Alexander Hebert

Feasibility cuts in the L-shaped method

Step 2 of the L-shaped method determines feasibility cuts for the stochastic LP.

Let $c = [0 \mid e \mid e]^T,\ x = [y \mid v^+ \mid v^-]^T,\ A = [\,W \mid I \mid \quad I\,]$, and $b_k = \quad_k \quad T_k x^v$

Step 2: For $k = 1 \dots K$ until $\tilde{J}(y, v^+, v^-) = 0$, solve the linear program

$$\min \tilde{J}\,(y, v^+, v^-) = e^T v^+ + e^T v^-$$

subject to $Wy + Iv^+ \quad Iv^- = \quad_k \quad T_k x^v$
$y \geq 0,\ v^+ \geq 0,\ v^- \geq 0$

where $e = [1, \dots, 1]^T$ with the same number of rows as W, and
v^+ and v^- are column vectors of auxiliary variables also with the same number of rows as W.

For $0 \leq k \leq K$ while $\tilde{J}(y, v^+, v^-) > 0$, solve the linear program.

Instead of solving the primal in step 2 directly, we solve the associated dual LP.

Let σ_k^v be a column vector of simplex multipliers for the dual LP.

Step 2 dual: For $k = 1 \dots K$ until $\tilde{J}'(\sigma_k^v) = 0$, solve the linear program

$$\max \tilde{J}'(\sigma) = b_k^T \sigma$$

subject to $A^T \sigma + s = c,\ s \geq 0$

or the equivalent inequality constraint $A^T \sigma \leq c$

For $0 \leq k \leq K$ while $\tilde{J}'(\sigma_k^v) < 0$, solve the linear program.

Define the feasibility cut for step 1:

$$D_l x \geq d_l\,,\quad l = 1 \dots r$$

where

$$D_{r+1} = (\sigma_k^v)^T T_k$$
$$d_{r+1} = (\sigma_k^v)^T \quad_k$$

If $\tilde{J}'(\sigma) = 0$ for all $0 \leq k \leq K$, go to step 3.

Feasibility cuts example for step 2 of the L-shaped method

This example is from *Introduction to Stochastic Programming* by J. R. Birge and F. Louveaux.

Example (Feasibility cuts): We consider the minimization problem

$$
\begin{aligned}
(1.47) \qquad & \text{minimize } 3x_1 + 2x_2 + E_{\boldsymbol{\xi}}(15y_1 + 12y_2) \\
& \text{subject to } 3y_1 + 2y_2 \le x_1 \ , \\
& 2y_1 + 5y_2 \le x_2 \ , \\
& 0.8\boldsymbol{\xi}_1 \le y_1 \le \boldsymbol{\xi}_1 \ , \\
& 0.8\boldsymbol{\xi}_2 \le y_2 \le \boldsymbol{\xi}_2 \ , \\
& x \ge 10 \ , \ y \ge 0 \ ,
\end{aligned}
$$

where $\boldsymbol{\xi} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2)^T$ with $\boldsymbol{\xi}_1 \in \{4, 6\}, \boldsymbol{\xi}_2 \in \{4, 8\}$ independently with probability 1/2 each.

The feasibility cuts for realization $\xi = (6,8)$ were computed for the above example using the Matlab code listed at the end of this report. Matlab's built-in function *linprog* (with options set for the simplex algorithm) was used for the LP optimization in steps 1 and 2.

Matlab command window output:

```
xi1 =

     6


xi2 =

     8


c =

     3
     2


q =

    15
    12
     0
     0
     0
     0
     0
     0
```

```
W =
  Columns 1 through 6

     3     2     1     0     0     0
     2     5     0     1     0     0
     1     0     0     0     1     0
    -1     0     0     0     0     1
     0     1     0     0     0     0
     0    -1     0     0     0     0

  Columns 7 through 8

     0     0
     0     0
     0     0
     0     0
     1     0
     0     1

A =

  Columns 1 through 6

     3     2     1     0     0     0
     2     5     0     1     0     0
     1     0     0     0     1     0
    -1     0     0     0     0     1
     0     1     0     0     0     0
     0    -1     0     0     0     0

  Columns 7 through 12

     0     0     1     0     0     0
     0     0     0     1     0     0
     0     0     0     0     1     0
     0     0     0     0     0     1
     1     0     0     0     0     0
     0     1     0     0     0     0

  Columns 13 through 18

     0     0    -1     0     0     0
     0     0     0    -1     0     0
     0     0     0     0    -1     0
     0     0     0     0     0    -1
     1     0     0     0     0     0
     0     1     0     0     0     0

  Columns 19 through 20

     0     0
     0     0
     0     0
     0     0
    -1     0
     0    -1
```

```
h =
         0
         0
    6.0000
   -4.8000
    8.0000
   -6.4000


T =

    -1     0
     0    -1
     0     0
     0     0
     0     0
     0     0
```

Iteration 1:

```
x =

     0
     0


b =

         0
         0
    6.0000
   -4.8000
    8.0000
   -6.4000

Optimization terminated.

sigma =

   -0.4467
   -0.3329
   -0.0000
   -1.0000
    0.0000
   -1.0000


fval =

  -11.2000


exitflag =

     1
```

```
output =

         iterations: 7
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
     constrviolation: 2.6176e-011
       firstorderopt: 2.0941e-010


lambda =

    ineqlin: [20x1 double]
      eqlin: [0x1 double]
      upper: [6x1 double]
      lower: [6x1 double]


D1 =

    0.4467    0.3329


d1 =

   11.2000
```

Iteration 2:

```
Optimization terminated.

x =

    0.0000
   33.6478


fval =

   67.2955


exitflag =

     1


output =

         iterations: 5
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
     constrviolation: 0
       firstorderopt: 5.8312e-008
```

```
lambda =

    ineqlin: 6.0085
      eqlin: [0x1 double]
      upper: [2x1 double]
      lower: [2x1 double]


b =

    0.0000
   33.6478
    6.0000
   -4.8000
    8.0000
   -6.4000

Optimization terminated.

sigma =

   -0.8343
   -0.0000
   -0.0000
   -1.0000
    0.0000
   -1.0000


fval =

  -11.2000


exitflag =

     1


output =

         iterations: 7
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
    constrviolation: 7.7080e-011
      firstorderopt: 1.0615e-008


lambda =

    ineqlin: [20x1 double]
      eqlin: [0x1 double]
      upper: [6x1 double]
      lower: [6x1 double]
```

```
D2 =

    0.8343    0.0000


d2 =

   11.2000
```

Iteration 3:

```
D =

    0.4467    0.3329
    0.8343    0.0000


d =

   11.2000
   11.2000

Optimization terminated.

x =

   13.4241
   15.6314


fval =

   71.5350


exitflag =

     1


output =

         iterations: 6
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
     constrviolation: 0
      firstorderopt: 2.1113e-013


lambda =

    ineqlin: [2x1 double]
      eqlin: [0x1 double]
      upper: [2x1 double]
      lower: [2x1 double]
```

```
b =

   13.4241
   15.6314
    6.0000
   -4.8000
    8.0000
   -6.4000

Optimization terminated.


sigma =

   -0.2727
   -0.0909
   -0.0000
   -1.0000
   -0.0000
   -1.0000


fval =

   -6.1179


exitflag =

     1

output =

         iterations: 7
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
     constrviolation: 2.4471e-009
      firstorderopt: 7.9772e-009

lambda =

    ineqlin: [20x1 double]
      eqlin: [0x1 double]
      upper: [6x1 double]
      lower: [6x1 double]


D3 =

    0.2727    0.0909


d3 =

   11.2000
```

Iteration 4:

```
D =

    0.4467    0.3329
    0.8343    0.0000
    0.2727    0.0909


d =

   11.2000
   11.2000
   11.2000

Optimization terminated.

x =

   41.0667
    0.0000


fval =

  123.2000

exitflag =

     1

output =

         iterations: 6
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
      constrviolation: 0
       firstorderopt: 1.0552e-012


lambda =

    ineqlin: [3x1 double]
      eqlin: [0x1 double]
      upper: [2x1 double]
      lower: [2x1 double]

b =

   41.0667
    0.0000
    6.0000
   -4.8000
    8.0000
   -6.4000
```

```
Optimization terminated.

sigma =

   -0.0000
   -0.7483
    0.0000
   -1.0000
    0.0000
   -1.0000


fval =

  -11.2000


exitflag =

     1

output =

         iterations: 8
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
    constrviolation: 1.7479e-011
      firstorderopt: 1.6189e-009


lambda =

    ineqlin: [20x1 double]
      eqlin: [0x1 double]
      upper: [6x1 double]
      lower: [6x1 double]


D4 =

    0.0000    0.7483


d4 =

   11.2000
```

Iteration 5:

```
D =

    0.4467    0.3329
    0.8343    0.0000
    0.2727    0.0909
    0.0000    0.7483
```

```
d =

   11.2000
   11.2000
   11.2000
   11.2000

Optimization terminated.

x =

   36.0776
   14.9673


fval =

  138.1673


exitflag =

     1


output =

         iterations: 6
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
    constrviolation: 0
      firstorderopt: 1.7586e-013


lambda =

    ineqlin: [4x1 double]
      eqlin: [0x1 double]
      upper: [2x1 double]
      lower: [2x1 double]


b =

   36.0776
   14.9673
    6.0000
   -4.8000
    8.0000
   -6.4000

Optimization terminated.
```

```
sigma =

   -0.0000
   -0.2000
    0.0000
   -0.4000
    0.0000
   -1.0000


fval =

   -5.3265


exitflag =

     1


output =

         iterations: 8
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
    constrviolation: 2.8905e-011
      firstorderopt: 3.0181e-009


lambda =

    ineqlin: [20x1 double]
      eqlin: [0x1 double]
      upper: [6x1 double]
      lower: [6x1 double]


D5 =

    0.0000    0.2000


d5 =

    8.3200
```

Iteration 6:

```
D =

    0.4467    0.3329
    0.8343    0.0000
    0.2727    0.0909
    0.0000    0.7483
    0.0000    0.2000
```

```
d =

   11.2000
   11.2000
   11.2000
   11.2000
    8.3200

Optimization terminated.

x =

   27.2000
   41.6000


fval =

  164.8000


exitflag =

     1


output =

         iterations: 5
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
      constrviolation: 0
      firstorderopt: 2.1259e-011


lambda =

    ineqlin: [5x1 double]
      eqlin: [0x1 double]
      upper: [2x1 double]
      lower: [2x1 double]


b =

   27.2000
   41.6000
    6.0000
   -4.8000
    8.0000
   -6.4000

Optimization terminated.
```

```
sigma =

   -0.2554
   -0.0831
    0.0000
   -0.9324
         0
   -0.9264


fval =

 -4.3380e-009


exitflag =

     1

output =

         iterations: 9
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
    constrviolation: 3.9790e-013
      firstorderopt: 2.5006e-010


lambda =

    ineqlin: [20x1 double]
      eqlin: [0x1 double]
      upper: [6x1 double]
      lower: [6x1 double]


D6 =

    0.2554    0.0831


d6 =

   10.4047
```

Iteration 7:

```
D =

    0.4467    0.3329
    0.8343    0.0000
    0.2727    0.0909
    0.0000    0.7483
    0.0000    0.2000
    0.2554    0.0831
```

```
d =

   11.2000
   11.2000
   11.2000
   11.2000
    8.3200
   10.4047

Optimization terminated.

x =

   27.2000
   41.6000


fval =

  164.8000


exitflag =

     1


output =

         iterations: 5
          algorithm: [1x27 char]
       cgiterations: 0
            message: [1x24 char]
    constrviolation: 0
      firstorderopt: 6.3768e-008


lambda =

    ineqlin: [6x1 double]
      eqlin: [0x1 double]
      upper: [2x1 double]
      lower: [2x1 double]

>>
```

Matlab code:

```
clear; clc;
% Feasibility cuts example

xi1 = 6
xi2 = 8
c = [3;2]
q=[15;12;0;0;0;0;0;0]
W=[3,2,1,0,0,0,0,0;...
   2,5,0,1,0,0,0,0;...
   1,0,0,0,1,0,0,0;...
  -1,0,0,0,0,1,0,0;...
   0,1,0,0,0,0,1,0;...
   0,-1,0,0,0,0,0,1]

W_num_rows = size(W,1);
W_num_cols = size(W,2);

A = [W,eye(W_num_rows),-1*eye(W_num_rows)]

e = [zeros(W_num_cols,1);ones(W_num_rows,1);ones(W_num_rows,1)];

h=[0;0;xi1;-0.8*xi1;xi2;-0.8*xi2]
T=[-1*eye(2);zeros(4,2)]

%lb = zeros(size(A,1),1)
options = optimset('Simplex','on');

% Iter 1
disp('Iteration 1:')

x = [0;0]
b = h - T*x
[sigma,fval,exitflag,output,lambda] = linprog(-b,A',e,[],[])

%lambda.ineqlin
D1 = sigma'*T
d1 = sigma'*h

% Iter 2
disp('Iteration 2:')

[x,fval,exitflag,output,lambda] = linprog(c,-1*D1,-1*d1,[],[],[0;0])

b = h - T*x

[sigma,fval,exitflag,output,lambda] = linprog(-b,A',e,[],[])

D2 = sigma'*T
d2 = sigma'*h

% Iter 3
disp('Iteration 3:')

D = [D1;D2]
d = [d1;d2]

[x,fval,exitflag,output,lambda] = linprog(c,-1*D,-1*d,[],[],[0;0])
```

```
b = h - T*x

[sigma,fval,exitflag,output,lambda] = linprog(-b,A',e,[],[])

D3 = sigma'*T
d3 = sigma'*h

% Iter 4
disp('Iteration 4:')

D = [D;D3]
d = [d;d3]

[x,fval,exitflag,output,lambda] = linprog(c,-1*D,-1*d,[],[],[0;0])

b = h - T*x

[sigma,fval,exitflag,output,lambda] = linprog(-b,A',e,[],[])

D4 = sigma'*T
d4 = sigma'*h

% Iter 5
disp('Iteration 5:')

D = [D;D4]
d = [d;d4]

[x,fval,exitflag,output,lambda] = linprog(c,-1*D,-1*d,[],[],[0;0])

b = h - T*x

[sigma,fval,exitflag,output,lambda] = linprog(-b,A',e,[],[])

D5 = sigma'*T
d5 = sigma'*h

% Iter 6
disp('Iteration 6:')

D = [D;D5]
d = [d;d5]

[x,fval,exitflag,output,lambda] = linprog(c,-1*D,-1*d,[],[],[0;0])

b = h - T*x

[sigma,fval,exitflag,output,lambda] = linprog(-b,A',e,[],[])

D6 = sigma'*T
d6 = sigma'*h

% Iter 7
disp('Iteration 7:')

D = [D;D6]
d = [d;d6]

[x,fval,exitflag,output,lambda] = linprog(c,-1*D,-1*d,[],[],[0;0])
```