Prof. Friedrich Eisenbrand
Location: MA A3 31

Question session: 9.12.09
Discussion: 16.12.09

Exercises

Optimization Methods in Finance

Fall 2009

Sheet 6

Note: This is just <u>one</u> way, a solution could look like. We do not guarantee correctness. It is your task to find and report mistakes.

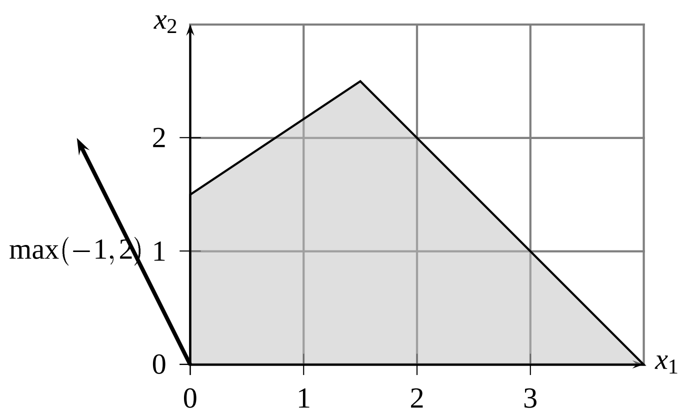
Exercise 6.1
Solve the following integer linear program with Branch & Bound.

$$\begin{aligned} &\min(1,-2)x \\ \begin{pmatrix} -4 & 6 \\ 1 & 1 \end{pmatrix} x &\leq \begin{pmatrix} 9 \\ 4 \end{pmatrix} \\ x &\geq \mathbf{0} \\ x &\in \mathbb{Z}^2 \end{aligned}$$

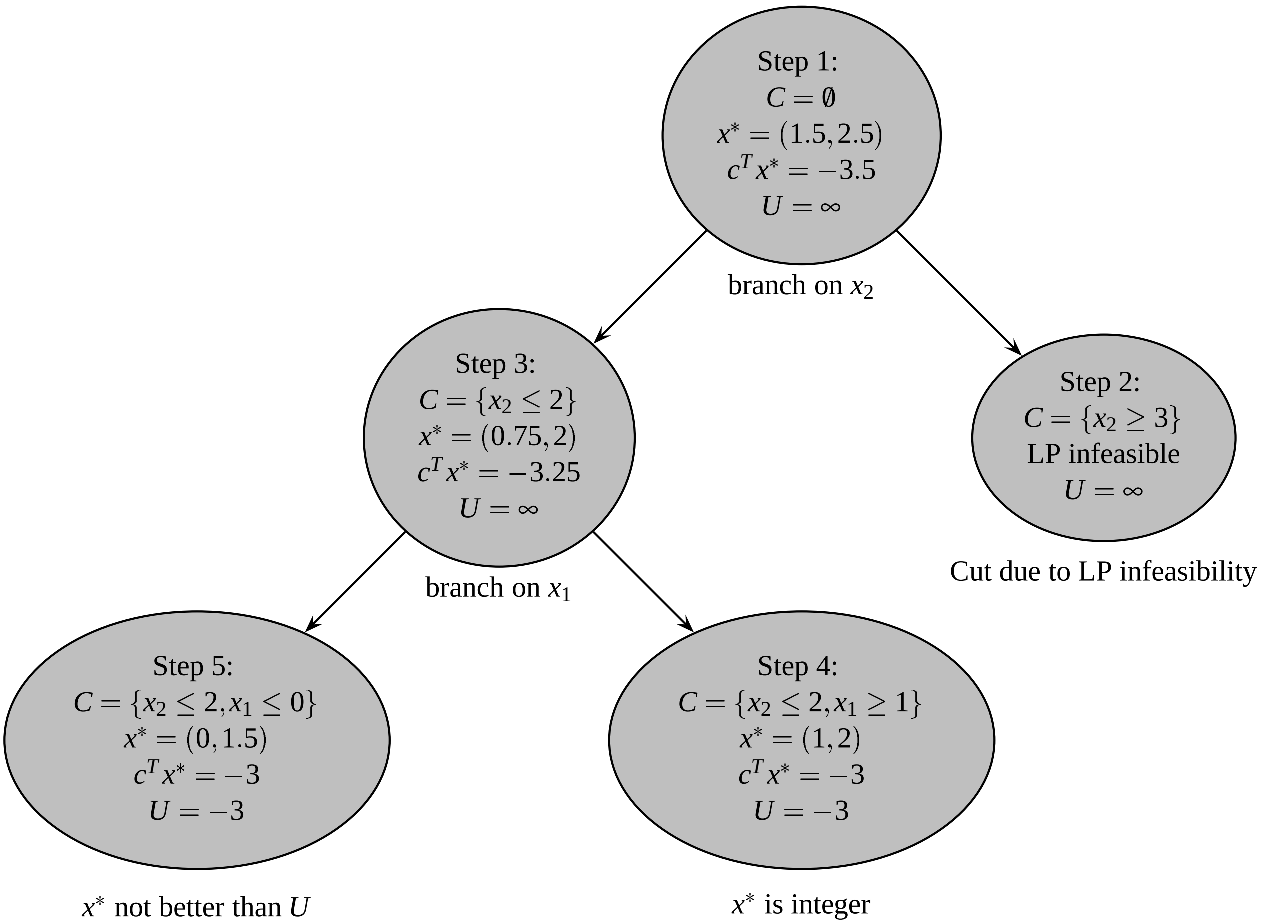
You can solve the LP subproblems with any computer algebra system or read optimum LP solutions from a drawing.

Solution:

The set of feasible solutions looks as follows



We display the branch & bound process in a tree structure. Each node corresponds to one iteration. C gives the additional constraints, U gives the value of the best found integer solution (at the end of the iteration). x^* gives the optimum fractional solution.



The optimum solution is consequently $x = (1, 2)$.

Exercise 6.2

We want to move to another city. We have n items which we would like to take. Each item $i \in \{1, \dots, n\}$ has a volume v_i and a weight w_i. We have m boxes, whereby box $j \in \{1, \dots, m\}$ has a volume V_j and can carry items of weight at most W_j.

Each item i that we cannot take with us, must be bought again for a price of c_i.

Introduce suitable decision variables (define their meaning) and state an integer linear program, which distributes the items among the boxes, such that for each box the volume bound and the weight bound are not exceeded and the cost of the items left behind is minimized.

Solution:

We use decision variables

$$x_{ij} = \begin{cases} 1 & \text{if item } i \text{ is put into box } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 & \text{if item } i \text{ is taken} \\ 0 & \text{otherwise} \end{cases}$$

Let $S := \sum_{i=1}^{n} c_i$ be the total cost of all items. Then the following ILP minimizes the cost of the items, left behind:

$$
\begin{aligned}
\min S - \sum_{i=1}^{n} y_i & & \\
\sum_{j=1}^{m} x_{ij} &= y_i \quad \forall i = 1,\dots,n \\
\sum_{i=1}^{n} v_i x_{ij} &\leq V_j \quad \forall j = 1,\dots,m \\
\sum_{i=1}^{n} w_i x_{ij} &\leq W_j \quad \forall j = 1,\dots,m \\
y_i, x_{ij} &\in \{0,1\}
\end{aligned}
$$

Exercise 6.3
In this exercise we develop a method, which computes a cutting plane, that cuts off optimal non-integer solutions.

Consider an integer linear program of the form

$$
\begin{aligned}
\min c^T x & \qquad (IP) \\
Ax &= b \\
x &\geq \mathbf{0} \\
x &\in \mathbb{Z}^n
\end{aligned}
$$

Suppose we use the simplex algorithm to compute a solution x^* to the LP relaxation (that means at least $Ax^* = b$ and $x^* \geq \mathbf{0}$). Let B be the optimal basis (i.e. $A_B x_B^* = b$, $x_{\bar{B}}^* = \mathbf{0}$). Assume there is index $i \in B$ with $x_i^* \notin \mathbb{Z}$. Since $Ax = b$ is a feasible equation for any solution x, also

$$
x_B + A_B^{-1} A_{\bar{B}} x_{\bar{B}} = \underbrace{A_B^{-1} A_B}_{=I} x_B + A_B^{-1} A_{\bar{B}} x_{\bar{B}} = A_B^{-1} A x = A_B^{-1} b \tag{1}
$$

holds for any feasible x. Abbreviate β as the ith entry of $A_B^{-1} b$ and let d be the ith row of $A_B^{-1} A_{\bar{B}}$. Then extracting the ith equation from (1) yields that

$$
x_i + \sum_{j \in \bar{B}} d_j x_j = \beta
$$

holds for any x with $Ax = b$. The *Gomory cut* is now the inequality $x_i + \sum_{j \in \bar{B}} \lfloor d_j \rfloor x_j \leq \lfloor \beta \rfloor$. Prove the following

i) The Gomory cut inequality holds for any solution x to (IP) (that means for any $x \in \mathbb{Z}_+^n$ with $Ax = b$).

ii) The cut inequality does not hold for the fractional solution $x^* \notin \mathbb{Z}^n$.

iii) An optimum solution to the LP relaxation of

$$
\begin{array}{rcl}
\min(1,-2)x & & \\
\begin{pmatrix} -4 & 6 \\ 1 & 1 \end{pmatrix} x & \leq & \begin{pmatrix} 9 \\ 4 \end{pmatrix} \\
x & \geq & \mathbf{0} \\
x & \in & \mathbb{Z}^2
\end{array}
$$

is $x^* = (1.5, 2.5)$. Obtain a Gomory cut, which cuts of x^*.

Solution:

For (1). Let $x \in \mathbb{Z}_+^n$ be an integer solution. We know that

$$x_i + \sum_{j \in \bar{B}} d_j x_j = \beta$$

Consequently

$$x_i + \sum_{j \in \bar{B}} \lfloor d_j \rfloor x_j \leq \beta$$

since $x_j \geq 0$ and $\lfloor d_j \rfloor \leq d_j$. But then the left hand side $x_i + \sum_{j \in \bar{B}} \lfloor d_j \rfloor x_j$ is integer and even

$$x_i + \sum_{j \in \bar{B}} \lfloor d_j \rfloor x_j \leq \lfloor \beta \rfloor$$

holds.

For (2). We know that

$$x_i^* = x_i^* + \sum_{j \in \bar{B}} \lfloor d_j \rfloor \underbrace{x_j^*}_{=0} = \beta$$

since $\bar{B}$ are non-basic variables. But we have chosen $i \in B$ with $x_i^* \notin \mathbb{Z}$, thus $x_i^* = \beta > \lfloor \beta \rfloor$ and x^* is cut off by the proposed inequality.

For (3). After adding slack variables the LP is

$$
\begin{array}{rcl}
\min(1,-2)x & & \\
\begin{pmatrix} -4 & 6 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{pmatrix} & = & \begin{pmatrix} 9 \\ 4 \end{pmatrix} \\
x & \geq & \mathbf{0} \\
x & \in & \mathbb{Z}^2
\end{array}
$$

(denote the left hand side matrix by A). A basis, corresponding to the solution $x^* = (1.5, 2.5, 0, 0)$ is $B = \{1, 2\}$. One has

$$A_B = \begin{pmatrix} -4 & 6 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad A_B^{-1} = \begin{pmatrix} -\frac{1}{10} & \frac{3}{5} \\ \frac{1}{10} & \frac{2}{5} \end{pmatrix}$$

Then $x_B + A_B^{-1} A_{\bar{B}} x_{\bar{B}} = A_B^{-1} b$ is the system

$$
\begin{array}{cccccc}
x_1 & & -0.1x_3 & +0.6x_4 & = & 1.5 \\
 & x_2 & +0.1x_3 & +0.4x_4 & = & 2.5
\end{array}
$$

We extract the second equation and round:

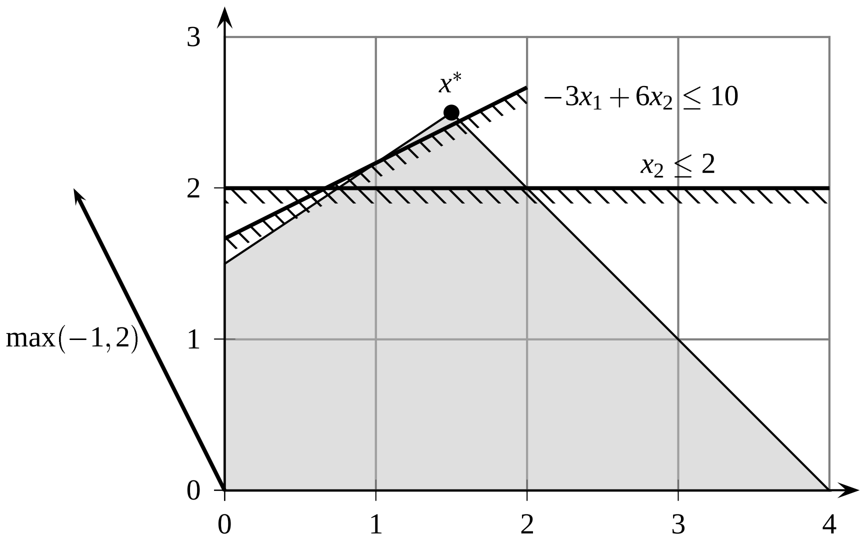
$$x_2 + \lfloor 0.1 \rfloor x_3 + \lfloor 0.4 \rfloor x_4 \leq \lfloor 2.5 \rfloor$$

thus $x_2 \leq 2$ is a feasible cut. Alternatively we can also use the first equation to obtain the Gomory Cut

$$x_1 + \lfloor -0.1 \rfloor x_3 + \lfloor 0.6 \rfloor x_4 \leq \lfloor 1.5 \rfloor \Rightarrow x_1 - x_3 \leq 1$$

To eliminate the slack variable x_3 we can add the equation $-4x_1 + 6x_2 + x_3 = 9$ to obtain the cut

$$x_1 - x_3 + (-4x_1 + 6x_2 + x_3) \leq 1 + 9 \Rightarrow -3x_1 + 6x_2 \leq 10$$



Exercise 6.4
Consider a lock-box problem, where c_{ij} is the cost of assigning region i to a lock-box in region j for $j = 1, \ldots, n$. Suppose we wish to open exactly q lock-boxes where $q \in \{1, \ldots, n\}$ is a given integer.

i) Formulate as an integer program the problem of opening exactly q lock-boxes so as to minimize the total cost of assigning each region to an open lock-box.

ii) Formulate in two different ways the constraint that regions cannot send checks to closed lock-boxes.

iii) For the following data

$$q = 2 \qquad \text{and} \qquad (c_{ij})_{1 \leq i,j \leq 5} = \begin{pmatrix} 0 & 4 & 5 & 8 & 2 \\ 4 & 0 & 3 & 4 & 6 \\ 5 & 3 & 0 & 1 & 7 \\ 8 & 4 & 1 & 0 & 4 \\ 2 & 6 & 7 & 4 & 0 \end{pmatrix}$$

compare the linear programming relaxations of your two formulations in question (ii).

Solution:

For (1). Use decision variables

$$x_{ij} = \begin{cases} 1 & \text{if region } i \text{ is assigned to lock-box } j \\ 0 & \text{otherwise} \end{cases}$$

$$y_j = \begin{cases} 1 & \text{if lock-box } j \text{ is opened} \\ 0 & \text{otherwise} \end{cases}$$

The lock-box ILP then is

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\begin{aligned} \sum_{j=1}^{n} y_j &= q \\ \sum_{j=1}^{n} x_{ij} &\geq 1 \quad \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} x_{ij} &\leq n \cdot y_j \quad \forall j = 1, \ldots, n \\ x_{ij}, y_j &\in \{0, 1\} \end{aligned}$$

For (2). The 3rd inequality in the ILP forbids that we can assign regions to closed lock-boxes. Alternatively it can be expressed with

$$x_{ij} \leq y_j \quad \forall i = 1, \ldots, n \, \forall j = 1, \ldots, n$$

Note that the integer solutions to both systems are exactly the same.

For (3). One obtains the LP relaxation for above ILP by replacing the constraint $x_{ij}, y_j \in \{0, 1\}$ by $0 \leq x_{ij}, y_j \leq 1$. An optimum fractional solution of the system in (1) is $x_{ii} = 1$ and $y_i = 0.4$ for $i = 1, \ldots, n$ (other values are 0). The objective value of this solution is 0.

On optimum solution of the same LP but with the constraint from (2) gives

$$y_3 = y_5 = 1, \quad x_{1,5} = x_{5,5} = x_{2,3} = x_{3,3} = x_{4,3} = 1$$

with a value of 6. This is even an integer solution.

Conclusion: The constraint from (2) is stronger (and therefore better). The solutions can again be obtained for example using ZIMPL+ QSOpt:

```
param n := 5;
param q := 2;
param c[{1..n}*{1..n}] := | 1,2,3,4,5 |
                        |1| 0,4,5,8,2 |
                        |2| 4,0,3,4,6 |
                        |3| 5,3,0,1,7 |
                        |4| 8,4,1,0,4 |
                        |5| 2,6,7,4,0 |;

var x[{1 to n}*{1 to n}] real >= 0 <= 1;
var y[{1 to n}] real >= 0 <= 1;
```

```
minimize cost: sum <i,j> in {1 to n}*{1 to n} : c[i,j]*x[i,j];

subto qNum:
  sum <j> in {1 to n}: y[j] == q;

subto assign:
  forall <i> in {1 to n} do
    sum <j> in {1 to n}: x[i,j] >= 1;

#subto closedLockbox1:
#  forall <j> in {1 to n} do
#    sum <i> in {1 to n} : x[i,j] <= n*y[j];

subto closedLockbox2:
  forall <i,j> in {1 to n}*{1 to n} do
    x[i,j] <= y[j];
```