Exercises

Optimization in Finance

Fall 2008

Sheet 4

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

Exercise 4.1
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

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

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

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

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];
```

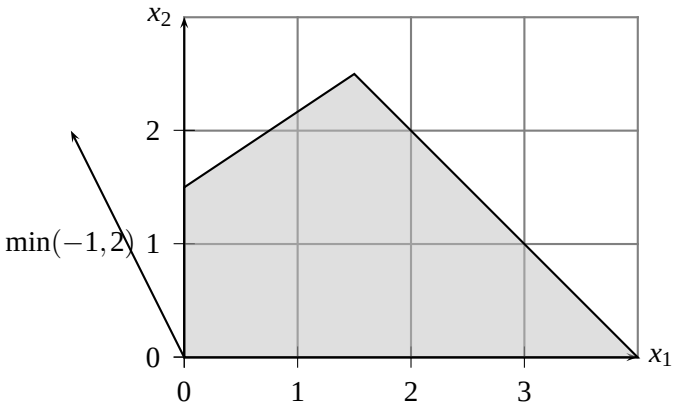
Exercise 4.2
Solve the following integer linear program with Branch & Bound.

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

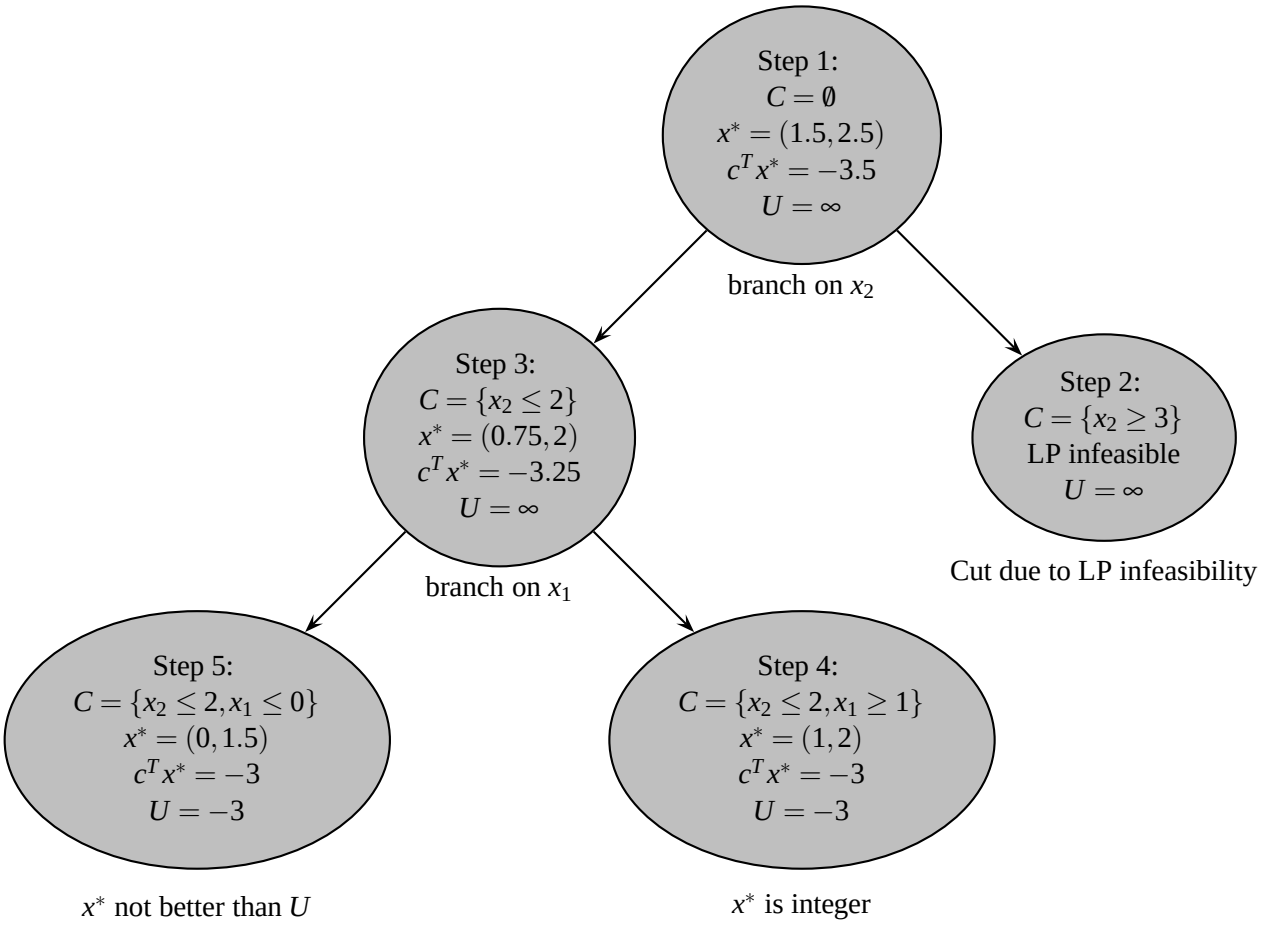
You can solve the LP subproblems with any computer algebra system.

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 4.3

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

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

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

1. The Gomory cut inequality holds for any solution x to (IP) (that means for any $x \in \mathbb{Z}_+^n$ with $Ax = b$).
2. The cut inequality does not hold for the fractional solution $x^* \notin \mathbb{Z}^n$.
3. 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

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

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

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

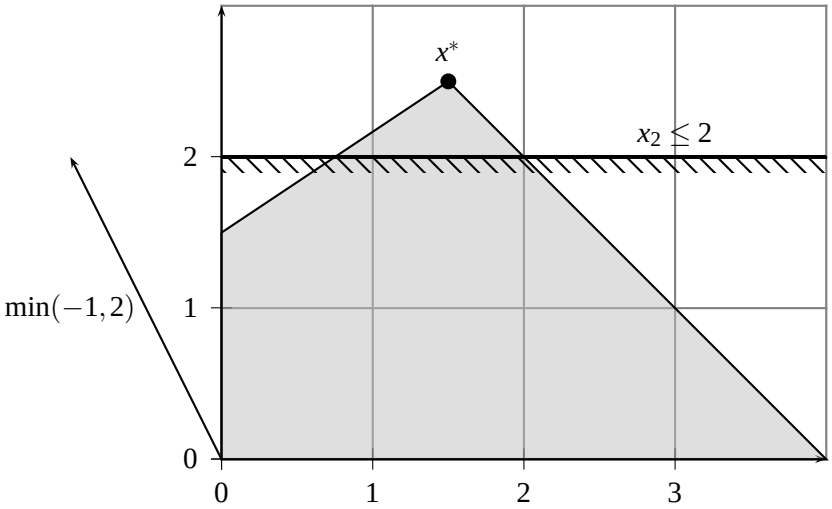
Then $x_B + A_B^{-1} A_{\bar{B}} x_{\bar{B}} = A_B^{-1} b$ is the system

$$\begin{array}{rrrcl} x_1 & -0.1x_3 & +0.3x_3 & = & 1.5 \\ x_2 & +0.1x_3 & +0.4x_4 & = & 2.5 \end{array}$$

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.



Exercise 4.4

We have a budget of 14.000.000 CHF and may invest into the following 4 projects

project	investment	net profit
1	7.000.000 CHF	4.101.000 CHF
2	5.000.000 CHF	3.430.000 CHF
3	4.000.000 CHF	2.005.000 CHF
4	3.000.000 CHF	1.210.000 CHF

The aim is to maximize the net profit, whereby the sum of the investments may not exceed our budget. The projects are 'take it or are leave it' decisions, that means we cannot just invest a fraction of the money. Solve this problem with a suitable dynamic programming approach.

Solution:

Clearly the problem can be formulated as Knapsack problem, where the items are investments. Recall that we have 2 different dynamic programs to solve Knapsack: One is fast for small weights, the other one is fast for small profits. Considering the numbers, clearly the first one is suitable. We compute table entries

$$A_{ij} = \text{profit (in 1000 CHF) which can be generated by investing } \leq j \text{ million CHF into projects } 1,\ldots,i$$

Let w_i be the investment needed for project i (in million CHF) and p_i its net profit (in 1000 CHF). Then the Bellman equations for computing table entries are

$$\begin{aligned} A_{1j} &= \begin{cases} p_1 & \text{if } w_1 \leq j \\ 0 & \text{otherwise} \end{cases} \\ A_{ij} &= \max\{ \underbrace{A_{i-1,j}}_{\text{don't take project } i}, \underbrace{A_{i-1,j-w_i} + p_i}_{\text{take project } i} \} \qquad \forall i > 1 \end{aligned}$$

(let $A_{ij} = -\infty$ for $j < 0$). Then

A_{ij}	j=0	j=1	j=2	j=3	j=4	j=5	j=6	j=7	j=8	j=9	j=10	j=11	j=12	j=13	j=14
i=1	0	0	0	0	0	0	0	4.101	4.101	4.101	4.101	4.101	4.101	4.101	4.101
i=2	0	0	0	0	0	3.430	3.430	4.101	4.101	4.101	4.101	4.101	7.531	7.531	7.531
i=3	0	0	0	0	2.005	3.430	3.430	4.101	4.101	5.435	5.435	6.106	7.531	7.531	7.531
i=4	0	0	0	1.210	2.005	3.430	3.430	4.101	4.640	5.435	5.435	6.106	7.531	7.531	7.531

We read from the table that a profit of 7.531.000 CHF is possible by choosing investments 1 and 2.

Exercise 4.5

Compute the value of an American call option on a stock with current price equal to 100 CHF, strike price equal to 102 CHF and expiration date four weeks from now. The yearly volatility of the logarithm of the stock return is $\sigma = 0.30$. The risk-free interest rate is 4%. Use a binomial lattice with $N = 4$.

Solution:

Using the formulas from the lecture (or the book of Cornuejols et al) one has

$$\Delta = \frac{1}{52}, \quad u = e^{\sigma\sqrt{\Delta}} = 1.0425, \quad d = e^{-\sigma\sqrt{\Delta}} = 0.9592,$$
$$R = 1 + \frac{0.04}{52} = 1.000769, \quad p_u = \frac{R-d}{u-d} = 0.4988, \quad p_d = \frac{u-R}{u-d} = 0.5012$$

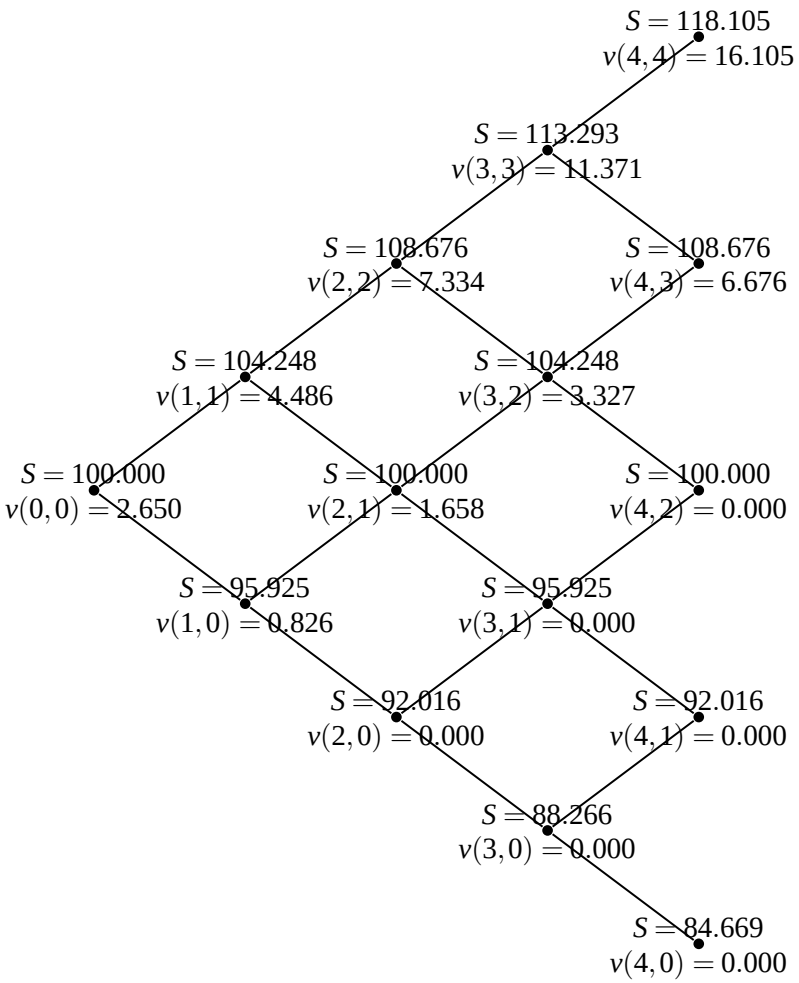
(strictly spoken some of this formulas are just approximating the true values). We compute $v(k,j)$ as the value of the option at time k if the underlying stock increased exactly j times. The formulas used in the dynamic program are then

$$v(n,j) = \max\{u^j d^{n-j} S_0 - c, 0\}$$

(since we are considering a call option) and

$$v(k,j) = \max\left\{\frac{1}{R}(p_u v(k+1,j+1) + p_d v(k+1,j)), u^j d^{k-j} S_0 - c\right\}$$

(since it is an American option) for $k = 0,\ldots,n-1$. The binomial lattice looks as follows



(S gives the price of the underlying asset w.r.t. time k and number of increases j). Thus the value of the option (at time 0) is $v(0,0) = 2.65$.

A side remark: Repeating the procedure with a larger binomial lattice of size $n = 1000$ gives a more precise price of 2.57.