Introduction to Mixed Integer Nonlinear Programming

CO@Work Berlin

Stefan Vigerske

Humboldt University Berlin

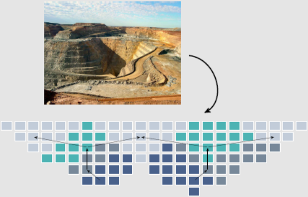
10/08/2009

Recall Monday last week (28/09/2009)...

Recall Monday last week (28/09/2009)...

Open Pit Mining Production Scheduling

Input



- ▷ block model with precedences
- ▷ deterministic block content
- ▷ ore prices and production costs
- ▷ equipment capacities
- ▷ possibly aggregation of blocks

Problem

Find an order of excavation with

- ▷ max. net present value

satisfying

- ▷ precedence constrai
- ▷ resource constraints.

Simultaneously: Detern

- ▷ opt. processing decis

for each block.

Ambros M. Gleixner (ZIB) Open Pit Mine Production Scheduling 09/

Open Pit Mine Production Scheduling Problem

- $x_{i,t} \in \{0,1\}$ whether block i is completely mined in period t
- $y_{i,t} \in [0,1]$ fraction of block i mined during period t
- $z_{i,t} \in [0,1]$ fraction of block i processed during period t

$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i z_{i,t}}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. (x, y) satisfy precedence constraints

$$\sum_t y_{i,t} \leq 1 \quad \forall i \qquad \text{(mine not more than the whole block)}$$

$$\sum_i \bar{R}_i y_{i,t} \leq M \quad \forall t \qquad \text{(obey mining capacity)}$$

$$\sum_i R_i z_{i,t} \leq P \quad \forall t \qquad \text{(obey processing capacity)}$$

$$0 \leq z_{i,t} \leq y_{i,t} \quad \forall i \forall t \qquad \text{(process not more than mined)}$$

$$\sum_{t' \leq t} y_{i,t'} \geq x_{i,t} \quad \forall i \forall t \qquad \text{(mining turns mined indicator off)}$$

Solve the MIP...

Solve the MIP...

```
Starting Cplex...
Tried aggregator 1 time.
MIP Presolve eliminated 154 rows and 137 columns.
Aggregator did 8 substitutions.
Reduced MIP has 1415 rows, 696 columns, and 6852 nonzeros.
[...]
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Node    ItCnt     Gap

*     0+    0                          0.0000                     729     ---
      0     0        8.0964    84      0.0000        8.0964       729     ---
      0     0        8.0043    87      0.0000      Fract: 6       779     ---
*     0+    0                          6.9018        8.0043       779   15.97%
*     0+    0                          7.0221        8.0043       779   13.99%
*     0+    0                          7.1293        8.0043       779   12.27%
      0     2        8.0043    87      7.1293        8.0042       779   12.27%
*    20+   16                          7.2666        7.9037      1402    8.77%
    100    45        cutoff              7.2666        7.8249      3115    7.68%
    200    76        7.2772    22      7.2666        7.6182      5411    4.84%
    300    83        cutoff              7.2666        7.4420      7467    2.41%
    400    53        cutoff              7.2666        7.3440      8627    1.07%

Gomory fractional cuts applied:   6
MIP status(101): integer optimal solution
```

Time: 1.2 seconds

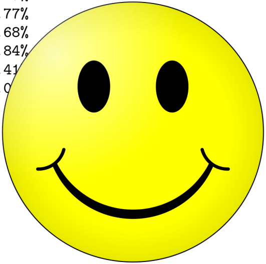
CPLEX 12.1	1.2s	7.27
GUROBI 2.0	1.9s	7.27
SCIP(CPX) 1.2.0	2.3s	7.27
CBC 2.3	78.9s	7.27
GLPK 4.39	140.7s	7.27

Solve the MIP...

```
Starting Cplex...
Tried aggregator 1 time.
MIP Presolve eliminated 154 rows and 137 columns.
Aggregator did 8 substitutions.
Reduced MIP has 1415 rows, 696 columns, and 6852 nonzeros.
[...]
        Nodes                                         Cuts/
   Node  Left     Objective  IInf  Best Integer    Best Node    ItCnt     Gap

*     0+    0                         0.0000                      729     ---
      0     0        8.0964    84     0.0000         8.0964       729     ---
      0     0        8.0043    87     0.0000       Fract: 6       779     ---
*     0+    0                         6.9018         8.0043       779   15.97%
*     0+    0                         7.0221         8.0043       779   13.99%
*     0+    0                         7.1293         8.0043       779   12.27%
      0     2        8.0043    87     7.1293         8.0042       779   12.27%
*    20+   16                         7.2666         7.9037      1402    8.77%
    100    45        cutoff           7.2666         7.8249      3115    7.68%
    200    76        7.2772    22     7.2666         7.6182      5411    4.84%
    300    83        cutoff           7.2666         7.4420      7467    2.41
    400    53        cutoff           7.2666         7.3440      8627    1.0

Gomory fractional cuts applied:  6
MIP status(101): integer optimal solution
```

Time: 1.2 seconds

CPLEX 12.1	1.2s	7.27
GUROBI 2.0	1.9s	7.27
SCIP(CPX) 1.2.0	2.3s	7.27
CBC 2.3	78.9s	7.27
GLPK 4.39	140.7s	7.27



"It work's like a charm..."

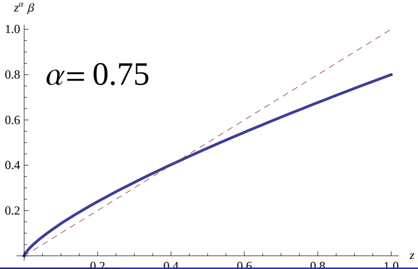
Change the objective a bit...

Consider "Economy of scales"...

$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i \beta z_{i,t}^{\alpha}}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. precedence constraints; mining and processing capacity constraints

Parameter $\alpha > 0$, $\quad \beta = 1/(2-\alpha)$

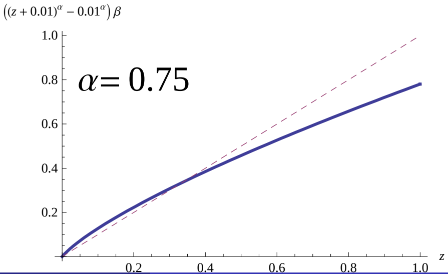


Consider "Economy of scales"...

$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i \beta((z_{i,t}+\varepsilon)^\alpha - \varepsilon^\alpha)}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. precedence constraints; mining and processing capacity constraints

Parameter $\alpha > 0$, $\beta = 1/(2-\alpha)$
$\varepsilon > 0$ for differentiability in $z_{i,t} = 0$



Change the objective a bit...

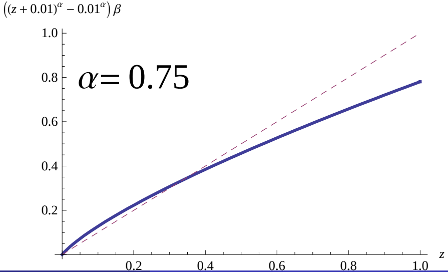
Consider "Economy of scales"...

$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i \beta((z_{i,t}+\varepsilon)^\alpha - \varepsilon^\alpha)}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. precedence constraints; mining and processing capacity constraints

Parameter $\alpha > 0$, $\beta = 1/(2-\alpha)$
$\varepsilon > 0$ for differentiability in $z_{i,t} = 0$

Nonlinear objective!



Solve the MINLP...

Solve the MINLP...

```
Simple B&B       ALFA  20ct09 23.3.0 LNX 12168.13781 LX3 x86/Linux
[...]
 Root node solved locally optimal.
 Resetting optcr to 1.0e-09
   Node  Act. Lev.  Objective  IInf  Best Int.   Best Bound   Gap  (1 secs)
      0     2   0      24.6966 260           -      24.6966        -
      1     3   1      24.6961 256           -      24.6966        -
[...]
   Node  Act. Lev.  Objective  IInf  Best Int.   Best Bound   Gap  (12 secs)
    200   202 200       0.9735   4           -      24.6966        -
    201   203 201       0.9735   3           -      24.6966        -
    202   204 202       0.9735   2           -      24.6966        -
    203   205 203       0.9735   1           -      24.6966        -
*   204   204 204       0.9735   0      0.9735      24.6966  0.960581
[...]
  28269 28176  27      17.1427 161      4.3680      19.3698  0.774496
 Resource limit exceeded
 Statistics:
    Iterations    :          1902560
    NLP Seconds   :      3600.000000
    B&B nodes     :            28270
    MIP solution  :         4.367976 found in node 14747
    Best possible :        19.369812
    Absolute gap  :        15.001836     optca :  0.000000
    Relative gap  :         0.774496     optcr :  0.000000
    Model Status  :                8
    Solver Status :                3
```

Solve the MINLP...

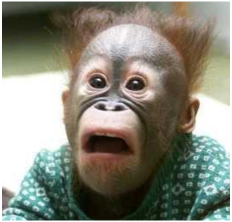
```
Simple B&B       ALFA  20ct09 23.3.0 LNX 12168.13781 LX3 x86/Linux
[...]
 Root node solved locally optimal.
 Resetting optcr to 1.0e-09
   Node  Act. Lev.  Objective  IInf  Best Int.    Best Bound   Gap  (1 secs)
      0     2   0      24.6966 260           -       24.6966        -
      1     3   1      24.6961 256           -       24.6966        -
[...]
   Node  Act. Lev.  Objective  IInf  Best Int.    Best Bound   Gap  (12 secs)
    200   202 200       0.9735   4           -       24.6966        -
    201   203 201       0.9735   3           -       24.6966        -
    202   204 202       0.9735   2           -       24.6966        -
    203   205 203       0.9735   1           -       24.6966        -
*   204   204 204       0.9735   0      0.9735       24.6966  0.960581
[...]
  28269 28176  27      17.1427 161      4.3680       19.3698  0.774496
 Resource limit exceeded
 Statistics:
    Iterations    :          1902560
    NLP Seconds   :      3600.000000
    B&B nodes     :            28270
    MIP solution  :         4.367976 found in node 14747
    Best possible :        19.369812
    Absolute gap  :        15.001836     optca :  0.000000
    Relative gap  :         0.774496     optcr :  0.000000
    Model Status  :                8
    Solver Status :                3
```

after 1 hour:

solver	primal	dual	gap
SBB	4.37	19.37	77.4%
AlphaECP	11.0	–	–
DICOPT	10.18	–	–
BONMIN-BB	11.21	**16.73**	33.0%
BONMIN-OA	11.44	24.70	53.7%
BONMIN-QG	11.45	24.15	52.6%
BONMIN-Hyb	10.86	24.00	54.7%
SCIP “1.3”	**12.42**	16.92	**26.6%**
BARON	11.72	18.96	38.2%
LINDOGlobal	10.15	19.17	47%
COUENNE	8.89	20.99	58%

Solve the MINLP...

```
Simple B&B       ALFA   2Oct09 23.3.0 LNX 12168.13781 LX3 x86/Linux
[...]
 Root node solved locally optimal.
 Resetting optcr to 1.0e-09
   Node  Act. Lev.  Objective  IInf  Best Int.    Best Bound   Gap  (1 secs)
      0     2   0     24.6966 260            -      24.6966           -
      1     3   1     24.6961 256            -      24.6966           -
[...]
   Node  Act. Lev.  Objective  IInf  Best Int.    Best Bound   Gap  (12 secs)
    200   202 200      0.9735   4            -      24.6966           -
    201   203 201      0.9735   3            -      24.6966           -
    202   204 202      0.9735   2            -      24.6966           -
    203   205 203      0.9735   1            -      24.6966           -
*   204   204 204      0.9735   0       0.9735      24.6966  0.960581
[...]
  28269 28176  27     17.1427 161       4.3680      19.3698  0.774496
 Resource limit exceeded
 Statistics:
    Iterations     :          1902560
    NLP Seconds    :      3600.000000
    B&B nodes      :            28270
    MIP solution   :         4.367976 found in node 14747
    Best possible  :        19.369812
    Absolute gap   :        15.001836     optca :   0.000000
    Relative gap   :         0.774496     optcr :   0.000000
    Model Status   :                8
    Solver Status  :                3
```



after 1 hour:

solver	primal	dual	gap
SBB	4.37	19.37	77.4%
AlphaECP	11.0	–	–
DICOPT	10.18	–	–
BONMIN-BB	11.21	**16.73**	33.0%
BONMIN-OA	11.44	24.70	53.7%
BONMIN-QG	11.45	24.15	52.6%
BONMIN-Hyb	10.86	24.00	54.7%
SCIP "1.3"	**12.42**	16.92	**26.6%**
BARON	11.72	18.96	38.2%
LINDOGlobal	10.15	19.17	47%
COUENNE	8.89	20.99	58%

MINLP is slooooooow!

What does SBB do?

MINLP: Let $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$

$$
\begin{aligned}
\min\ & f(x) \\
\text{s.t. } & g(x) \leq 0 \\
& x \in [x^L, x^U] \\
& x_i \in \mathbb{Z}, \quad i \in I
\end{aligned}
$$

What does SBB do?

MINLP: Let $f : \mathbb{R}^n \to \mathbb{R}$, $g : \mathbb{R}^n \to \mathbb{R}^m$

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g(x) \le 0 \\ & x \in [x^L, x^U] \\ & x_i \in \mathbb{Z}, \quad i \in I \end{aligned}$$

NLP based Branch–and–Bound:

- Dual Bounds: Solve NLP relaxation

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g(x) \le 0, \\ & x \in [x^L, x^U], \qquad x_i \in [\ell_i, u_i], \quad i \in I \end{aligned}$$

- Primal Bounds: integer feasible solution of NLP relaxation
- Branching: On integer variables with fractional value in NLP relax.

Solvers that implement NLP based Branch–and–Bound

SBB = Simple Branch and Bound

[ARKI Consulting & Development A/S and GAMS Inc., 2002]

- developed by A. Drud (ARKI Consulting)
- NLP-based Branch-and-Bound
- available with GAMS

Solvers that implement NLP based Branch–and–Bound

SBB = Simple Branch and Bound

[ARKI Consulting & Development A/S and GAMS Inc., 2002]

- developed by A. Drud (ARKI Consulting)
- NLP-based Branch-and-Bound
- available with GAMS

Bonmin = Basic Open-source Nonlinear Mixed Integer prog.

[Bonami et al., 2008]

- COIN-OR solver developed by P. Bonami, A. Wächter, et.al. (CMU, IBM, Bologna)
- based on CBC and NLP solvers IPOPT and FilterSQP
- option B-BB = NLP–based Branch–and–Bound

Solvers that implement NLP based Branch–and–Bound

SBB = Simple Branch and Bound

[ARKI Consulting & Development A/S and GAMS Inc., 2002]

- developed by A. Drud (ARKI Consulting)
- NLP-based Branch-and-Bound
- available with GAMS

Bonmin = Basic Open-source Nonlinear Mixed Integer prog.

[Bonami et al., 2008]

- COIN-OR solver developed by P. Bonami, A. Wächter, et.al. (CMU, IBM, Bologna)
- based on CBC and NLP solvers IPOPT and FilterSQP
- option B-BB = NLP–based Branch–and–Bound

MINLPBB = Mixed Integer NonLinear Programming Branch and Bound

[Leyffer, 2001]

- developed by S. Leyffer
- based on FilterSQP and bqpd

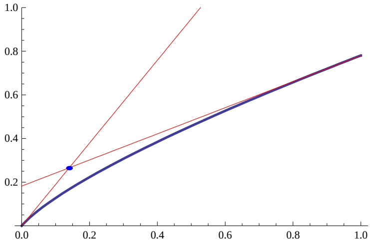
Solve the MINLP again...

Solve the MINLP again...

```
List of user-set options:
                                    Name   Value                used
                        bonmin.algorithm = B-OA                  yes

******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open source code under the Common Public License (CPL).
         For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

NLP0012I
              Num      Status      Obj             It       time
NLP0013I     1        OPT        -24.69658700869476      31       0.420026
OA0003I New best feasible of -11.3798 found after 93.0778 sec.
OA0010I After   184.4 seconds, upper bound   -11.3797, lower bound   -24.0727
OA0003I New best feasible of -11.442 found after 184.492 sec.
OA0010I After   321.3 seconds, upper bound   -11.4418, lower bound   -24.0708
OA0010I After   565.1 seconds, upper bound   -11.4418, lower bound   -24.0665
OA0010I After   830.0 seconds, upper bound   -11.4418, lower bound   -24.0318
OA0010I After  1064.1 seconds, upper bound   -11.4418, lower bound   -24.0262
OA0010I After  1804.6 seconds, upper bound   -11.4418, lower bound   -23.9881
OA0010I After  2983.6 seconds, upper bound   -11.4418, lower bound    -23.948
OA0010I After  3600.1 seconds, upper bound   -11.4418, lower bound   -23.9747
Time limit exceeded.

Bonmin finished. Found feasible point. Objective function = -11.441962.

MINLP solution:         11.44196163   (0 nodes, 3600.37 seconds)
Best possible:          24.69658706
Absolute gap:                13.255
Relative gap:                0.5367
```

What does Bonmin (B-OA) do?

$$\begin{aligned} &\min\ f(x) \\ &\text{s.t. } g(x) \leq 0 \\ &\quad x \in [x^L, x^U] \\ &\quad x_i \in \mathbb{Z}, \quad i \in I \end{aligned}$$



Outer Approximation Algorithm:

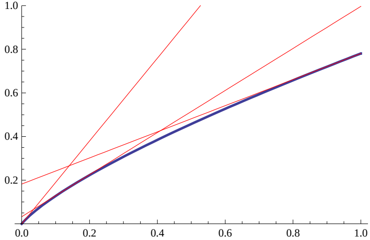
1. Solve MIP relaxation

$$\min z \quad \text{s.t. } f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq z, \quad \hat{x} \in S,$$
$$g(\hat{x}) + \nabla g(\hat{x})(x - \hat{x}) \leq 0, \quad \hat{x} \in S,$$
$$x^L \leq x \leq x^U, \qquad x_i \in \mathbb{Z}, i \in I$$

[Duran and Grossmann, 1986, Fletcher and Leyffer, 1994]

What does Bonmin (B-OA) do?

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g(x) \le 0 \\ & x \in [x^L, x^U] \\ & x_i \in \mathbb{Z}, \quad i \in I \end{aligned}$$



Outer Approximation Algorithm:

1. Solve MIP relaxation

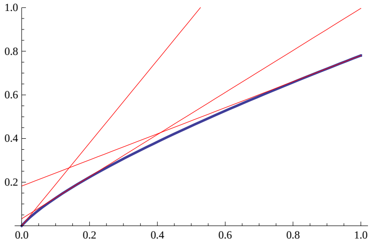
$$\begin{aligned} \min z \quad \text{s.t. } & f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \le z, \quad \hat{x} \in S, \\ & g(\hat{x}) + \nabla g(\hat{x})(x - \hat{x}) \le 0, \quad \hat{x} \in S, \\ & x^L \le x \le x^U, \qquad x_i \in \mathbb{Z}, i \in I \end{aligned}$$

2. If solution x^* satisfies $f(x^*) = z$, $g(x^*) \le 0$, then finish.
3. Otherwise, add x^* to $S \Rightarrow$ cut off x^* from relaxation and repeat.

[Duran and Grossmann, 1986, Fletcher and Leyffer, 1994]

What does Bonmin (B-OA) do?

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g(x) \leq 0 \\ & x \in [x^L, x^U] \\ & x_i \in \mathbb{Z}, \quad i \in I \end{aligned}$$



Outer Approximation Algorithm:

1. Solve MIP relaxation

$$\min z \quad \text{s.t. } f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq z, \quad \hat{x} \in S,$$
$$g(\hat{x}) + \nabla g(\hat{x})(x - \hat{x}) \leq 0, \quad \hat{x} \in S,$$
$$x^L \leq x \leq x^U, \qquad x_i \in \mathbb{Z}, i \in I$$

2. If solution x^* satisfies $f(x^*) = z$, $g(x^*) \leq 0$, then finish.
3. Otherwise, add x^* to $S \Rightarrow$ cut off x^* from relaxation and repeat.

Primal Solutions: Solve NLP

$\min\{f(x) : g(x) \leq 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\ i \in I\}$ and add solution to S.

[Duran and Grossmann, 1986, Fletcher and Leyffer, 1994]

Solvers that implement Outer Approximation Algorithm

DICOPT = Discrete and Continuous Optimizer

[Duran and Grossmann, 1986]

- developed by J. Viswanathan and I.E. Grossmann (CMU)
- pure GAMS solver

Solvers that implement Outer Approximation Algorithm

DICOPT = Discrete and Continuous Optimizer

[Duran and Grossmann, 1986]

- developed by J. Viswanathan and I.E. Grossmann (CMU)
- pure GAMS solver

AlphaECP = α Extended Cutting Plane

[Westerlund and Pörn, 2002]

- developed by T. Westerlund and T. Lastusilta at Åbo Akademi University, Finland

Solvers that implement Outer Approximation Algorithm

DICOPT = Discrete and Continuous Optimizer

[Duran and Grossmann, 1986]

- developed by J. Viswanathan and I.E. Grossmann (CMU)
- pure GAMS solver

AlphaECP = α Extended Cutting Plane

[Westerlund and Pörn, 2002]

- developed by T. Westerlund and T. Lastusilta at Åbo Akademi University, Finland

Bonmin = Basic Open-source Nonlinear Mixed Integer programming

[Bonami et al., 2008]

- COIN-OR solver developed by P. Bonami, A. Wächter et.al.
- option B-OA = Outer Approximation Algorithm

Solve the MINLP a third time...

Solve the MINLP a third time...

```
SCIP version 1.1.0.11 [precision: 8 byte] [memory: block] [mode: optimized] [LP solver: CPLEX 11.21]
Copyright (c) 2002-2009 Konrad-Zuse-Zentrum fuer Informationstechnik Berlin (ZIB)

Original problem has 1576 linear constraints and 1 nonlinear constraints.
original problem has 841 variables (280 bin, 0 int, 0 impl, 561 cont) and 1577 constraints
presolving:
(round 1) 7 del vars, 8 del conss, 0 chg bounds, 0 chg sides, 0 chg coeffs, 1 upgd conss, 1916 impls, 0 clqs
[...]
presolving (8 rounds):
 28 deleted vars, 52 deleted constraints, 2 tightened bounds, 0 added holes, 0 changed sides, 0 changed coeffic
 14761 implications, 0 cliques
presolved problem has 1093 variables (263 bin, 0 int, 0 impl, 830 cont) and 1805 constraints
    115 constraints of type <varbound>
   1182 constraints of type <linear>
    229 constraints of type <logicor>
    279 constraints of type <nonlinear>
Presolving Time: 1.28
Initializing expression interpreter CppAD 20090303.0
Oracle initializes expression interpreter CppAD 20090303.0

 time | node  | left  |LP iter|frac |cols |rows |cuts |confs|strbr|  dualbound   | primalbound  |  gap
  1.4s|     1 |     0 |  1369 | 261 |1930 |3758 |   0 |   0 |   0 | 3.520238e+01 |      --      |   Inf
  1.9s|     1 |     0 |  2732 | 261 |1930 |3972 | 214 |   0 |   0 | 2.952667e+01 |      --      |   Inf
[...]
X12.6s|     1 |     0 |  4492 | 242 |1930 |5515 |1757 |   0 |   0 | 2.461582e+01 |-7.578685e+02 |   Inf
[...]
 3600s| 86900 | 10242 | 15252k|   - |1930 |4748 |6663k|   3 |6969 | 1.691782e+01 | 1.241630e+01 |  26.61%
[...]
Constraints        :     Number  #Separate #Propagate    #EnfoLP    #EnfoPS    DomReds       Cuts   Children
  integral         :          0          0          0      52468          0       1351          0     102882
  nonlinear        :        279      90608      52890        243          1     292358    6663347          0
```

What does SCIP do?

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0,\, x \in [x^L, x^U],\, x_i \in \mathbb{Z}, \quad i \in I$$

Integrate Outer Approximation into LP based Branch–and–Bound:

- Dual Bounds: Solve LP relaxation

$$\begin{aligned} \min z \quad \text{s.t. } & f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq z, \quad \hat{x} \in S, \\ & g(\hat{x}) + \nabla g(\hat{x})(x - \hat{x}) \leq 0, \quad \hat{x} \in S, \\ & x \in [x^L, x^U], \qquad x_i \in [\ell_i, u_i], \quad i \in I \end{aligned}$$

If solution x^* violates $g(x^*) \leq 0$, $f(x^*) = z$, add x^* to S.

[Quesada and Grossmann, 1992]

What does SCIP do?

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0,\ x \in [x^L, x^U],\ x_i \in \mathbb{Z}, \quad i \in I$$

Integrate Outer Approximation into LP based Branch–and–Bound:

- Dual Bounds: Solve LP relaxation

$$\begin{aligned} \min z \quad \text{s.t. } & f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq z, \quad \hat{x} \in S, \\ & g(\hat{x}) + \nabla g(\hat{x})(x - \hat{x}) \leq 0, \quad \hat{x} \in S, \\ & x \in [x^L, x^U], \qquad x_i \in [\ell_i, u_i], \quad i \in I \end{aligned}$$

  If solution x^* violates $g(x^*) \leq 0$, $f(x^*) = z$, add x^* to S.

- Primal Bounds: Round x_i^*, $i \in I$, and solve NLP

$$\begin{aligned} & \min f(x) \\ & \text{s.t. } g(x) \leq 0, \\ & \quad x \in [x^L, x^U], \qquad x_i = x_i^*, \quad i \in I \end{aligned}$$

[Quesada and Grossmann, 1992]

What does SCIP do?

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0,\ x \in [x^L, x^U],\ x_i \in \mathbb{Z}, \quad i \in I$$

Integrate Outer Approximation into LP based Branch–and–Bound:

- Dual Bounds: Solve LP relaxation

$$\begin{aligned} \min z \quad \text{s.t. } & f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq z, \quad \hat{x} \in S, \\ & g(\hat{x}) + \nabla g(\hat{x})(x - \hat{x}) \leq 0, \quad \hat{x} \in S, \\ & x \in [x^L, x^U], \qquad x_i \in [\ell_i, u_i], \quad i \in I \end{aligned}$$

  If solution x^* violates $g(x^*) \leq 0$, $f(x^*) = z$, add x^* to S.
- Primal Bounds: Round x_i^*, $i \in I$, and solve NLP

$$\begin{aligned} \min\ & f(x) \\ \text{s.t. } & g(x) \leq 0, \\ & x \in [x^L, x^U], \qquad x_i = x_i^*, \quad i \in I \end{aligned}$$

- Branching: On integer variables with fractional value in LP relax.

[Quesada and Grossmann, 1992]

Solvers that implement LP-based Branch–and–Bound

FilMINT = Filter-Mixed INTeger optimizer

[Abhishek et al., 2006]

- developed by K. Abhishek, J.T. Linderoth (Lehigh), and S. Leyffer (Argonne)
- combines MIP solver MINTO (Nemhauser et.al.) with NLP solver filterSQP (Fletcher)


Solvers that implement LP-based Branch–and–Bound

FilMINT = Filter-Mixed INTeger optimizer

[Abhishek et al., 2006]

- developed by K. Abhishek, J.T. Linderoth (Lehigh), and S. Leyffer (Argonne)
- combines MIP solver MINTO (Nemhauser et.al.) with NLP solver filterSQP (Fletcher)

SCIP = Solving Constraint Integer Programs

[Achterberg, 2007, Berthold et al., 2009]

- SCIP 1.2.0: support for quadratic MINLP
- general MINLP experimental

Solvers that implement LP-based Branch–and–Bound

FilMINT = Filter-Mixed INTeger optimizer

[Abhishek et al., 2006]

- developed by K. Abhishek, J.T. Linderoth (Lehigh), and S. Leyffer (Argonne)
- combines MIP solver MINTO (Nemhauser et.al.) with NLP solver filterSQP (Fletcher)

SCIP = Solving Constraint Integer Programs

[Achterberg, 2007, Berthold et al., 2009]

- SCIP 1.2.0: support for quadratic MINLP
- general MINLP experimental

Bonmin = Basic Open-source Nonlinear Mixed Integer programming

- COIN-OR solver developed by P. Bonami, A. Wächter et.al.
- option B-QG = Quesada-Grossmann LP–based B&B
- option B-Hyb = solve NLP (instead of LP) relaxation at some nodes

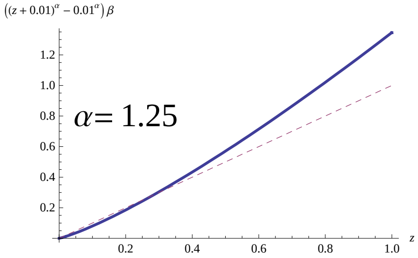
Let's try $\alpha = 1.25$...

$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i \beta((z_{i,t}+\varepsilon)^\alpha - \varepsilon^\alpha)}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. (x, y) satisfy precedence constraints

obey mining and processing capacities

$\dots$



$\alpha = 1.25$: Try SBB again

$\alpha = 1.25$: Try SBB again – Ups...

```
Simple B&B       ALFA  20ct09 23.3.0 LNX 12168.13781 LX3 x86/Linux

 841 columns (280 discrete), 1577 rows, 7951 nonzeros
C O N O P T 3    ALFA  20ct09 23.3.0 LNX 13062.13781 LX3 x86/Linux


    C O N O P T 3   version 3.14T
    Copyright (C)   ARKI Consulting and Development A/S
                    Bagsvaerdvej 246 A
                    DK-2880 Bagsvaerd, Denmark

 Using default options.

 Reading Data

   Iter Phase Ninf   Infeasibility   RGmax    NSB   Step InItr MX OK
      0   0        4.4408920985E-16 (Input point)
                               Pre-triangular equations:        0
                               Post-triangular equations:       1
      1   0        0.0000000000E+00 (After pre-processing)
      2   0        0.0000000000E+00 (After scaling)

 ** Feasible solution. Value of objective =    4.440892098501E-16

   Iter Phase Ninf     Objective     RGmax    NSB   Step InItr MX OK
      4   3        4.4408920985E-16 0.0E+00     0

 ** Optimal solution. There are no superbasic variables.

 Root node solved locally optimal.
 Relaxed problem gives integer solution.
 Terminating.
```

$\alpha = 1.25$: Multistart for root node NLP

$\alpha = 1.25$: Multistart for root node NLP

```
Simple B&B       ALFA  20ct09 23.3.0 LNX 12168.13781 LX3 x86/Linux

Reading parameter(s) from "/nfs/optimi/kombadon/bzfviger/talk_coatwork/sbb.opt"
>>  rootsolver oqnlp.1
Finished reading from "/nfs/optimi/kombadon/bzfviger/talk_coatwork/sbb.opt"
Root node solver: oqnlp.1
 841 columns (280 discrete), 1577 rows, 7951 nonzeros
OQNLP MStart     ALFA  20ct09 23.3.0 LNX 13463.13781 LX3 x86/Linux
Reading parameter(s) from "/nfs/optimi/kombadon/bzfviger/talk_coatwork/oqnlp.opt"
>>  nlpsolver conopt
>>  stage1_iterations 10
>>  max_solver_calls 10
Finished reading from "/nfs/optimi/kombadon/bzfviger/talk_coatwork/oqnlp.opt"
[...]
   Itn     Penval   Merit     Merit     Dist     Best       Solver     Term     Sinf
                    Filter Threshold  Filter    Obj          Obj      Code
     0   +1.000e+30      -1.000e+30      -0.000e+00  -0.000e+00 KTC  +0.000e+00
    80   +3.289e+04 REJ  +2.748e+04  ACC +6.751e+00
   100   +3.683e+04 REJ  +2.525e+04  ACC +2.587e+01
   158   +9.643e+03 ACC  +2.635e+04  ACC +2.962e+01  +2.862e+01 KTC  +1.104e-13
[...]
--- Exiting OQNLP
 Root node solved locally optimal.
 Resetting optcr to 1.0e-09
   Node  Act. Lev.  Objective  IInf  Best Int.    Best Bound   Gap  (1 secs)
      0     2   0      29.6202  19            -      29.6202          -
[...]
     11    13  11      29.7968   1            -      29.7968          -
*    12    11  12      29.7968   0      29.7968      29.7968  0.000000
 Integer solution
 Terminating.
```

$\alpha = 1.25$: Results for all solvers

solver	time	primal	dual	gap
SBB	0s	0	0	0%
SBB+MS	1s	29.8	29.8	0%
AlphaECP	1s	10.3	–	–
DICOPT	1s	0	–	–
BONMIN-BB	3600s	–	26.3	∞
BONMIN-OA	7s	21.1	21.1	0%
BONMIN-QG	2s	12.6	12.6	0%
BONMIN-Hyb	9s	19.3	19.3	0%
SCIP "1.3"	3600s	31.8	33.3	4.4%
BARON	3600s	**32.9**	33.3	1.2%
LINDOGlobal	3600s	31.3	34.0	7.7%
COUENNE	3600s	32.0	34.2	6.6%

$\alpha = 1.25$: Results for all solvers

solver	time	primal	dual	gap
SBB	0s	0	0	0%
SBB+MS	1s	29.8	29.8	0%
AlphaECP	1s	10.3	–	–
DICOPT	1s	0	–	–
BONMIN-BB	3600s	–	26.3	∞
BONMIN-OA	7s	21.1	21.1	0%
BONMIN-QG	2s	12.6	12.6	0%
BONMIN-Hyb	9s	19.3	19.3	0%
SCIP "1.3"	3600s	31.8	33.3	4.4%
BARON	3600s	**32.9**	33.3	1.2%
LINDOGlobal	3600s	31.3	34.0	7.7%
COUENNE	3600s	32.0	34.2	6.6%

Wow - MINLP can be fast! ;-)

Why the different results?

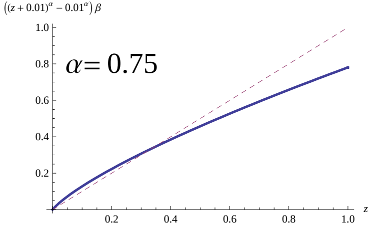
Recall ...

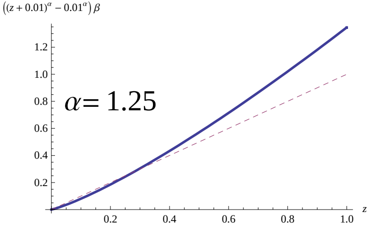
$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i((z_{i,t}+\varepsilon)^\alpha - \varepsilon^\alpha)}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. (x, y) satisfy precedence constraints; obey mining and processing capacities

$\ldots$

All we did was changing α a bit...





Why the different results?

Recall ...

$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i((z_{i,t}+\varepsilon)^\alpha - \varepsilon^\alpha)}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. (x, y) satisfy precedence constraints; obey mining and processing capacities

$\ldots$

$\alpha = 1.25 \Rightarrow$ Objective is convex $\oplus$ Problem is maximization

NONCONVEX MINLP

- NLP relaxation may have several local (non-global) minima
- linearizations $f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq 0$ etc. are not "outer-approximating"

Why the different results?

Recall ...

$$\max \sum_{t=1}^{T} \left(\frac{1}{1+q}\right)^{t-1} \left(\sum_i \underbrace{c_i((z_{i,t}+\varepsilon)^{\alpha} - \varepsilon^{\alpha})}_{\text{income by sale}} - \underbrace{p_i z_{i,t}}_{\text{processing cost}} - \underbrace{m_i y_{i,t}}_{\text{mining cost}} \right)$$

s.t. (x, y) satisfy precedence constraints; obey mining and processing capacities

$\dots$

$\alpha = 1.25 \Rightarrow$ Objective is convex $\oplus$ Problem is maximization

NONCONVEX MINLP

- NLP relaxation may have several local (non-global) minima
- linearizations $f(\hat{x}) + \nabla f(\hat{x})(x - \hat{x}) \leq 0$ etc. are not "outer-approximating"

PANIC NOW!

The table again...

solver	time	primal	dual	gap
SBB	0s	0	0	0%
SBB+MS	1s	29.8	29.8	0%
AlphaECP	1s	10.3	–	–
DICOPT	1s	0	–	–
BONMIN-BB	3600s	–	26.3	∞
BONMIN-OA	7s	21.1	21.1	0%
BONMIN-QG	2s	12.6	12.6	0%
BONMIN-Hyb	9s	19.3	19.3	0%
SCIP "1.3"	3600s	31.8	33.3	4.4%
BARON	3600s	**32.9**	33.3	1.2%
LINDOGlobal	3600s	31.3	34.0	7.7%
COUENNE	3600s	32.0	34.2	6.6%

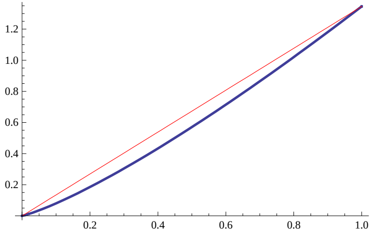
What does BARON do?

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0,\, x \in [x^L, x^U],\, x_i \in \mathbb{Z}, \quad i \in I$$

LP based Branch–and–Bound:

- Dual Bounds: Solve a LP relaxation

$$\begin{aligned} &\min z \\ &\text{s.t. } Ax + A'z \leq b, \\ &\quad x \in [x^L, x^U], \\ &\quad x_i \in [\ell_i, u_i], \quad i \in I \end{aligned}$$



[Tawarmalani and Sahinidis, 2002, 2004]

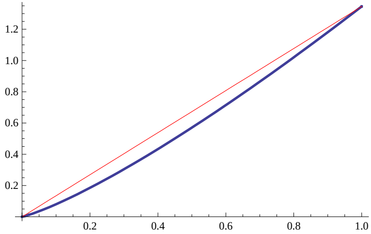
What does BARON do?

$$\min f(x) \quad \text{s.t.} \quad g(x) \le 0,\, x \in [x^L, x^U],\, x_i \in \mathbb{Z}, \quad i \in I$$

LP based Branch–and–Bound:

- Dual Bounds: Solve a LP relaxation

$$\begin{aligned} &\min z \\ &\text{s.t. } Ax + A'z \le b, \\ &\quad x \in [x^L, x^U], \\ &\quad x_i \in [\ell_i, u_i], \quad i \in I \end{aligned}$$



- Primal Bounds: Round x_i^*, $i \in I$, and solve (locally) the NLP

$$\min\ f(x) \quad \text{s.t.} \quad g(x) \le 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\, i \in I$$

[Tawarmalani and Sahinidis, 2002, 2004]

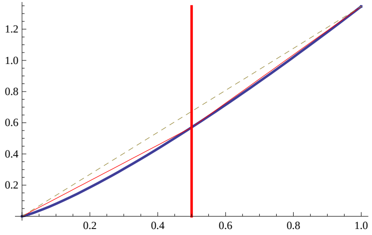
What does BARON do?

$$\min f(x) \quad \text{s.t.} \quad g(x) \leq 0,\, x \in [x^L, x^U],\, x_i \in \mathbb{Z}, \quad i \in I$$

LP based Branch–and–Bound:

- Dual Bounds: Solve a LP relaxation

$$\begin{aligned} &\min z \\ &\text{s.t. } Ax + A'z \leq b, \\ &\quad x \in [x^L, x^U], \\ &\quad x_i \in [\ell_i, u_i], \quad i \in I \end{aligned}$$



- Primal Bounds: Round x_i^*, $i \in I$, and solve (locally) the NLP

$$\min\ f(x) \quad \text{s.t.} \quad g(x) \leq 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\, i \in I$$

- Branching: On integer variables with fractional value in LP relax. And on continuous variables in nonconvex terms.

[Tawarmalani and Sahinidis, 2002, 2004]

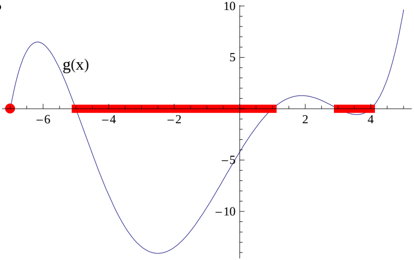
The linear outer–approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$

Seek: A (linear) outer-approximation of

$$\{x \in [x^L, x^U] : g(x) \leq 0\}$$



The linear outer–approximation of a nonconvex MINLP

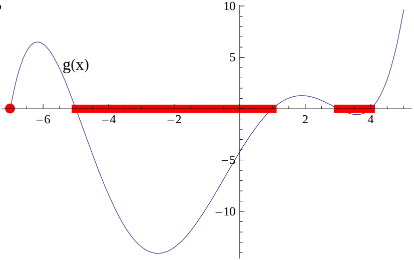
Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$

Seek: A (linear) outer-approximation of

$$\{x \in [x^L, x^U] : g(x) \leq 0\}$$

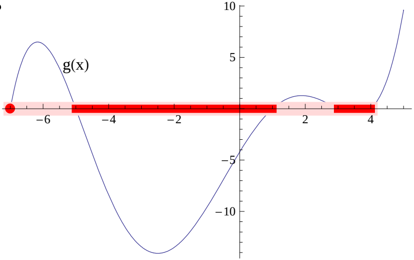
$\Rightarrow$ too difficult in general



The linear outer–approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$



Seek: A (linear) outer-approximation of

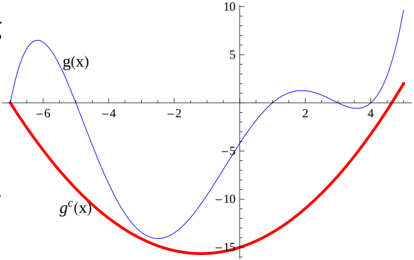
$$\operatorname{conv}\{x \in [x^L, x^U] : g(x) \leq 0\}$$

$\Rightarrow$ too difficult in general

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$

Seek: A convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[x^L, x^U]$:



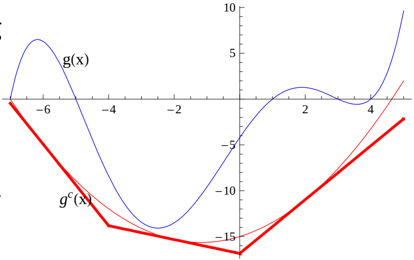
$$\{x \in [x^L, x^U] : g(x) \leq 0\} \subseteq \{x \in [x^L, x^U] : g^c(x) \leq 0\}$$

The linear outer–approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$

Seek: A convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[x^L, x^U]$:



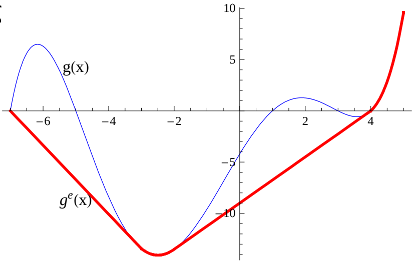
$$\{x \in [x^L, x^U] : g(x) \leq 0\} \subseteq \{x \in [x^L, x^U] : g^c(x) \leq 0\}$$
$$\subseteq \{x \in [x^L, x^U] : g^c(\hat{x}) + \nabla g^c(\hat{x})(x - \hat{x}) \leq 0,\ \hat{x} \in S\}$$

The linear outer–approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$

Seek: The tightest convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[x^L, x^U]$:



The convex envelope $g^e : [x^L, x^U] \to \mathbb{R}$ is a function such that

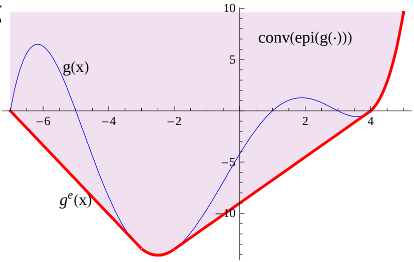
- analytically:
  - $g^e(x)$ is convex on $[x^L, x^U]$
  - $g^e(x) \leq g(x)$ for all $x \in [x^L, x^U]$
  - $\forall h : [x^L, x^U] \to \mathbb{R}$, $h(\cdot)$ convex, $h(\cdot) \leq g(\cdot)$: $h(\cdot) \leq g^e(\cdot)$

The linear outer–approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$

Seek: The tightest convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[x^L, x^U]$:



The convex envelope $g^e : [x^L, x^U] \to \mathbb{R}$ is a function such that

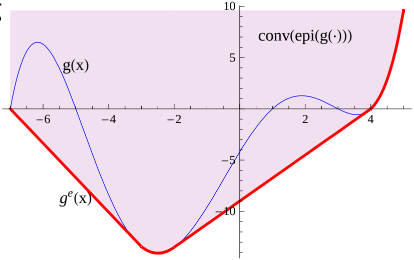
- analytically:
  - $g^e(x)$ is convex on $[x^L, x^U]$
  - $g^e(x) \leq g(x)$ for all $x \in [x^L, x^U]$
  - $\forall h : [x^L, x^U] \to \mathbb{R}$, $h(\cdot)$ convex, $h(\cdot) \leq g(\cdot)$: $h(\cdot) \leq g^e(\cdot)$
- geometrically: $\mathrm{epi}(g^e(\cdot)) = \mathrm{conv}(\mathrm{epi}(g(\cdot)))$

The linear outer–approximation of a nonconvex MINLP

Given: A function $g : \mathbb{R}^n \to \mathbb{R}$ defining the constraint

$$g(x) \leq 0, \qquad x \in [x^L, x^U]$$

Seek: The tightest convex function $g^c(\cdot)$ that underestimates $g(\cdot)$ on $[x^L, x^U]$:



The convex envelope $g^e : [x^L, x^U] \to \mathbb{R}$ is a function such that

- analytically:
  - $g^e(x)$ is convex on $[x^L, x^U]$
  - $g^e(x) \leq g(x)$ for all $x \in [x^L, x^U]$
  - $\forall h : [x^L, x^U] \to \mathbb{R}$, $h(\cdot)$ convex, $h(\cdot) \leq g(\cdot)$: $h(\cdot) \leq g^e(\cdot)$
- geometrically: $\text{epi}(g^e(\cdot)) = \text{conv}(\text{epi}(g(\cdot)))$

In general: Convex envelopes are hard to find.

(finding the convex envelope of a function is as hard as finding its global min.)

Convex envelopes for concave functions

Convex envelopes for concave functions

Theorem 1 (of 1): Let $v_1, \ldots, v_k$ be the vertices of a polytope P. The convex envelope $g^e(x)$ of a concave function $g(x)$ over P can be expressed as

$$g^e(x) = \min_{\alpha} \sum_{i=1}^{k} \alpha_i g(v_i)$$
$$\text{s.t. } x = \sum_{i=1}^{k} \alpha_i v_i, \qquad \sum_{i=1}^{k} \alpha_i = 1, \qquad \alpha_i \geq 0, \quad i = 1, \ldots, k$$

[Tawarmalani and Sahinidis, 2002]

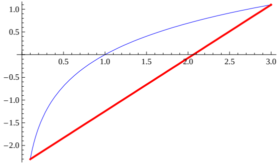
Convex envelopes for concave functions

Theorem 1 (of 1): Let $v_1, \dots, v_k$ be the vertices of a polytope P. The convex envelope $g^e(x)$ of a concave function $g(x)$ over P can be expressed as

$$g^e(x) = \min_{\alpha} \sum_{i=1}^{k} \alpha_i g(v_i)$$

$$\text{s.t.}\, x = \sum_{i=1}^{k} \alpha_i v_i, \qquad \sum_{i=1}^{k} \alpha_i = 1, \qquad \alpha_i \geq 0, \quad i = 1, \dots, k$$

- For now: Special case $n = 1$, $P = [x^L, x^U]$
- Convex envelope for univariate concave functions is the secant between x^L and x^U.
  $\Rightarrow \log(x)$, $-\exp(x)$, $-x^{2k}$, $\sqrt{x}$, ...



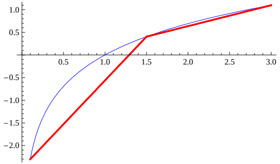
[Tawarmalani and Sahinidis, 2002]

Theorem 1 (of 1): Let $v_1, \dots, v_k$ be the vertices of a polytope P. The convex envelope $g^e(x)$ of a concave function $g(x)$ over P can be expressed as

$$g^e(x) = \min_{\alpha} \sum_{i=1}^{k} \alpha_i g(v_i)$$

$$\text{s.t.}\, x = \sum_{i=1}^{k} \alpha_i v_i, \qquad \sum_{i=1}^{k} \alpha_i = 1, \qquad \alpha_i \geq 0, \quad i = 1, \dots, k$$

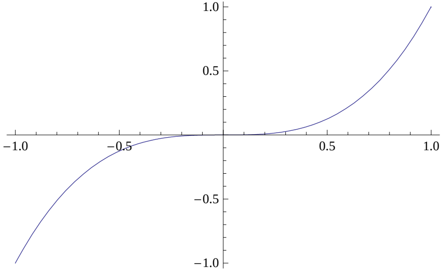
- For now: Special case $n = 1$, $P = [x^L, x^U]$
- Convex envelope for univariate concave functions is the secant between x^L and x^U.
  $\Rightarrow \log(x)$, $-\exp(x)$, $-x^{2k}$, $\sqrt{x}$, ...
- In Branch–and–Bound: tighten by branching



[Tawarmalani and Sahinidis, 2002]

Convex envelope for monomials of odd degree

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$



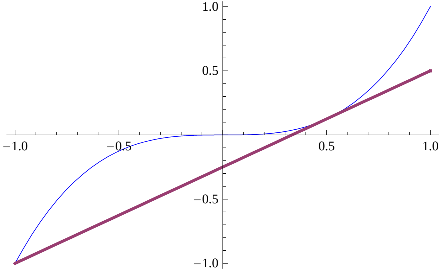
[Liberti and Pantelides, 2003]

Convex envelope for monomials of odd degree

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$

- Seek for secant from x^L to c, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - (x^L)^{2k+1}}{c - x^L}$$



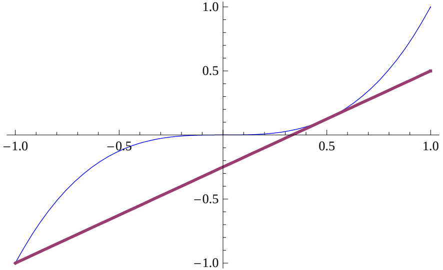
[Liberti and Pantelides, 2003]

Convex envelope for monomials of odd degree

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$

- Seek for secant from x^L to c, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - (x^L)^{2k+1}}{c - x^L}$$



$\Rightarrow$ c is the positive real root of the polynomial

$$2kx^{2k+1} - x^L(2k+1)x^{2k} + (x^L)^{2k+1}$$

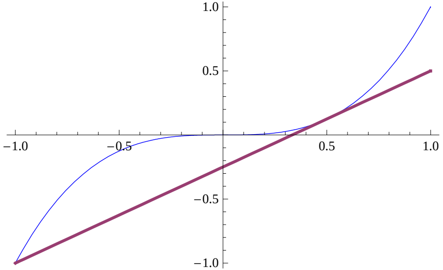
[Liberti and Pantelides, 2003]

Convex envelope for monomials of odd degree

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$

- Seek for secant from x^L to c, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - (x^L)^{2k+1}}{c - x^L}$$



$\Rightarrow$ c is the positive real root of the polynomial

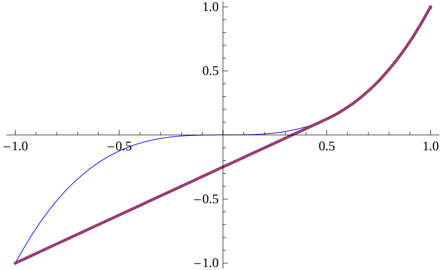
$$2kx^{2k+1} - x^L(2k+1)x^{2k} + (x^L)^{2k+1}$$

k	c (for $x^L = -1$)
1	0.5000000000
2	0.6058295862
3	0.6703320476
4	0.7145377272
5	0.7470540749
6	0.7721416355
7	0.7921778546

[Liberti and Pantelides, 2003]

Convex envelope for monomials of odd degree

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$



- Seek for secant from x^L to c, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - (x^L)^{2k+1}}{c - x^L}$$

$\Rightarrow$ c is the positive real root of the polynomial

$$2kx^{2k+1} - x^L(2k+1)x^{2k} + (x^L)^{2k+1}$$

- obtain convex envelope

k	c (for $x^L = -1$)
1	0.5000000000
2	0.6058295862
3	0.6703320476
4	0.7145377272
5	0.7470540749
6	0.7721416355
7	0.7921778546

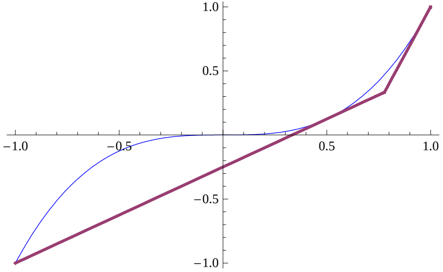
[Liberti and Pantelides, 2003]

Convex envelope for monomials of odd degree

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$

- Seek for secant from x^L to c, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - (x^L)^{2k+1}}{c - x^L}$$



$\Rightarrow$ c is the positive real root of the polynomial

$$2kx^{2k+1} - x^L(2k+1)x^{2k} + (x^L)^{2k+1}$$

- obtain convex envelope and linear approximation

k	c (for $x^L = -1$)
1	0.5000000000
2	0.6058295862
3	0.6703320476
4	0.7145377272
5	0.7470540749
6	0.7721416355
7	0.7921778546

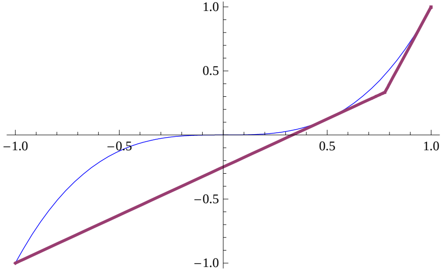
[Liberti and Pantelides, 2003]

Convex envelope for monomials of odd degree

$$g(x) = x^{2k+1}, \qquad k \in \mathbb{N}$$

- Seek for secant from x^L to c, where $c \geq 0$ is the coordinate where

$$g'(c) = \frac{c^{2k+1} - (x^L)^{2k+1}}{c - x^L}$$



$\Rightarrow$ c is the positive real root of the polynomial

$$2kx^{2k+1} - x^L(2k+1)x^{2k} + (x^L)^{2k+1}$$

k	c (for $x^L = -1$)
1	0.5000000000
2	0.6058295862
3	0.6703320476
4	0.7145377272
5	0.7470540749
6	0.7721416355
7	0.7921778546

- obtain convex envelope and linear approximation
- easily generalized to $x|x|^{n-1} = \text{sign}(x)|x|^n$, $n \geq 1$ (application for flow of water and gas in pipes)

[Liberti and Pantelides, 2003]

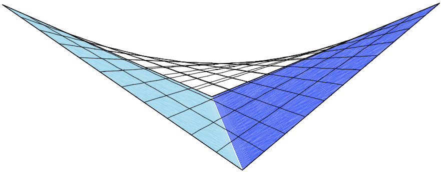
Convex envelope for $x \cdot y$

McCormick [McCormick, 1976]:
The convex envelope of

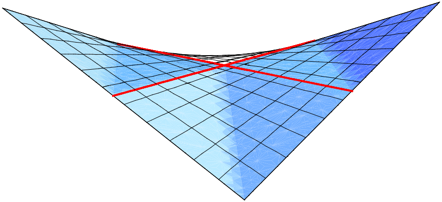
$$f(x) = xy$$

for $x \in [x^L, x^U]$, $y \in [y^L, y^U]$ is

$$xy \geq \max \left\{ \begin{array}{c} x^U y + y^U x - x^U y^U \\ x^L y + y^L x - x^L y^L \end{array} \right\}$$



Convex envelope for $x \cdot y$



McCormick [McCormick, 1976]:
The convex envelope of

$$f(x) = xy$$

for $x \in [x^L, x^U]$, $y \in [y^L, y^U]$ is

$$xy \geq \max \left\{ \begin{array}{c} x^U y + y^U x - x^U y^U \\ x^L y + y^L x - x^L y^L \end{array} \right\}$$

- apply McCormick underestimators to (nonconvex) quadratic forms

$$f(x) = \sum_{i,j} a_{i,j} x_i x_j$$

- branching on x and y changes variable bounds
  $\Rightarrow$ tighter underestimator $\Rightarrow$ improving dual bounds

Reformulation for the general (factorable) case

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation for the general (factorable) case

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$g = \sqrt{\exp(y_1) y_2}$$
$$y_1 = x_1 x_2 + x_3 \ln(x_4)$$
$$y_2 = x_3^3$$

[Tawarmalani and Sahinidis, 2004]

Reformulation for the general (factorable) case

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{\exp(y_1) y_2} \\
y_1 &= x_1 x_2 + x_3 \ln(x_4) \\
y_2 &= x_3^3
\end{aligned}
$$

[Tawarmalani and Sahinidis, 2004]

Reformulation for the general (factorable) case

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_3 y_2} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4)
\end{aligned}
$$

[Tawarmalani and Sahinidis, 2004]

Reformulation for the general (factorable) case

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_3 y_2} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4)
\end{aligned}
$$

[Tawarmalani and Sahinidis, 2004]

Reformulation for the general (factorable) case

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_5} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4) \\
y_5 &= y_3 y_2
\end{aligned}
$$

[Tawarmalani and Sahinidis, 2004]

Reformulation for the general (factorable) case

Let $g : \mathbb{R}^m \to \mathbb{R}$ be factorable, i.e., a recursive sum and product of univariate functions

Examples: $g(x) = x_1 x_2$, $g(x) = x_1/x_2$, $g(x) = \sqrt{\exp(x_1 x_2 + x_3 \ln x_4) x_3^3}$

Reformulation: replace products of functions or variables by new variables

$$
\begin{aligned}
g &= \sqrt{y_5} \\
y_1 &= x_1 x_2 + x_3 y_4 \\
y_2 &= x_3^3 \\
y_3 &= \exp(y_1) \\
y_4 &= \ln(x_4) \\
y_5 &= y_3 y_2
\end{aligned}
$$

Convex outer-approximation by

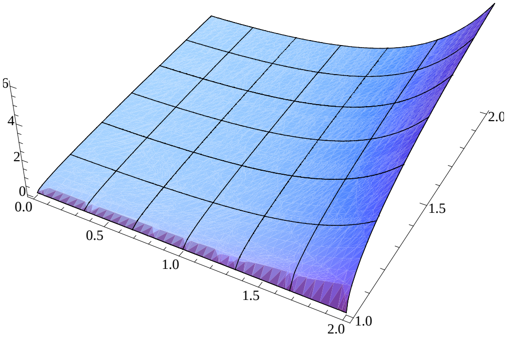
- estimating univariate functions by known convex envelopes
- estimating bilinear terms by McCormick underestimators

[Tawarmalani and Sahinidis, 2004]

Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$

$$x_1 \in [0,2], \quad x_2 \in [1,2]$$



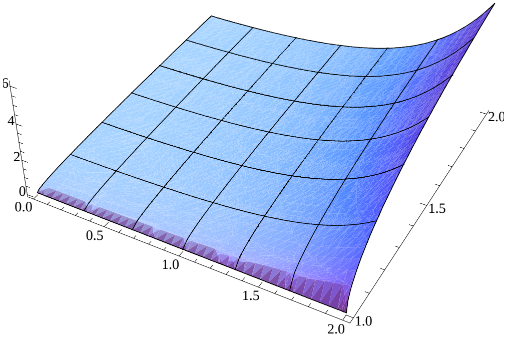
Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$

$$x_1 \in [0, 2], \quad x_2 \in [1, 2]$$

Factorable Reformulation:

$$g = \sqrt{y_1}$$
$$y_1 = y_2 y_3$$
$$y_2 = \exp(y_4)$$
$$y_3 = \ln(x_2)$$
$$y_4 = x_1^2$$

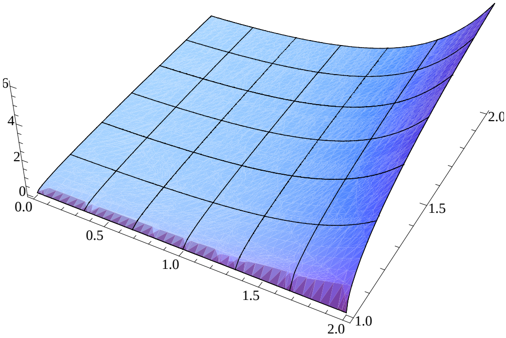


Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$

Factorable Reformulation:

$$
\begin{aligned}
g &= \sqrt{y_1} & & \\
y_1 &= y_2 y_3 & & [0, \ln(2)e^4] \\
y_2 &= \exp(y_4) & & [1, e^4] \\
y_3 &= \ln(x_2) & & [0, \ln(2)] \\
y_4 &= x_1^2 & & [0, 4]
\end{aligned}
$$

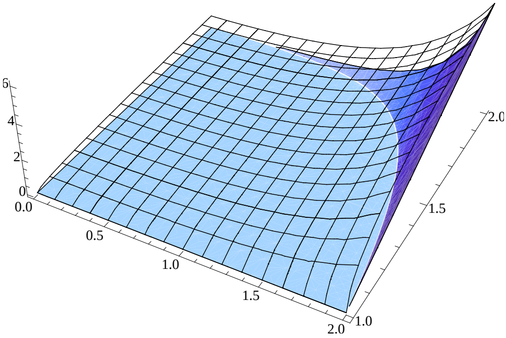


Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$

Factorable Reformulation:

$$\begin{aligned}
g &= \sqrt{y_1} & \\
y_1 &= y_2 y_3 & [0, \ln(2)e^4] \\
y_2 &= \exp(y_4) & [1, e^4] \\
y_3 &= \ln(x_2) & [0, \ln(2)] \\
y_4 &= x_1^2 & [0, 4]
\end{aligned}$$



Convex relaxation:

$$\sqrt{y_1^L} + \frac{y_1 - y_1^L}{\sqrt{y_1^U} + \sqrt{y_1^L}} \le g \le \sqrt{y_1}; \quad \ln x_1^L + (x_2 - x_2^L)\frac{\ln x_2^U - \ln x_2^L}{x_2^U - x_2^L} \le y_3 \le \ln(x_2)$$

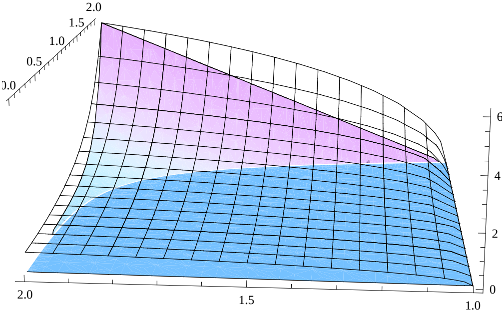
$$\max\left\{ \begin{matrix} y_2^U y_3 + y_3^U y_2 - y_2^U y_3^U \\ y_2^L y_3 + y_3^L y_2 - y_2^L y_3^L \end{matrix} \right\} \le y_1 \le \min\left\{ \begin{matrix} y_2^U y_3 + y_3^L y_2 - y_2^U y_3^L \\ y_2^L y_3 + y_3^U y_2 - y_2^L y_3^U \end{matrix} \right\}$$

...

Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$

$$x_1 \in [0,2], \quad x_2 \in [1,2]$$



Factorable Reformulation:

$$g = \sqrt{y_1}$$

$$y_1 = y_2 y_3 \qquad [0, \ln(2)e^4]$$

$$y_2 = \exp(y_4) \qquad [1, e^4]$$

$$y_3 = \ln(x_2) \qquad [0, \ln(2)]$$

$$y_4 = x_1^2 \qquad [0, 4]$$

Convex relaxation:

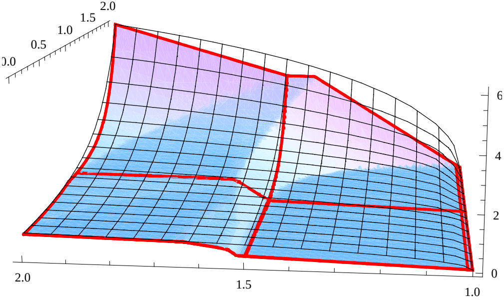
$$\sqrt{y_1^L} + \frac{y_1 - y_1^L}{\sqrt{y_1^U} + \sqrt{y_1^L}} \le g \le \sqrt{y_1}; \quad \ln x_1^L + (x_2 - x_2^L)\frac{\ln x_2^U - \ln x_2^L}{x_2^U - x_2^L} \le y_3 \le \ln(x_2)$$

$$\max\left\{ \begin{array}{c} y_2^U y_3 + y_3^U y_2 - y_2^U y_3^U \\ y_2^L y_3 + y_3^L y_2 - y_2^L y_3^L \end{array} \right\} \le y_1 \le \min\left\{ \begin{array}{c} y_2^U y_3 + y_3^L y_2 - y_2^U y_3^L \\ y_2^L y_3 + y_3^U y_2 - y_2^L y_3^U \end{array} \right\}$$

...

Example for convexification via factorable reformulation

$$g(x) = \sqrt{\exp(x_1^2)\ln(x_2)}$$
$$x_1 \in [0,2], \quad x_2 \in [1,2]$$



Factorable Reformulation:

$g = \sqrt{y_1}$		
$y_1 = y_2 y_3$	$[0, \ln(2)e^4]$
$y_2 = \exp(y_4)$	$[1, e^4]$
$y_3 = \ln(x_2)$	$[0, \ln(2)]$
$y_4 = x_1^2$	$[0, 4]$

Convex relaxation:

$$\sqrt{y_1^L} + \frac{y_1 - y_1^L}{\sqrt{y_1^U} + \sqrt{y_1^L}} \le g \le \sqrt{y_1}; \quad \ln x_1^L + (x_2 - x_2^L)\frac{\ln x_2^U - \ln x_2^L}{x_2^U - x_2^L} \le y_3 \le \ln(x_2)$$

$$\max\left\{ \begin{array}{c} y_2^U y_3 + y_3^U y_2 - y_2^U y_3^U \\ y_2^L y_3 + y_3^L y_2 - y_2^L y_3^L \end{array} \right\} \le y_1 \le \min\left\{ \begin{array}{c} y_2^U y_3 + y_3^L y_2 - y_2^U y_3^L \\ y_2^L y_3 + y_3^U y_2 - y_2^L y_3^U \end{array} \right\}$$

...

Adding a stockpile to open pit mining production

- stockpile allows to store mined material before processing

[Bley et al., 2009a,b]

Adding a stockpile to open pit mining production

- stockpile allows to store mined material before processing
- BUT: materials of different quality (metal/ore ratio) are mixed

$$\frac{(\text{metal send from stockpile to processing})_t}{(\text{ore send from stockpile to processing})_t} = \frac{(\text{metal in stockpile at})_{t-1}}{(\text{ore in stockpile at})_{t-1}}$$

[Bley et al., 2009a,b]

Ask SCIP to solve it...

Ask SCIP to solve it...

```
SCIP version 1.1.0.11 [precision: 8 byte] [memory: block] [mode: optimized] [LP solver: CPLEX 11.21]
Copyright (c) 2002-2009 Konrad-Zuse-Zentrum fuer Informationstechnik Berlin (ZIB)

Original problem has 1856 linear constraints and 280 nonlinear constraints.
original problem has 1688 variables (280 bin, 0 int, 0 impl, 1408 cont) and 2136 constraints
[...]
presolved problem has 1431 variables (263 bin, 0 int, 0 impl, 1168 cont) and 2004 constraints
    115 constraints of type <varbound>
   1416 constraints of type <linear>
    229 constraints of type <logicor>
    244 constraints of type <quadratic>
Presolving Time: 0.20
There are nonconvex quadratic constraints.
[...]
SCIP Status        : problem is solved [optimal solution found]
Solving Time (sec) : 9.78
Solving Nodes      : 418
Primal Bound       : +7.32628041419470e+00 (24 solutions)
Dual Bound         : +7.32628041419470e+00
Gap                : 0.00 %
[...]
Constraints        :     Number  #Separate #Propagate    #EnfoLP    DomReds       Cuts      Conss   Children
  branchnonlinear  :          0          0          0         35          0          0          0         38
  integral         :          0          0          0        315         90          0          0        470
  varbound         :        115          2       4316         16         23          0          0          0
  linear           :       1416          2       4316         16       7222          0          0          0
  logicor          :        229          2       2437         16       1812          0          0          0
  quadratic        :        244        229       2107         41         54         66          0          0
  countsols        :          0          0          0         16          0          0          0          0
```

Ask SCIP to solve it...

```
SCIP version 1.1.0.11 [precision: 8 byte] [memory: block] [mode: optimized] [LP solver: CPLEX 11.21]
Copyright (c) 2002-2009 Konrad-Zuse-Zentrum fuer Informationstechnik Berlin (ZIB)

Original problem has 1856 linear constraints and 280 nonlinear constraints.
original problem has 1688 variables (280 bin, 0 int, 0 impl, 1408 cont) and 2136 constraints
[...]
presolved problem has 1431 variables (263 bin, 0 int, 0 impl, 1168 cont) and 2004 constraints
    115 constraints of type <varbound>
   1416 constraints of type <linear>
    229 constraints of type <logicor>
    244 constraints of type <quadratic>
Presolving Time: 0.20
There are nonconvex quadratic constraints.
[...]
SCIP Status         : problem is solved [optimal solution found]
Solving Time (sec) : 9.78
Solving Nodes       : 418
Primal Bound        : +7.32628041419470e+00 (24 solutions)
Dual Bound          : +7.32628041419470e+00
Gap                 : 0.00 %
[...]
Constraints         :     Number  #Separate #Propagate    #EnfoLP    DomRe
  branchnonlinear   :          0          0          0         35
  integral          :          0          0          0        315
  varbound          :        115          2       4316         16
  linear            :       1416          2       4316         16       72
  logicor           :        229          2       2437         16       18
  quadratic         :        244        229       2107         41
  countsols         :          0          0          0         16
```

MINLP can be fun!

Another convexification technique: α-underestimators

Given $g \in C^2(\mathbb{R}^n, \mathbb{R})$. Let

$$\alpha \leq \lambda_1(\nabla^2 f(x)) \qquad \forall x \in [x^L, x^U].$$

Then

$$g(x) + \frac{1}{2}\alpha(x - x^L)^\top (x^U - x)$$

is a convex underestimator of $g(x)$.

[Adjiman and Floudas, 1997]

Another convexification technique: α-underestimators

Given $g \in C^2(\mathbb{R}^n, \mathbb{R})$. Let

$$\alpha \leq \lambda_1(\nabla^2 f(x)) \qquad \forall x \in [x^L, x^U].$$

Then

$$g(x) + \frac{1}{2}\alpha(x - x^L)^\top (x^U - x)$$

is a convex underestimator of $g(x)$.

How to find α?

1. Use Interval-Arithmetic to compute Interval-Hessian:

$$\nabla^2 g([x^L, x^U]) = \begin{pmatrix} [g^L_{1,1}, g^U_{1,1}] & \cdots & [g^L_{1,n}, g^U_{1,n}] \\ \vdots & & \vdots \\ [g^L_{n,1}, g^U_{n,1}] & \cdots & [g^L_{n,n}, g^U_{n,n}] \end{pmatrix}.$$

   That is, $\frac{\partial^2 g}{\partial x_i \partial x_j}(x) \in [g^L_{i,j}, g^U_{i,j}]$ for all $x \in [x^L, x^U]$.

[Adjiman and Floudas, 1997]

Another convexification technique: α-underestimators

Given $g \in C^2(\mathbb{R}^n, \mathbb{R})$. Let

$$\alpha \le \lambda_1(\nabla^2 f(x)) \qquad \forall x \in [x^L, x^U].$$

Then

$$g(x) + \frac{1}{2}\alpha(x - x^L)^\top (x^U - x)$$

is a convex underestimator of $g(x)$.

How to find α?

1. Use Interval-Arithmetic to compute Interval-Hessian:

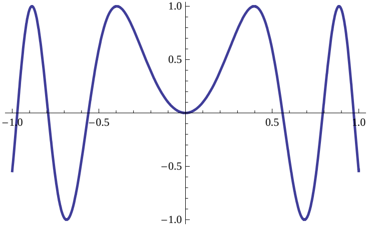
$$\nabla^2 g([x^L, x^U]) = \begin{pmatrix} [g^L_{1,1}, g^U_{1,1}] & \cdots & [g^L_{1,n}, g^U_{1,n}] \\ \vdots & & \vdots \\ [g^L_{n,1}, g^U_{n,1}] & \cdots & [g^L_{n,n}, g^U_{n,n}] \end{pmatrix}.$$

   That is, $\frac{\partial^2 g}{\partial x_i \partial x_j}(x) \in [g^L_{i,j}, g^U_{i,j}]$ for all $x \in [x^L, x^U]$.

2. Estimate α as minimal eigenvalue over 2^{n-1} vertex matrices of $\nabla^2 g([x^L, x^U])$. Or: use Gershgorin circles [Adjiman and Floudas, 1997]

(Bad) Example for α-underestimator

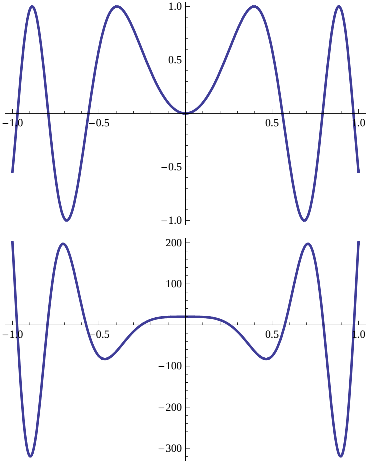
$$g(x) = \sin(10x^2), \qquad x \in [-1, 1]$$



(Bad) Example for α-underestimator

$$g(x) = \sin(10x^2), \qquad x \in [-1, 1]$$

$$\nabla^2 g(x) = 20\cos(10x^2) - 400x^2 \sin(10x^2)$$



(Bad) Example for α-underestimator

$$g(x) = \sin(10x^2), \qquad x \in [-1, 1]$$

$$\nabla^2 g(x) = 20\cos(10x^2) - 400x^2 \sin(10x^2)$$

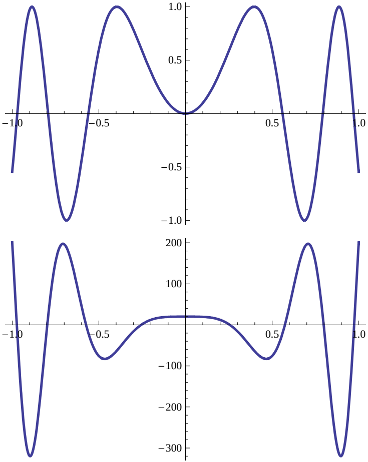
Bounds on $\lambda_1(\nabla^2 g(x))$:

- Tight:

$$\lambda_1(\nabla^2 g(x)) \geq -320$$

- By Intervalarithmetic:

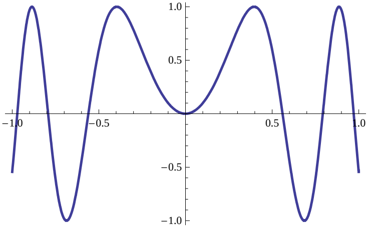
$$\lambda_1(\nabla^2 g(x)) \geq -420$$



(Bad) Example for α-underestimator

$$g(x) = \sin(10x^2), \qquad x \in [-1, 1]$$

$$\nabla^2 g(x) = 20\cos(10x^2) - 400x^2 \sin(10x^2)$$



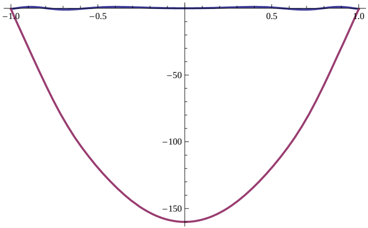
Bounds on $\lambda_1(\nabla^2 g(x))$:

- Tight:

$$\lambda_1(\nabla^2 g(x)) \geq -320$$

- By Intervalarithmetic:

$$\lambda_1(\nabla^2 g(x)) \geq -420$$



Convex underestimator:

$$\sin(10x^2) - \frac{320}{2}(x+1)(1-x) = \sin(10x^2) + 160x^2 - 160$$

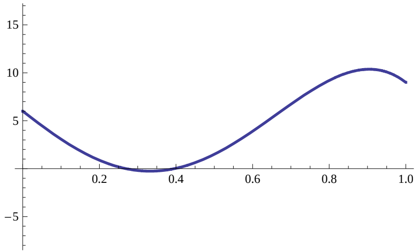
Underestimators for polynomials

$$f(x) = \sum_{i=0}^{\ell} a_i x^i$$

a multivariate polynomial over $[0, 1]$

(i and ℓ are multiindices).

$f(x) = 6 - 31x + 6x^2 + 124x^3 - 96x^4$



[Garloff, Jansson, and Smith, 2003]

Underestimators for polynomials

$$f(x) = \sum_{i=0}^{\ell} a_i x^i$$

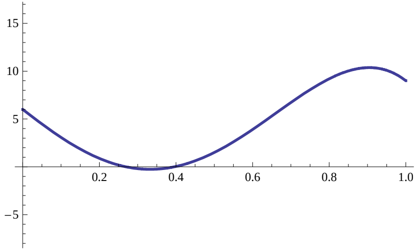
a multivariate polynomial over $[0,1]$

(i and ℓ are multiindices).

Bernstein basis:

$$B_i(x) = \binom{\ell}{i} x^i (1-x)^{\ell - i}$$

$f(x) = 6 - 31x + 6x^2 + 124x^3 - 96x^4$



[Garloff, Jansson, and Smith, 2003]

Underestimators for polynomials

$$f(x) = \sum_{i=0}^{\ell} a_i x^i$$

a multivariate polynomial over $[0, 1]$
(i and ℓ are multiindices).

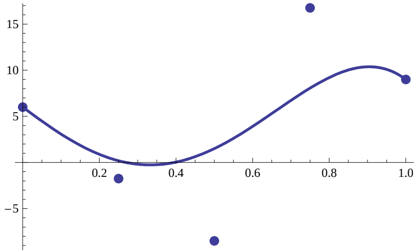
Bernstein basis:

$$B_i(x) = \binom{\ell}{i} x^i (1-x)^{\ell-i}$$

Write $f(x)$ w.r.t. Bernstein basis:

$$f(x) = \sum_{i=0}^{\ell} b_i B_i(x), \quad b_i = \sum_{j=0}^{\ell} \frac{\binom{i}{\ell}}{\binom{j}{\ell}} a_j$$

$f(x) = 6 - 31x + 6x^2 + 124x^3 - 96x^4$



[Garloff, Jansson, and Smith, 2003]

Underestimators for polynomials

$$f(x) = \sum_{i=0}^{\ell} a_i x^i$$

a multivariate polynomial over $[0,1]$ (i and ℓ are multiindices).

Bernstein basis:

$$B_i(x) = \binom{\ell}{i} x^i (1-x)^{\ell-i}$$

Write $f(x)$ w.r.t. Bernstein basis:

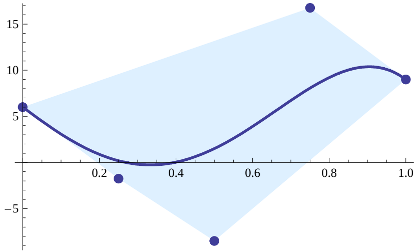
$$f(x) = \sum_{i=0}^{\ell} b_i B_i(x), \quad b_i = \sum_{j=0}^{\ell} \frac{\binom{i}{\ell}}{\binom{j}{\ell}} a_j$$

Convex Hull Property:

$$\{(x, f(x)) : x \in [0,1]\} \subseteq \text{conv}\,\{(i/\ell, b_i) : 0 \leq i \leq \ell\}$$

[Garloff, Jansson, and Smith, 2003]

$f(x) = 6 - 31x + 6x^2 + 124x^3 - 96x^4$



Underestimators for polynomials

$$f(x) = \sum_{i=0}^{\ell} a_i x^i$$

a multivariate polynomial over $[0,1]$

(i and ℓ are multiindices).

Bernstein basis:

$$B_i(x) = \binom{\ell}{i} x^i (1-x)^{\ell-i}$$

Write $f(x)$ w.r.t. Bernstein basis:

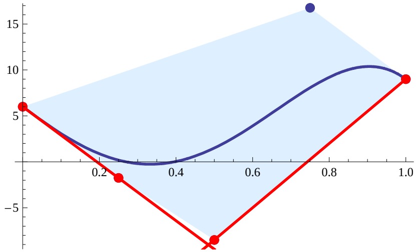
$$f(x) = \sum_{i=0}^{\ell} b_i B_i(x), \quad b_i = \sum_{j=0}^{\ell} \frac{\binom{i}{\ell}}{\binom{j}{\ell}} a_j$$

Convex Hull Property:

$$\{(x, f(x)) : x \in [0,1]\} \subseteq \text{conv}\,\{(i/\ell, b_i) : 0 \leq i \leq \ell\}$$

[Garloff, Jansson, and Smith, 2003]

$f(x) = 6 - 31x + 6x^2 + 124x^3 - 96x^4$



Underestimators for polynomials

$$f(x) = \sum_{i=0}^{\ell} a_i x^i$$

a multivariate polynomial over $[0,1]$

(i and ℓ are multiindices).

Bernstein basis:

$$B_i(x) = \binom{\ell}{i} x^i (1-x)^{\ell-i}$$

Write $f(x)$ w.r.t. Bernstein basis:

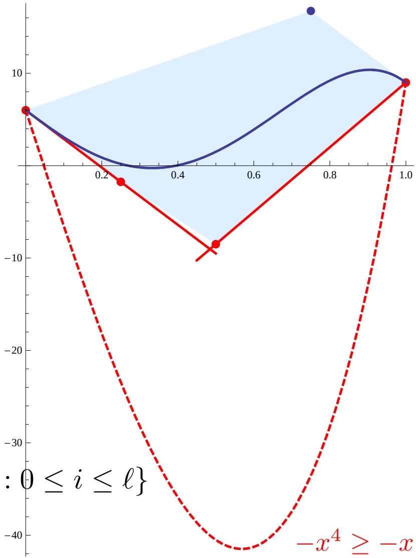
$$f(x) = \sum_{i=0}^{\ell} b_i B_i(x), \quad b_i = \sum_{j=0}^{\ell} \frac{\binom{i}{\ell}}{\binom{j}{\ell}} a_j$$

Convex Hull Property:

$$\{(x, f(x)) : x \in [0,1]\} \subseteq \text{conv}\,\{(i/\ell, b_i) : 0 \le i \le \ell\}$$

[Garloff, Jansson, and Smith, 2003]

$f(x) = 6 - 31x + 6x^2 + 124x^3 - 96x^4$



Solvers for nonconvex MINLP

BARON = Branch And Reduce Optimization Navigator [Tawarmalani and Sahinidis, 2002, 2004]

- developed by N. Sahinidis, M. Tawarmalani
- convexification for factorable programs

Solvers for nonconvex MINLP

BARON = Branch And Reduce Optimization Navigator [Tawarmalani and Sahinidis, 2002, 2004]

- developed by N. Sahinidis, M. Tawarmalani
- convexification for factorable programs

Couenne = Convex Over- and Under ENvelopes for Nonlinear Estimation [Belotti et al., 2008]

- extends Bonmin; developed by P. Belotti et.al.
- methodology similar to BARON

Solvers for nonconvex MINLP

BARON = Branch And Reduce Optimization Navigator [Tawarmalani and Sahinidis, 2002, 2004]

- developed by N. Sahinidis, M. Tawarmalani
- convexification for factorable programs

Couenne = Convex Over- and Under ENvelopes for Nonlinear Estimation [Belotti et al., 2008]

- extends Bonmin; developed by P. Belotti et.al.
- methodology similar to BARON

AlphaBB = α Branch-and-Bound [Adjiman et al., 1998a,b]

- developed by C. Adjiman, S. Dallwig, C.A. Floudas (Princeton), A. Neumaier (Vienna)
- convex α-underest. derived from Interval-Hessian

Solvers for nonconvex MINLP

BARON = Branch And Reduce Optimization Navigator [Tawarmalani and Sahinidis, 2002, 2004]

- developed by N. Sahinidis, M. Tawarmalani
- convexification for factorable programs

Couenne = Convex Over- and Under ENvelopes for Nonlinear Estimation [Belotti et al., 2008]

- extends Bonmin; developed by P. Belotti et.al.
- methodology similar to BARON

AlphaBB = α Branch-and-Bound [Adjiman et al., 1998a,b]

- developed by C. Adjiman, S. Dallwig, C.A. Floudas (Princeton), A. Neumaier (Vienna)
- convex α-underest. derived from Interval-Hessian

MINOTAUR = Mixed-Integer Nonconvex Optimization Toolbox – Algorithms, Underestimators, Relaxations

- J.T. Linderoth, S. Leyffer, T. Munson et.al.
- in development

Heuristics in MINLP

We know how to get dual bounds... What about the primal bounds?

Heuristics in MINLP

We know how to get dual bounds... What about the primal bounds?

- wait until the LP (or NLP) relaxation finds a feasible point

Heuristics in MINLP

We know how to get dual bounds... What about the primal bounds?

- wait until the LP (or NLP) relaxation finds a feasible point
- fix integer variables and solve remaining NLP locally

$$\min\{f(x) : g(x) \leq 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\ i \in I\}$$

Heuristics in MINLP

We know how to get dual bounds... What about the primal bounds?

- wait until the LP (or NLP) relaxation finds a feasible point
- fix integer variables and solve remaining NLP locally

$$\min\{f(x) : g(x) \leq 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\ i \in I\}$$

- extend a MIP heuristic to MINLP:

  BONMIN extends diving heuristics and feasibility pump [Bonami et al., 2009, Bonami and Gonçalves, 2008]

  Local Branching and VNS [Liberti et al., 2009, Nannicini et al., 2009]

  SCIP extends RENS to MINLP [Berthold et al., 2009]

Heuristics in MINLP

We know how to get dual bounds... What about the primal bounds?

- wait until the LP (or NLP) relaxation finds a feasible point
- fix integer variables and solve remaining NLP locally

$$\min\{f(x) : g(x) \leq 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\ i \in I\}$$

- extend a MIP heuristic to MINLP:

  BONMIN extends diving heuristics and feasibility pump [Bonami et al., 2009, Bonami and Gonçalves, 2008]

  Local Branching and VNS [Liberti et al., 2009, Nannicini et al., 2009]

  SCIP extends RENS to MINLP [Berthold et al., 2009]:
  - let $\tilde{x}$ be optimum of LP relaxation

Heuristics in MINLP

We know how to get dual bounds... What about the primal bounds?

- wait until the LP (or NLP) relaxation finds a feasible point
- fix integer variables and solve remaining NLP locally

$$\min\{f(x) : g(x) \leq 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\ i \in I\}$$

- extend a MIP heuristic to MINLP:

  BONMIN extends diving heuristics and feasibility pump [Bonami et al., 2009, Bonami and Gonçalves, 2008]

  Local Branching and VNS [Liberti et al., 2009, Nannicini et al., 2009]

  SCIP extends RENS to MINLP [Berthold et al., 2009]:

  - let $\tilde{x}$ be optimum of LP relaxation
  - create sub-MIQCP:
    1. copy MIP relaxation of current node into new SCIP instance
    2. copy quadratic (and nonlinear) constraints into new SCIP instance
    3. fix integer variables with integral value in $\tilde{x}$; restrict other integer variables to two nearest integral values

Heuristics in MINLP

We know how to get dual bounds... What about the primal bounds?

- wait until the LP (or NLP) relaxation finds a feasible point
- fix integer variables and solve remaining NLP locally

$$\min\{f(x) : g(x) \leq 0,\ x \in [x^L, x^U],\ x_i = x_i^*,\ i \in I\}$$

- extend a MIP heuristic to MINLP:

  BONMIN extends diving heuristics and feasibility pump [Bonami et al., 2009, Bonami and Gonçalves, 2008]

  Local Branching and VNS [Liberti et al., 2009, Nannicini et al., 2009]

  SCIP extends RENS to MINLP [Berthold et al., 2009]:
  - let $\tilde{x}$ be optimum of LP relaxation
  - create sub-MIQCP:
    1. copy MIP relaxation of current node into new SCIP instance
    2. copy quadratic (and nonlinear) constraints into new SCIP instance
    3. fix integer variables with integral value in $\tilde{x}$; restrict other integer variables to two nearest integral values
  - (partially) solve sub-MIQCP, including presolve

The Feasibility Pump for MINLP

$$\min\{c^\top x : g(x) \leq 0, \quad x \in [x^L, x^U], \quad x_i \in \mathbb{Z}, i \in I\}$$

[Bonami et al., 2009]

The Feasibility Pump for MINLP

$$\min\{c^\top x : g(x) \leq 0, \quad x \in [x^L, x^U], \quad x_i \in \mathbb{Z}, i \in I\}$$

Basic Idea: Alternating projection on

MIP relaxation

$$g(\bar{x}^k) + \nabla g(\bar{x}^k)(x - \bar{x}^k) \leq 0, \qquad k < j$$
$$x \in [x^L, x^U], \qquad x_i \in \mathbb{Z},\ i \in I$$

NLP relaxation

$$g(x) \leq 0$$
$$x \in [x^L, x^U]$$

[Bonami et al., 2009]

The Feasibility Pump for MINLP

$$\min\{c^\top x : g(x) \leq 0, \quad x \in [x^L, x^U], \quad x_i \in \mathbb{Z}, i \in I\}$$

Basic Idea: Alternate solution of

MIP relaxation

$$\begin{aligned} \min\ & \|x - \bar{x}^{j-1}\|_1 \\ \text{s.t. } & g(\bar{x}^k) + \nabla g(\bar{x}^k)(x - \bar{x}^k) \leq 0, \qquad k < j \\ & x \in [x^L, x^U], \qquad x_i \in \mathbb{Z},\ i \in I \end{aligned}$$

$\Rightarrow$ new solution $\hat{x}^j$

NLP relaxation

$$\begin{aligned} \min\ & \|x - \hat{x}^j\|_2 \\ \text{s.t. } & g(x) \leq 0 \\ & x \in [x^L, x^U] \end{aligned}$$

$\Rightarrow$ new solution $\bar{x}^j$

[Bonami et al., 2009]

The Feasibility Pump for MINLP

$$\min\{c^\top x : g(x) \leq 0, \quad x \in [x^L, x^U], \quad x_i \in \mathbb{Z}, i \in I\}$$

Basic Idea: Alternate solution of

MIP relaxation

$$\begin{aligned} \min\ & \|x - \bar{x}^{j-1}\|_1 \\ \text{s.t. } & g(\bar{x}^k) + \nabla g(\bar{x}^k)(x - \bar{x}^k) \leq 0, && k < j \\ & x \in [x^L, x^U], \quad x_i \in \mathbb{Z},\ i \in I \\ & (\bar{x}^k - \hat{x}^k)^\top (x - \bar{x}^k) \geq 0, && k < j \end{aligned}$$

$\Rightarrow$ new solution $\hat{x}^j$

NLP relaxation

$$\begin{aligned} \min\ & \|x - \hat{x}^j\|_2 \\ \text{s.t. } & g(x) \leq 0 \\ & x \in [x^L, x^U] \end{aligned}$$

$\Rightarrow$ new solution $\bar{x}^j$

[Bonami et al., 2009]

The Feasibility Pump for MINLP

$$\min\{c^\top x : g(x) \leq 0, \quad x \in [x^L, x^U], \quad x_i \in \mathbb{Z}, i \in I\}$$

Basic Idea: Alternate solution of

MIP relaxation

$$\begin{aligned} \min\ & \|x - \bar{x}^{j-1}\|_1 \\ \text{s.t. } & g(\bar{x}^k) + \nabla g(\bar{x}^k)(x - \bar{x}^k) \leq 0, \qquad k < j \\ & x \in [x^L, x^U], \qquad x_i \in \mathbb{Z},\ i \in I \\ & (\bar{x}^k - \hat{x}^k)^\top (x - \bar{x}^k) \geq 0, \qquad k < j \end{aligned}$$

$\Rightarrow$ new solution $\hat{x}^j$

NLP relaxation

$$\begin{aligned} \min\ & \|x - \hat{x}^j\|_2 \\ \text{s.t. } & g(x) \leq 0 \\ & x \in [x^L, x^U] \end{aligned}$$

$\Rightarrow$ new solution $\bar{x}^j$

- Convergence for convex problems: finds feasible point ($\bar{x}^j$) or proofs infeasibility
- Heuristic for nonconvex problems

[Bonami et al., 2009]

Domain Propagation (Bound Tightening)

Tighten variable bounds $[x^L, x^U]$

Domain Propagation (Bound Tightening)

Tighten variable bounds $[x^L, x^U]$ in order to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)

Domain Propagation (Bound Tightening)

Tighten variable bounds $[x^L, x^U]$ in order to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching

Domain Propagation (Bound Tightening)

Tighten variable bounds $[x^L, x^U]$ in order to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching
- very important: improve underestimators without branching $\Rightarrow$ tighter lower bounds

Domain Propagation (Bound Tightening)

Tighten variable bounds $[x^L, x^U]$ in order to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching
- very important: improve underestimators without branching $\Rightarrow$ tighter lower bounds

Applied in

- preprocessing (excessive)
- during Branch and Bound (restrained)

Domain Propagation (Bound Tightening)

Tighten variable bounds $[x^L, x^U]$ in order to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching
- very important: improve underestimators without branching $\Rightarrow$ tighter lower bounds

Applied in

- preprocessing (excessive)
- during Branch and Bound (restrained)

Linear Case:

$$y = \sum_{i \in \mathcal{P}} a_i x_i + \sum_{i \in \mathcal{N}} a_i x_i, \qquad (a_i > 0, i \in \mathcal{P},\ a_i < 0, i \in \mathcal{N})$$

Domain Propagation (Bound Tightening)

Tighten variable bounds $[x^L, x^U]$ in order to

- find infeasible subproblems (when bound tightening gives inconsistent bounds)
- reduce possible integral values of discrete variables $\Rightarrow$ less branching
- very important: improve underestimators without branching $\Rightarrow$ tighter lower bounds

Applied in

- preprocessing (excessive)
- during Branch and Bound (restrained)

Linear Case:

$$y = \sum_{i\in\mathcal{P}} a_i x_i + \sum_{i\in\mathcal{N}} a_i x_i, \qquad (a_i > 0, i \in \mathcal{P},\ a_i < 0, i \in \mathcal{N})$$

$$\Rightarrow \quad y \leq \sum_{i\in\mathcal{P}} a_i x_i^U + \sum_{i\in\mathcal{N}} a_i x_i^L$$

$$y \geq \sum_{i\in\mathcal{P}} a_i x_i^L + \sum_{i\in\mathcal{N}} a_i x_i^U$$

Interval Propagation on DAGs

Represent algebraic structure of problem in one directed acyclic graph:

- nodes: variables, operations, constraints
- arcs: flow of computation

[Schichl and Neumaier, 2005, Vu et al., 2008]

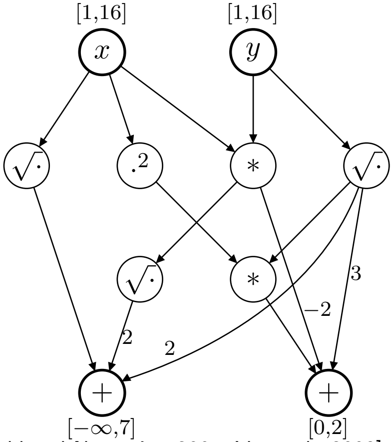
Interval Propagation on DAGs

Represent algebraic structure of problem in one directed acyclic graph:

- nodes: variables, operations, constraints
- arcs: flow of computation

Example:

$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$
$$x^2\sqrt{y} - 2xy + 3\sqrt{y} \in [0, 2]$$
$$x, y \in [1, 16]$$



[Schichl and Neumaier, 2005, Vu et al., 2008]

Interval Propagation on DAGs

Represent algebraic structure of problem in one directed acyclic graph:

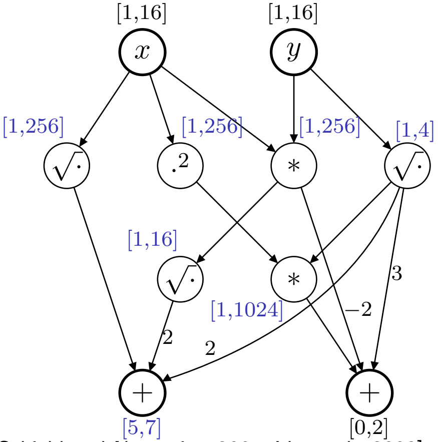
- nodes: variables, operations, constraints
- arcs: flow of computation

Example:

$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$
$$x^2\sqrt{y} - 2xy + 3\sqrt{y} \in [0, 2]$$
$$x, y \in [1, 16]$$

Forward propagation:

- compute bounds on intermediate nodes (top-down)



[Schichl and Neumaier, 2005, Vu et al., 2008]

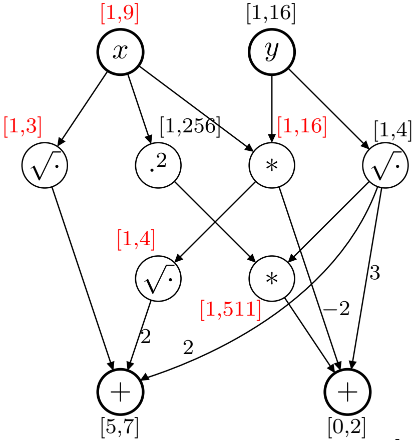
Interval Propagation on DAGs

Represent algebraic structure of problem in one directed acyclic graph:

- nodes: variables, operations, constraints
- arcs: flow of computation

Example:

$$\sqrt{x} + 2\sqrt{xy} + 2\sqrt{y} \in [-\infty, 7]$$
$$x^2\sqrt{y} - 2xy + 3\sqrt{y} \in [0, 2]$$
$$x, y \in [1, 16]$$



[Schichl and Neumaier, 2005, Vu et al., 2008]

Forward propagation:

- compute bounds on intermediate nodes (top-down)

Backward propagation:

- reduce bounds using reverse operations (bottom-up)

Reformulation: Power transformations for signomial terms

For $c, p_i \in \mathbb{R}$, $i = 1, \ldots, n$, let $$f(x) = c \prod_{i=1}^{n} x_i^{p_i}.$$

[Lundell et al., 2009]

Reformulation: Power transformations for signomial terms

For $c, p_i \in \mathbb{R}$, $i = 1, \ldots, n$, let $$f(x) = c \prod_{i=1}^{n} x_i^{p_i}.$$

$f(x)$ is convex if

$$p_i \leq 0 \quad \forall i$$

or $$\sum_{i=1}^{n} p_i \geq 1 \ \wedge \ \exists k \, \forall i \neq k : p_i \leq 0.$$

[Lundell et al., 2009]

Reformulation: Power transformations for signomial terms

For $c, p_i \in \mathbb{R}$, $i = 1, \ldots, n$, let $$f(x) = c \prod_{i=1}^{n} x_i^{p_i}.$$

$f(x)$ is convex if

$$p_i \leq 0 \quad \forall i$$

or $$\sum_{i=1}^{n} p_i \geq 1 \ \wedge \ \exists k \, \forall i \neq k : p_i \leq 0.$$

$f(x)$ is concave if

$$\sum_{i=1}^{n} p_i \leq 1 \quad \wedge \quad p_i \geq 0 \quad \forall i.$$

[Lundell et al., 2009]

Reformulation: Power transformations for signomial terms

For $c, p_i \in \mathbb{R}$, $i = 1, \ldots, n$, let $\quad f(x) = c \prod_{i=1}^{n} x_i^{p_i}$.

$f(x)$ is convex if

$$p_i \leq 0 \quad \forall i$$

or $\sum_{i=1}^{n} p_i \geq 1 \wedge \exists k \, \forall i \neq k : p_i \leq 0$.

$f(x)$ is concave if

$$\sum_{i=1}^{n} p_i \leq 1 \quad \wedge \quad p_i \geq 0 \quad \forall i.$$

Introduce transformations $\quad x_i = X_i^{Q_i}$. $\quad$ Then

$$c \prod_{i=1}^{n} X_i^{p_i Q_i} \quad \text{is convex/concave if}$$

$$\begin{cases} Q_i \geq 1, & \text{if } p_i > 0, i = k, \\ Q_i < 0, & \text{if } p_i > 0, i \neq k, \\ Q_i = 1 & \text{if } p_i < 0. \end{cases} \sum_{i=1}^{n} p_i Q_i \geq 1 \qquad \begin{cases} 0 < Q_i \leq 1, & \text{if } p_i > 0, \\ Q_i < 0, & \text{if } p_i < 0, \end{cases} \sum_{i=1}^{n} p_i Q_i \leq 1$$

[Lundell et al., 2009]

Reformulation: Power transformations for signomial terms

$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)

Reformulation: Power transformations for signomial terms

$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)
- find exponents Q_i so that the number of transformations is minimized
  $\Rightarrow$ find Q_i by solving MIP in preprocessing

Reformulation: Power transformations for signomial terms

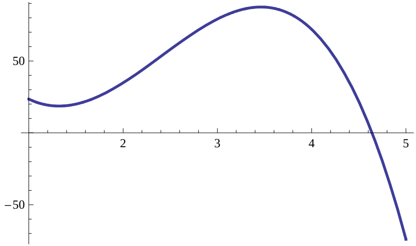
$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)
- find exponents Q_i so that the number of transformations is minimized
  $\Rightarrow$ find Q_i by solving MIP in preprocessing

Example: $f(x) = x^4 + 79.5x^2 - 170x + 120 - 7x^{3.5}, \quad x \in [1, 6]$

$\Rightarrow \quad f(X) = x^4 + 79.5x^2 - 170x + 120 - 7X$ and $x = X^{3.5}$

[Lundell et al., 2009]



Reformulation: Power transformations for signomial terms

$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

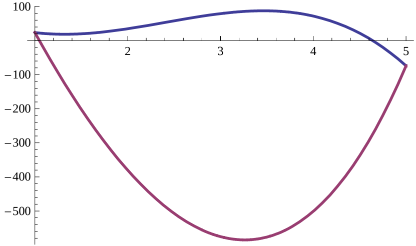
- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)
- find exponents Q_i so that the number of transformations is minimized
  $\Rightarrow$ find Q_i by solving MIP in preprocessing

Example: $f(x) = x^4 + 79.5x^2 - 170x + 120 - 7x^{3.5}, \quad x \in [1,6]$

$\Rightarrow \quad f(X) = x^4 + 79.5x^2 - 170x + 120 - 7X$ and $x = X^{3.5}$

approximating X piecewise linear with 1 breakpoint

[Lundell et al., 2009]



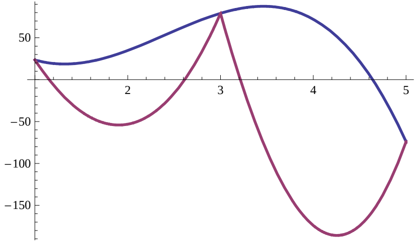
Reformulation: Power transformations for signomial terms

$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)
- find exponents Q_i so that the number of transformations is minimized
  $\Rightarrow$ find Q_i by solving MIP in preprocessing

Example: $f(x) = x^4 + 79.5x^2 - 170x + 120 - 7x^{3.5}, \quad x \in [1, 6]$
$\Rightarrow \quad f(X) = x^4 + 79.5x^2 - 170x + 120 - 7X$ and $x = X^{3.5}$

approximating X piecewise linear with 2 breakpoints



[Lundell et al., 2009]

Reformulation: Power transformations for signomial terms

$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

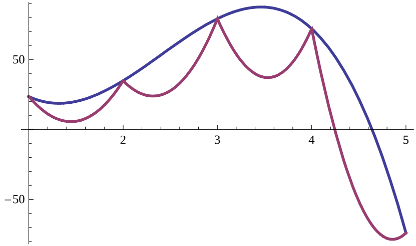
- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)
- find exponents Q_i so that the number of transformations is minimized
  $\Rightarrow$ find Q_i by solving MIP in preprocessing

Example: $f(x) = x^4 + 79.5x^2 - 170x + 120 - 7x^{3.5}, \quad x \in [1,6]$

$\Rightarrow \quad f(X) = x^4 + 79.5x^2 - 170x + 120 - 7X$ and $x = X^{3.5}$

approximating X piecewise linear with 4 breakpoints

[Lundell et al., 2009]



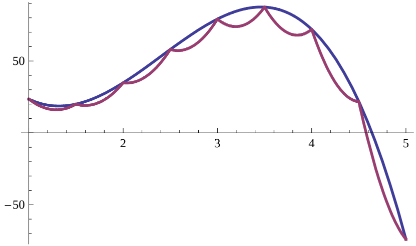
Reformulation: Power transformations for signomial terms

$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)
- find exponents Q_i so that the number of transformations is minimized
  $\Rightarrow$ find Q_i by solving MIP in preprocessing

Example: $f(x) = x^4 + 79.5x^2 - 170x + 120 - 7x^{3.5}, \quad x \in [1,6]$
$\Rightarrow \quad f(X) = x^4 + 79.5x^2 - 170x + 120 - 7X$ and $x = X^{3.5}$

approximating X piecewise linear with 8 breakpoints



[Lundell et al., 2009]

Reformulation: Power transformations for signomial terms

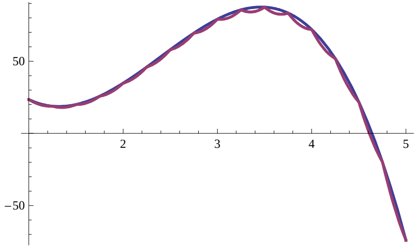
$$f(x) = c\prod_{i=1}^{n} x_i^{p_i} \qquad \Rightarrow \qquad f(X) = c\prod_{i=1}^{n} X_i^{p_i Q_i}, \quad x_i = X_i^{Q_i}$$

- approximate $X_i = x_i^{1/Q_i}$ by a piecewise linear approximation (SOS2)
- find exponents Q_i so that the number of transformations is minimized
  $\Rightarrow$ find Q_i by solving MIP in preprocessing

Example: $f(x) = x^4 + 79.5x^2 - 170x + 120 - 7x^{3.5}, \quad x \in [1, 6]$

$\Rightarrow \quad f(X) = x^4 + 79.5x^2 - 170x + 120 - 7X$ and $x = X^{3.5}$

approximating X piecewise linear with 16 breakpoints



[Lundell et al., 2009]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

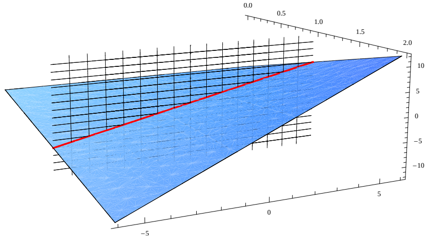
[Liberti and Pantelides, 2006]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

Simple example:

$A = \{(w, x, y) : w = xy \;\wedge\; x = 1\}$



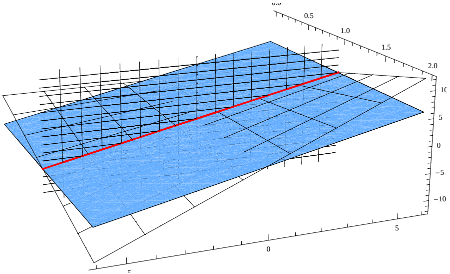
[Liberti and Pantelides, 2006]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

Simple example:
$A = \{(w, x, y) : w = xy \ \wedge \ x = 1\}$

Multiply $x = 1$ by $y \Rightarrow xy = y$
$\Rightarrow A = \{(w, x, y) : w = y \ \wedge \ x = 1\}$

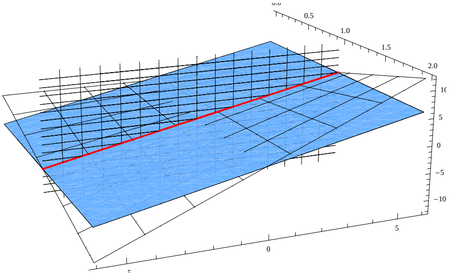


[Liberti and Pantelides, 2006]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

Simple example:
$A = \{(w, x, y) : w = xy \ \wedge \ x = 1\}$



Multiply $x = 1$ by $y \Rightarrow xy = y$
$\Rightarrow A = \{(w, x, y) : w = y \ \wedge \ x = 1\}$

General Idea:

$$w_j = x_j y, \qquad j = 1, \ldots, n,$$

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i$$

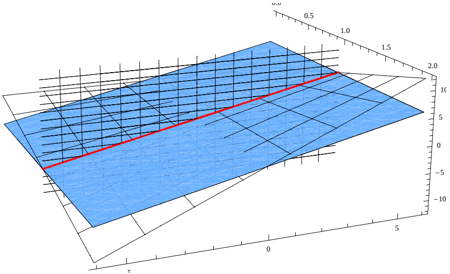
[Liberti and Pantelides, 2006]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

Simple example:
$A = \{(w, x, y) : w = xy \ \wedge \ x = 1\}$

Multiply $x = 1$ by $y \Rightarrow xy = y$
$\Rightarrow A = \{(w, x, y) : w = y \ \wedge \ x = 1\}$



General Idea:

$$w_j = x_j y, \qquad j = 1, \dots, n,$$

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i$$

$$\sum_{j=1}^{n} a_{i,j} x_j y = b_i y$$

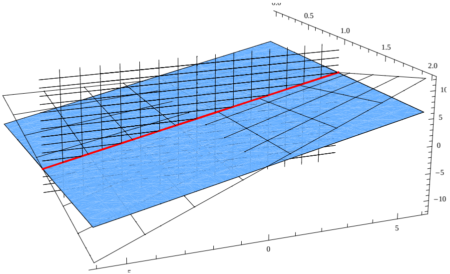
[Liberti and Pantelides, 2006]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

Simple example:
$A = \{(w, x, y) : w = xy \ \wedge \ x = 1\}$

Multiply $x = 1$ by $y \Rightarrow xy = y$
$\Rightarrow A = \{(w, x, y) : w = y \ \wedge \ x = 1\}$



General Idea:

$$w_j = x_j y, \qquad j = 1, \ldots, n,$$

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i$$

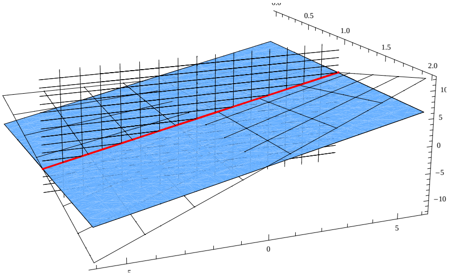
$$\sum_{j=1}^{n} a_{i,j} w_j = b_i y$$

[Liberti and Pantelides, 2006]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

Simple example:
$A = \{(w, x, y) : w = xy \ \wedge \ x = 1\}$



Multiply $x = 1$ by $y \Rightarrow xy = y$
$\Rightarrow A = \{(w, x, y) : w = y \ \wedge \ x = 1\}$

General Idea:

$$w_j = x_j y, \qquad j = 1, \ldots, n,$$

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i$$

$$\sum_{j=1}^{n} a_{i,j} w_j = b_i y \qquad \text{assume } a_{i,k} \neq 0 \quad \Rightarrow \ w_k = x_k y \text{ redundant}$$

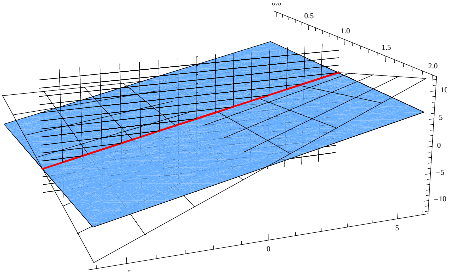
[Liberti and Pantelides, 2006]

Reformulation: Elimination of bilinear terms

Idea: Reveal hidden linearities in intersections of bilinear and linear constraints.

Simple example:
$A = \{(w, x, y) : w = xy \ \wedge \ x = 1\}$

Multiply $x = 1$ by $y \Rightarrow xy = y$
$\Rightarrow A = \{(w, x, y) : w = y \ \wedge \ x = 1\}$



General Idea:

$$w_j = x_j y, \qquad j = 1, \ldots, n, \quad j \neq k,$$

$$\sum_{j=1}^{n} a_{i,j} x_j = b_i$$

$$\sum_{j=1}^{n} a_{i,j} w_j = b_i y \qquad \text{assume } a_{i,k} \neq 0 \quad \Rightarrow \ w_k = x_k y \text{ redundant}$$

[Liberti and Pantelides, 2006]

Literature I

K. Abhishek, S. Leyffer, and J.T. Linderoth. FilMINT: An outer-approximation-based solver for nonlinear mixed integer programs. Technical Report ANL/MCS-P1374-0906, Argonne National Laboratory, Mathematics and Computer Science Division, 2006.

T. Achterberg. *Constraint Integer Programming*. PhD thesis, TU Berlin, 2007.

C. S. Adjiman and C. A. Floudas. Rigorous convex underestimators for general twice-differentiable problems. *Journal of Global Optimization*, 9:23–40, 1997.

C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. A global optimization method, αBB, for general twice-differentiable constrained NLPs – II. Implementation and computational results. *Computers & Chemical Engineering*, 22(9):1159–1179, 1998a.

C. S. Adjiman, S. Dallwig, C. A. Floudas, and A. Neumaier. A global optimization method, αBB, for general twice-differentiable constrained NLPs – I. Theoretical advances. *Computers & Chemical Engineering*, 22:1137–1158, 1998b.

Literature II

ARKI Consulting & Development A/S and GAMS Inc. SBB. http://www.gams.com/solvers/solvers.htm#SBB, 2002.

P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Waechter. Branching and bounds tightening techniques for non-convex MINLP. Technical Report RC24620, IBM Research, 2008.

T. Berthold, S. Heinz, and S. Vigerske. Extending a CIP framework to solve MIQCPs. Technical Report 09-23, Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB), 2009. URL http://opus.kobv.de/zib/volltexte/2009/1186.

Andreas Bley, Natashia Boland, Gary Froyland, and Mark Zuckerberg. Solving mixed integer nonlinear programming problems for mine production planning with a single stockpile. Technical Report 2009/21, Institute of Mathematics, TU Berlin, 2009a.

Andreas Bley, Ambros M. Gleixner, Thorsten Koch, and Stefan Vigerske. Comparing MIQCP solvers to a specialised algorithm for mine production scheduling. in preparation, 2009b.

Literature III

P. Bonami, L.T. Biegler, A.R. Conn, G. Cornuéjols, I.E. Grossmann, C.D. Laird, J. Lee, A. Lodi, F. Margot, N. Sawaya, and A. Wächter. An algorithmic framework for convex mixed integer nonlinear programs. *Discrete Optimization*, 5:186–204, 2008.

P. Bonami, G. Cornuéjols, A. Lodi, and F. Margot. A feasibility pump for mixed integer nonlinear programs. *Mathematical Programming*, 119:331–352, 2009.

Pierre Bonami and João P. M. Gonçalves. Primal heuristics for mixed integer nonlinear programs. Technical Report RC24639, IBM Research, 2008.

M. A. Duran and I. E. Grossmann. An outer-approximation algorithm for a class of mixed-integer nonlinear programs. *Mathematical Programming*, 36:307–339, 1986.

R. Fletcher and S. Leyffer. Solving Mixed Integer Nonlinear Programs by Outer Approximation. *Mathematical Programming*, 66(3(A)):327–349, 1994.

J. Garloff, C. Jansson, and A. P. Smith. Lower bound functions for polynomials. *Journal of Computational and Applied Mathematics*, 157:207–225, 2003.

Literature IV

S. Leyffer. Integrating SQP and branch-and-bound for mixed integer nonlinear programming. *Computational Optimization and Applications*, 18:295–309, 2001.

L. Liberti and C. Pantelides. Convex envelopes of monomials of odd degree. *J. Global Optimization*, 25:157–168, 2003.

L. Liberti and C. Pantelides. An exact reformulation algorithm for large nonconvex NLPs involving bilinear terms. *Journal of Global Optimization*, 36: 161–189, 2006.

Leo Liberti, Giacomo Nannicini, and Nenad Mladenovic. A good recipe for solving minlps. In V. Maniezzo, T. Stützle, and S. Voss, editors, *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, volume 10 of *Annals of Information Systems*. Springer, 2009.

A. Lundell, J. Westerlund, and T. Westerlund. Some transformation techniques with applications in global optimization. *Journal of Global Optimization*, 43: 391–405, 2009.

Literature V

G.P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I-Convex Underestimating Problems. *Mathematical Programming*, 10:147–175, 1976.

G. Nannicini, P. Belotti, and L. Liberti. A local branching heuristic for MINLPs. ArXiv, paper 0812.2188, 2009. URL `http://arxiv.org/abs/0812.2188`.

L. Quesada and I. E. Grossmann. An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. *Comp. Chem. Eng.*, 16:937–947, 1992.

H. Schichl and A. Neumaier. Interval analysis on directed acyclic graphs for global optimization. *J. Global Optimization*, 33:541–562, 2005.

M. Tawarmalani and N. V. Sahinidis. Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming*, 99:563–591, 2004.

M. Tawarmalani and N.V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*. Kluwer Academic Publishers, 2002.

Literature VI

X. Vu, H. Schichl, and D. Sam-Haroud. Interval propagation and search on directed acyclic graphs for numerical constraint solving. available online at http://www.mat.univie.ac.at/~herman/papers.html, 2008.

T. Westerlund and R. Pörn. Solving pseudo-convex mixed integer optimization problems by cutting plane techniques. *Optimization and Engineering*, 3: 253–280, 2002.

End

Let's stop here (I'M HUNGRY!)...