

Insufficient approximation

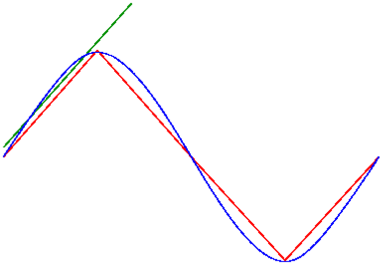
- Building of a linear approximation problem subject to SOS conditions
- There exists an efficient method (Martin)
- – solution of an approximation problem
  – the solution of that problem has little chance to be feasible for our our problem
  – physical constraints must be absolutely satisfied



$\Rightarrow$ use outer approximations to guarantee solution

Insufficient approximation

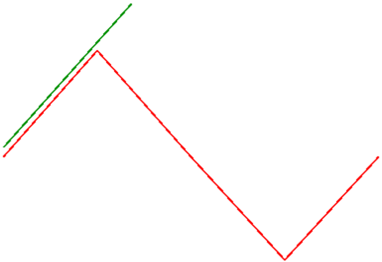
- Building of a linear approximation problem subject to SOS conditions
- There exists an efficient method (Martin)
- – solution of an approximation problem
  – the solution of that problem has little chance to be feasible for our our problem
  – physical constraints must be absolutely satisfied



$\Rightarrow$ use outer approximations to guarantee solution

Insufficient approximation

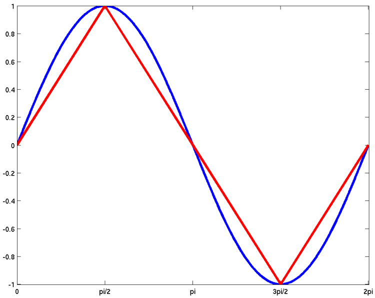
- Building of a linear approximation problem subject to SOS conditions
- There exists an efficient method (Martin)
- – solution of an approximation problem
  – the solution of that problem has little chance to be feasible for our our problem
  – physical constraints must be absolutely satisfied



$\Rightarrow$ use outer approximations to guarantee solution

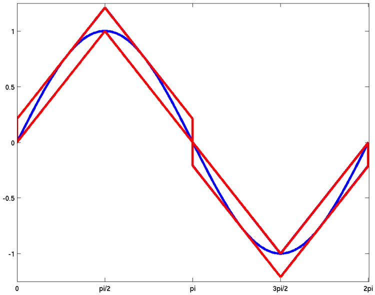
Outer approximations

Idea: replace each nonlinear component f by a linear domain which includes the nonlinear function.



Outer approximations

Idea: replace each nonlinear component f by a linear domain which includes the nonlinear function.



Idea recently used (Gatzke)

Difference: use of linear big M approximations instead of SOS approximations

Determination of an outer domain

For each component f, compute the approximation errors

$$\begin{aligned} \epsilon_L(x_i, x_{i+1}) = & \quad \max_{x \in [x_i, x_{i+1}]}(\tilde{f}(x) - f(x), 0) \\ \epsilon_U(x_i, x_{i+1}) = & \quad \max_{x \in [x_i, x_{i+1}]}(f(x) - \tilde{f}(x), 0) \end{aligned}$$

and replace $f(x) \approx \tilde{f}(x)$ by

$$\tilde{f}(x) - \epsilon_L(x_i, x_{i+1}) \leq f(x) \leq \tilde{f}(x) + \epsilon_U(x_i, x_{i+1}), \quad x_i \leq x \leq x_{i+1}$$

on each piece

Too coarse approximations

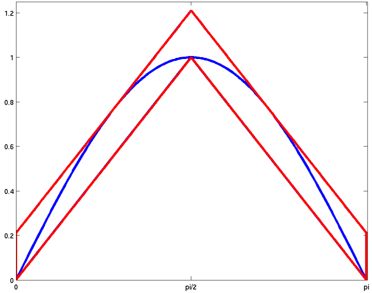
- Results in an outer approximation
- BUT its solution can be very far from the true solution

$\longrightarrow$ Need to refine the approximations

Refinement of approximations

$\longrightarrow$ Use of a branch-and-bound tree:
reduce the approximation interval, refine the mesh

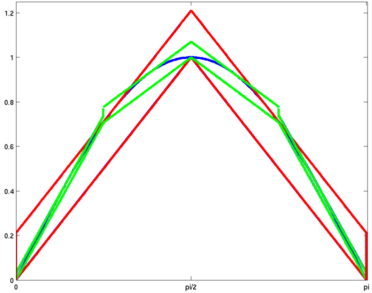
- better approximations



Refinement of approximations

→ Use of a branch-and-bound tree:
reduce the approximation interval, refine the mesh

- better approximations



Refinement of approximations

$\longrightarrow$ Use of a branch-and-bound tree:
reduce the approximation interval, refine the mesh

- better approximations
- ideal framework to treat discrete variables
  $\longrightarrow$ 2 types of division
- guaranteed convergence to the global optimum in the end

Choices associated to the branch-and-bound process

- Choice of the node to refine:
  depth-first search
- Choice of the variable to divide:
  - the variable of the starting problem leading to the largest error of approximation
  - not on the SOS variables λ . . . inefficient
- Upper bound:
  the solution of the 1st linear problem is employed as starting point for the NLP problem to generate an upper bound

Algorithm

1. Build an outer approximation problem, (LP^0), for (P)
   $k := 0$
2. Propagate bounds through the computational graph and compute the approximation errors. Update (LP^k)
3. Solve $(LP^k) \rightarrow (\tilde{x}, \tilde{f})$
   If $\tilde{f} \geq U \Rightarrow$ the node can be cut,
   else if $\tilde{x}$ is feasible for (P) and $f(\tilde{x}) < U$
   $\Rightarrow U = f(\tilde{x}),\ x^* = \tilde{x}$ and the node can be cut
   else choose a variable j and divide the pbm (LP^k) into 2 new subproblems
4. **If** the tree is completely explored: STOP
   else $k := k + 1$
   choose a node which has not been examined yet and go to 2.

Numerical results

Toy problem:

$$
(P)\left\{\begin{array}{ll}
\min & w_1 \sin w_4 \\
\text{s.t.} & 4w_1 - {w_2}^2 - 0.2 w_2 w_4 \sin w_3 \leq 1 \\
 & w_2 - 0.5 w_2 w_4 \cos w_3 \leq -2 \\
 & 0 \leq w_1 \leq 4 \\
 & 0 \leq w_2 \leq 3 \\
 & 0 \leq w_3 \leq 2\pi \\
 & 0 \leq w_4 \leq 2\pi
\end{array}\right.
$$

- no discrete variables
- 5 breakpoints for the trigonometric components, 3 for the others
- approximation problem: 69 variables and 46 constraints

Numerical results (continued)

- Nonlinear local optimization solvers available on NEOS: KO for 87.5% of the solvers
- Nonlinear global optimization solver, ACRS: OK but random
- BARON: not applicable due to $\sin(x), \cos(x)$
- Our method: global solution obtained (and proved) after the solution of 103 LP and 2 NLP ($\epsilon =$ 10E-6).

Future work

- More tests problems
- Increase the speed of convergence by
  - improving presolve
  - developing better rules to choose the variable to divide and the place to divide
  - testing finer approximations (quadratic, inequalities of McCormick,...)
  - adding cuts to the problem of approximation
- Introduction of discrete variables into the problem

Conclusion

- Promising approach
- Able to ensure convergence to the global optimum
  But convergence can be slow
- Solution of linear problems only
  But needs the introduction of new variables and constraints into the approximation problem