Department of Mathematics and Computer Science
University of Southern Denmark, Odense

April 11, 2018
Marco Chiarandini

# DM559/DM545 – Linear and integer programming

## Sheet 3, Spring 2018 [pdf format]

**Solution:**
Included.

## Exercise 1* Simplex method

This is part of the first exercise (Opgave 1) in the exam of 2008
Consider the following linear programming problem (P1)

$$\begin{aligned} \text{maximize } & 2x_1 + 4x_2 - x_3 \\ \text{subject to } & 2x_1 - x_3 \leq 6 \\ & 3x_2 - x_3 \leq 9 \\ & x_1 + x_2 \leq 4 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

- Rewrite the problem in equational standard form adding the slack variables $x_4, x_5, x_6$ to the three constraints above, respectively, and write the first simplex tableau with $x_4, x_5, x_6$ as basis solution.

**Solution:**

We write the initial tableaux:

$$\begin{bmatrix} 2 & 0 & -1 & 1 & 0 & 0 & 0 & 6 \\ 0 & 3 & -1 & 0 & 1 & 0 & 0 & 9 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 & 4 \\ 2 & 4 & -1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Or also:

```
|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      2 |      0 |     -1 |      1 |      0 |      0 |      0 |      6 |
|      0 |      3 |     -1 |      0 |      1 |      0 |      0 |      9 |
|      1 |      1 |      0 |      0 |      0 |      1 |      0 |      4 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      2 |      4 |     -1 |      0 |      0 |      0 |      1 |      0 |
|--------+--------+--------+--------+--------+--------+--------+--------+
```

- Argue that $x_2$ can be brought in the basis with advantage and perform one pivot iteration that brings $x_2$ into the basis solution.

**Solution:**

$x_2$ has positive reduced cost, hence worth bringing up.

pivot column: 2 pivot row: 2 pivot: 3

$$\begin{bmatrix} 2 & 0 & -1 & 1 & 0 & 0 & 0 & 6 \\ 0 & 1 & -1/3 & 0 & 1/3 & 0 & 0 & 3 \\ 1 & 0 & 1/3 & 0 & -1/3 & 1 & 0 & 1 \\ 2 & 0 & 1/3 & 0 & -4/3 & 0 & 1 & -12 \end{bmatrix}$$

- After another pivot iteration, it is $x_1$ that can be brought with advantage in the basis (you do not have to perform this iteration), reaching the following simplex tableau:

```
|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |      0 |   -5/3 |      1 |    2/3 |     -2 |      0 |      4 |
|      0 |      1 |   -1/3 |      0 |    1/3 |      0 |      0 |      3 |
|      1 |      0 |    1/3 |      0 |   -1/3 |      1 |      0 |      1 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |      0 |   -1/3 |      0 |   -2/3 |     -2 |      1 |    -14 |
|--------+--------+--------+--------+--------+--------+--------+--------+
```

  Argue that an optimal solution is found and give the solution together with its objective value.

**Solution:**

The optimal solution is found because all reduced costs are non-positive. The objective function value is 14.
We show here for completeness all iterations of the simplex from the first tableau above:

```
pivot column: 2
pivot row: 2
 pivot: 3

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      2 |      0 |     -1 |      1 |      0 |      0 |      0 |      6 |
|      0 |      1 |   -1/3 |      0 |    1/3 |      0 |      0 |      3 |
|      1 |      0 |    1/3 |      0 |   -1/3 |      1 |      0 |      1 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      2 |      0 |    1/3 |      0 |   -4/3 |      0 |      1 |    -12 |
|--------+--------+--------+--------+--------+--------+--------+--------+

pivot column: 1
pivot row: 3
 pivot: 1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |      0 |   -5/3 |      1 |    2/3 |     -2 |      0 |      4 |
|      0 |      1 |   -1/3 |      0 |    1/3 |      0 |      0 |      3 |
|      1 |      0 |    1/3 |      0 |   -1/3 |      1 |      0 |      1 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |      0 |   -1/3 |      0 |   -2/3 |     -2 |      1 |    -14 |
|--------+--------+--------+--------+--------+--------+--------+--------+
```

We check the solution also with gurobi:

```
#!/usr/bin/python

from gurobipy import *

# Model
model = Model("prod")
model.setParam(GRB.param.Method, 0)

# Create decision variables
```

```
x1 = model.addVar(lb=0.0, ub=GRB.INFINITY, obj=2.0, vtype=GRB.CONTINUOUS, name="x1") #
    arguments by name
x2 = model.addVar(0.0, GRB.INFINITY, 4.0, GRB.CONTINUOUS, "x2") # arguments by position
x3 = model.addVar(name="x3",obj=-1) # arguments by deafult

# The objective is to maximize (this is redundant now, but it will overwrite Var
    declaration
model.setObjective(2*x1 + 4*x2 -1*x3, GRB.MAXIMIZE)

# Add constraints to the model
model.addConstr(2*x1 -x3 <= 6, "c1")
model.addConstr(3*x2 -x3, GRB.LESS_EQUAL, 9.0, "c2")
model.addConstr(x1+x2, GRB.LESS_EQUAL, 4.0, "c3")

# Solve
model.optimize()

# Let's print the solution
for v in model.getVars():
    print v.varName, v.x

for c in model.getConstrs():
    print c.ConstrName, c.pi
```

```
Changed value of parameter Method to 0
   Prev: -1  Min: -1  Max: 4  Default: -1
Optimize a model with 3 rows, 3 columns and 6 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 3e+00]
  Objective range [1e+00, 4e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [4e+00, 9e+00]
Presolve time: 0.01s
Presolved: 3 rows, 3 columns, 6 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0   -0.0000000e+00   0.000000e+00   6.000000e+00      0s
       2    1.4000000e+01   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.01 seconds
Optimal objective  1.400000000e+01
x1 1.0
x2 3.0
x3 0.0
c1 0.0
c2 0.666666666667
c3 2.0
```

## Exercise 2* Simplex method

Solve the following LP problem carrying out the simplex operations by hand:

$$\begin{aligned}
\text{maximize} \quad & 5x_1 + 4x_2 + 3x_3 \\
\text{subject to} \quad & 2x_1 + 3x_2 + x_3 \leq 5 \\
& 4x_1 + x_2 + 2x_3 \leq 11 \\
& 3x_1 + 4x_2 + 2x_3 \leq 8 \\
& x_1, x_2, x_3 \geq 0
\end{aligned}$$

You are free to use any of the two representations, tableau or dictionary.
You can also get help from Python. You find a tutorial for what you need at this link:

http://www.imada.sdu.dk/~marco/DM545/Resources/Ipython/Tutorial4Exam.html.

**Solution:**

```
%run utils
A=array([[2,f(3,1),1,1,0,0,0,5],[4,1,2,0,1,0,0,11],[3,4,2,0,0,1,0,8],[5,4,3,0,0,0,1,0]])
tableau(A)
# enough that one is a fraction to make all matrix of type fraction
# First simplex iteration
A[0,:] = f(1,2)*A[0,:]
A[1,:] = A[1,:]-f(4,1)*A[0,:]
A[2,:] = A[2,:]-f(3,1)*A[0,:]
A[3,:] = A[3,:]-f(5,1)*A[0,:]
tableau(A)
```

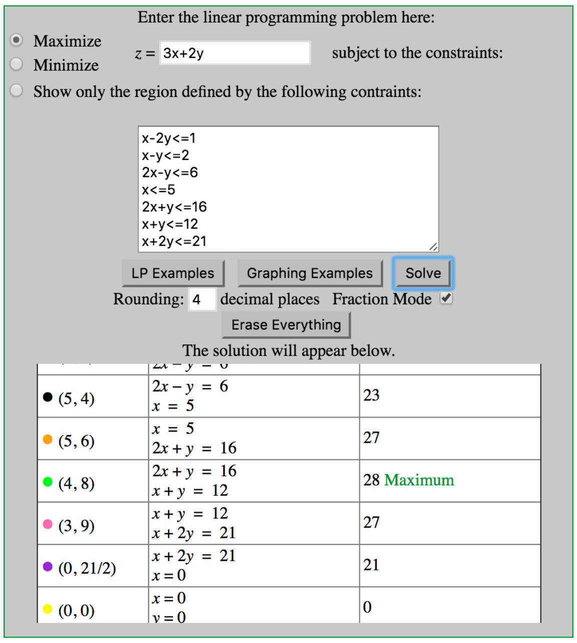
```
|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      2 |      3 |      1 |      1 |      0 |      0 |      0 |      5 |
|      4 |      1 |      2 |      0 |      1 |      0 |      0 |     11 |
|      3 |      4 |      2 |      0 |      0 |      1 |      0 |      8 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      5 |      4 |      3 |      0 |      0 |      0 |      1 |      0 |
|--------+--------+--------+--------+--------+--------+--------+--------+
pivot column: 1
pivot row: 1
pivot: 2


|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |    3/2 |    1/2 |    1/2 |      0 |      0 |      0 |    5/2 |
|      0 |     -5 |      0 |     -2 |      1 |      0 |      0 |      1 |
|      0 |   -1/2 |    1/2 |   -3/2 |      0 |      1 |      0 |    1/2 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |   -7/2 |    1/2 |   -5/2 |      0 |      0 |      1 |  -25/2 |
|--------+--------+--------+--------+--------+--------+--------+--------+


pivot column: 3
pivot row: 3
pivot: 1/2
|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      2 |      0 |      2 |      0 |     -1 |      0 |      2 |
|      0 |     -5 |      0 |     -2 |      1 |      0 |      0 |      1 |
|      0 |     -1 |      1 |     -3 |      0 |      2 |      0 |      1 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |     -3 |      0 |     -1 |      0 |     -1 |      1 |    -13 |
|--------+--------+--------+--------+--------+--------+--------+--------+
```

## Exercise 3

Solve the following linear programming problem applying the simplex algorithm:



Enter the linear programming problem here:

[✓ Maximize] Minimize $z =$ 3x+2y subject to the constraints:

Show only the region defined by the following contraints:

```
x-2y<=1
x-y<=2
2x-y<=6
x<=5
2x+y<=16
x+y<=12
x+2y<=21
```

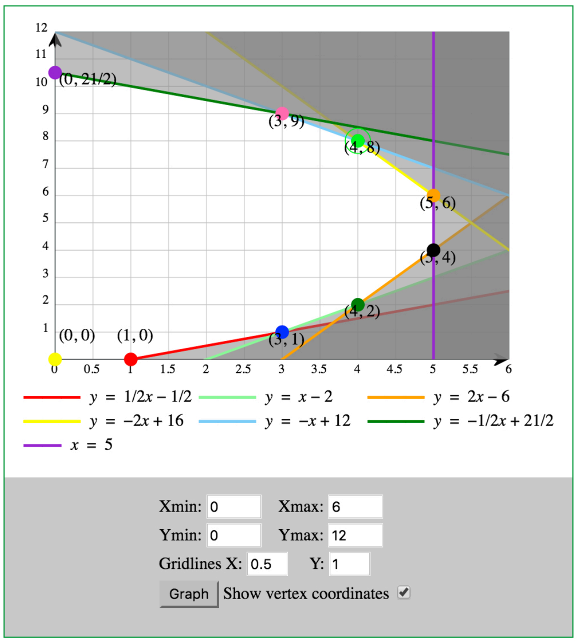
LP Examples | Graphing Examples | Solve

Rounding: 4 decimal places Fraction Mode ✓

Erase Everything

The solution will appear below.

| | | |
|---|---|---|
| ● (5, 4) | $2x - y = 6$ $x = 5$ | 23 |
| ● (5, 6) | $x = 5$ $2x + y = 16$ | 27 |
| ● (4, 8) | $2x + y = 16$ $x + y = 12$ | 28 Maximum |
| ● (3, 9) | $x + y = 12$ $x + 2y = 21$ | 27 |
| ● (0, 21/2) | $x + 2y = 21$ $x = 0$ | 21 |
| ● (0, 0) | $x = 0$ $y = 0$ | 0 |



Xmin: 0 Xmax: 6

Ymin: 0 Ymax: 12

Gridlines X: 0.5 Y: 1

Graph | Show vertex coordinates ✓

$$
\begin{array}{ll}
\text{maximize} & 3x_1 + 2x_2 \\
\text{subject to} & x_1 - 2x_2 \leq 1 \\
& x_1 - x_2 \leq 2 \\
& 2x_1 - x_2 \leq 6 \\
& x_1 \leq 5 \\
& 2x_1 + x_2 \leq 16 \\
& x_1 + x_2 \leq 12 \\
& x_1 + 2x_2 \leq 21 \\
& x_2 \leq 10 \\
& x_1, x_2 \geq 0.
\end{array}
$$

[Hint: you can plot the feasibility region with one of the tools linked at the course web page: "Tools" -> "Web applications on the simplex" -> "LP Simplex" and use the clairvoyant's rule to minimize the number of operations to carry out.]

**Solution:**

The initial solution of the simplex is at [0,0]. Then we can follow one of the two paths. The clairvoyant ruel says that we should choose the direction that minimizes the path. Thihs can be achieved by taking $x_2$. This yeilds a path of 3 arcs. Taking instead $x_1$ in the basis at the beginning would lead to a path of length 6.

## Exercise 4*

Consider the following problem:

$$
\begin{array}{rl}
\max & z = 4x_2 \\
\text{s.t.} & 2x_2 \geq 0 \\
& -3x_1 + 4x_2 \geq 1 \\
& x_1, x_2 \geq 0
\end{array}
$$

a. Write the LP in standard form (that is, in equation form with slack or surplus variables) and say why it does not provide immediately an initial feasible basis for the simplex method.

**Solution:**

In the equational standard form we have a negative $b$ term. The implication of this is that the initial solution of the simplex is infesible because $x_B \ngeq 0$. If we try to make the term positive we end up not having an identity matrix in the tableau.

## Exercise 5

Solve the following problem, known as the Klee-Minty problem, using the largest coefficient pivoting rule.

$$\begin{array}{ll} \text{maximize} & 100x_1 + 10x_2 + x_3 \\ \text{subject to} & x_1 \leq 1 \\ & 20x_1 + x_2 \leq 100 \\ & 200x_1 + 20x_2 + x_3 \leq 10000 \\ & x_1, x_2 \geq 0 \end{array}$$

Can you generalize the example to $n$ variables and guess what will be the number of iterations the simplex will do?

**Solution:**

```
|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|     20 |      1 |      0 |      0 |      1 |      0 |      0 |    100 |
|    200 |     20 |      1 |      0 |      0 |      1 |      0 |  10000 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|    100 |     10 |      1 |      0 |      0 |      0 |      1 |      0 |
|--------+--------+--------+--------+--------+--------+--------+--------+
PRIMAL SIMPLEX

pivot column: 1
pivot row: 1
 pivot: 1
1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|      0 |      1 |      0 |    -20 |      1 |      0 |      0 |     80 |
|      0 |     20 |      1 |   -200 |      0 |      1 |      0 |   9800 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |     10 |      1 |   -100 |      0 |      0 |      1 |   -100 |
|--------+--------+--------+--------+--------+--------+--------+--------+

pivot column: 2
pivot row: 2
 pivot: 1
1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|      0 |      1 |      0 |    -20 |      1 |      0 |      0 |     80 |
|      0 |      0 |      1 |    200 |    -20 |      1 |      0 |   8200 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |      0 |      1 |    100 |    -10 |      0 |      1 |   -900 |
|--------+--------+--------+--------+--------+--------+--------+--------+

pivot column: 4
pivot row: 1
 pivot: 1
```

```
1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|     20 |      1 |      0 |      0 |      1 |      0 |      0 |    100 |
|   -200 |      0 |      1 |      0 |    -20 |      1 |      0 |   8000 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|   -100 |      0 |      1 |      0 |    -10 |      0 |      1 |  -1000 |
|--------+--------+--------+--------+--------+--------+--------+--------+

pivot column: 3
pivot row: 3
 pivot: 1
1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|     20 |      1 |      0 |      0 |      1 |      0 |      0 |    100 |
|   -200 |      0 |      1 |      0 |    -20 |      1 |      0 |   8000 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|    100 |      0 |      0 |      0 |     10 |     -1 |      1 |  -9000 |
|--------+--------+--------+--------+--------+--------+--------+--------+

pivot column: 1
pivot row: 1
 pivot: 1
1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|      0 |      1 |      0 |    -20 |      1 |      0 |      0 |     80 |
|      0 |      0 |      1 |    200 |    -20 |      1 |      0 |   8200 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |      0 |      0 |   -100 |     10 |     -1 |      1 |  -9100 |
|--------+--------+--------+--------+--------+--------+--------+--------+

pivot column: 5
pivot row: 2
 pivot: 1
1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|      0 |      1 |      0 |    -20 |      1 |      0 |      0 |     80 |
|      0 |     20 |      1 |   -200 |      0 |      1 |      0 |   9800 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      0 |    -10 |      0 |    100 |      0 |     -1 |      1 |  -9900 |
|--------+--------+--------+--------+--------+--------+--------+--------+

pivot column: 4
```

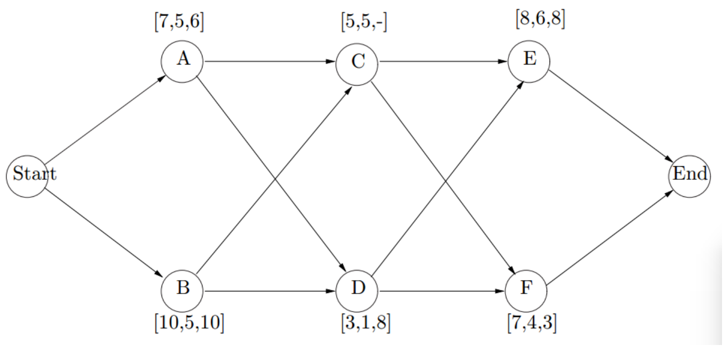


Figure 1: A network with activities on nodes for a small project with 6 activities. For each activity the following data is given in that order from left to right: normal time, minimum time in weeks, and the cost of shortening the duration of the activity by one week.

```
pivot row: 1
 pivot: 1
1

|--------+--------+--------+--------+--------+--------+--------+--------+
|     x1 |     x2 |     x3 |     x4 |     x5 |     x6 |     -z |      b |
|--------+--------+--------+--------+--------+--------+--------+--------+
|      1 |      0 |      0 |      1 |      0 |      0 |      0 |      1 |
|     20 |      1 |      0 |      0 |      1 |      0 |      0 |    100 |
|    200 |     20 |      1 |      0 |      0 |      1 |      0 |  10000 |
|--------+--------+--------+--------+--------+--------+--------+--------+
|   -100 |    -10 |      0 |      0 |      0 |     -1 |      1 | -10000 |
|--------+--------+--------+--------+--------+--------+--------+--------+
```

## Exercise 6* Project Scheduling

[This exercise is a part of one that appeared in Exam 2011] A small project has 6 sub-activities A, B, C, D, E, F whose individual dependency (shown by the immediate predecessors) is given in Figure 1. Here we also list the normal time (in weeks), the absolute minimum time and the cost of shortening the activity by one week.
The goal is to shorten the duration of the project to 19 weeks. This means that the duration of one or more activities has to be shortened. Of course we want to select these so that the total cost of shortening the duration to 19 weeks is minimized. Formulate this problem as a linear programming problem and argue that the optimal solution to this LP will provide the correct answer. Note that you must use the actual data in the LP formulation!

**Solution:**

We will use a variable $x_i$ to indicate how much we will shorten activity $i$ and another set of variables $y_i$ which will indicate the earliest starting time of activity $i$. For each arc $i \to j$ in the project network we will add the constraint $y_j \leq y_i + (d_i - x_i)$. We also use a variable yend to express that the dummy activity "end" cannot start before all its immediate predecessors have finished. Finally we add the constraint $y_{end} \leq 19$ to force the total project time to be less or equal than 19.

$$\begin{aligned}
\min\ & 6x_A + 10x_B + 8x_D + 8x_E + 3x_F \\
\text{subject to } & y_C \geq y_A + (7 - x_A) \\
& y_C \geq y_B + (10 - x_B) \\
& y_D \geq y_A + (7 - x_B) \\
& y_D \geq y_B + (10 - x_B) \\
& y_E \geq y_C + 5 \\
& y_E \geq y_D + (3 - x_D) \\
& y_F \geq y_C + 5 \\
& y_F \geq y_D + (3 - x_D) \\
& y_{end} \geq y_E + (8 - x_E) \\
& y_{end} \geq y_F + (7 - x_F) \\
& y_{end} \geq 19 \\
& x_A \leq 2 \\
& x_B \leq 5 \\
& x_C \leq 2 \\
& x_D \leq 2 \\
& x_E \leq 3 \\
& x_F \leq 2 \\
& x_A, x_B, x_C, x_D, x_E, x_F \geq 0 \\
& y_A, y_B, y_C, y_D, y_E, y_F, y_{end} \geq 0
\end{aligned}$$

The optimal solution to this LP will tell us to shorten activity $i$ by $x_i \geq 0$ units and since the cost we apply to each $x_i$ is the per unit shortening cost of that activity, the cost of the solution will be that of shortening the project in the way suggested by the $x_i$'s. Conversely, any feasible shortening of projects corresponds to a solution to this LP whose cost (in the LP) is the actual cost of shortening the activities in the way suggested.

## Exercise 7 The pooling problem

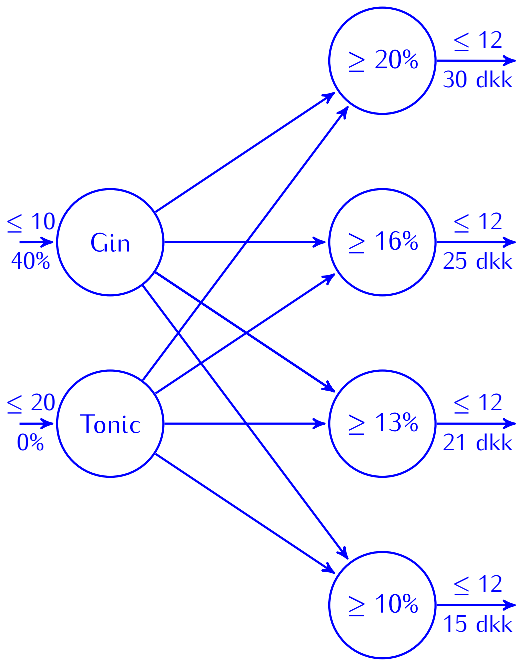
[This exercise appeared in Exam 2013] A bartender serves usually alcoholic beverages behind the bar. A bartender can generally mix classic cocktails such as a Gin-and-Tonic, Caipirinha and Mojito. In order to achieve this task the bartender has to maintain the supplies and inventory for the bar.
End drinks for the customers are created by directly mixing the raw materials. Restricting our attention to the Gin-and-Tonic drink, the suggested ratios of gin and tonic are 1:1, 2:3, 1:2, and 1:3 (source Wikipedia). The historical data indicate nevertheless that the typical demand on a Saturday evening in the premise of our bartender is as follows:

| Alcohol | quantity | price per deciliter |
|---|---|---|
| $\geq 20\%$ | $\leq 12$ l | 30 dkk |
| $\geq 16\%$ | $\leq 12$ l | 25 dkk |
| $\geq 13\%$ | $\leq 12$ l | 21 dkk |
| $\geq 10\%$ | $\leq 12$ l | 15 dkk |

For example, the first row indicates that there are customers that are willing to intake 20% or more alcohol consuming all together up to 12 liters and willing to pay 30 krone per deciliter.
Our bartender for economical and logistic issues can buy up to 10 liters of gin that contain 40% alcohol and up to 20 liters of tonic that contain no alcohol.
The mixing process occurs in a way such that at the end the input products no longer exist in their original forms, but are mixed to form new mixed products with new property values, in this case, the percentage of alcohol content.
The objective is to maximize the total profit from the sell.

The problem is an example of blending problem that arises, beside bar keeping also in the oil refinery industry.
Represent the situation as a network and model the problem as a linear programming problem.

**Solution:**



Let $x_{ij}$ be the amount of product in liters that flows from $i$ to $j$. Let $I$ be the set of sources of gin and of tonic. Let $J$ be the set of four customer categories.

$$\max \sum_{j\in J}\sum_{i\in I} r_j x_{ij} \tag{1}$$

$$\sum_{j\in J} x_{ij} \leq b_i \qquad \forall i \in I \tag{2}$$

$$\sum_{i\in I} x_{ij} \leq d_j \qquad \forall j \in J \tag{3}$$

$$\sum_{i\in I} p_i x_{ij} \geq p_j \sum_{i\in I} x_{ij} \qquad \forall j \in J \tag{4}$$

$$x_{ij} \geq 0 \qquad \forall i \in I, j \in J \tag{5}$$

```
set INPUT;
set OUTPUT;

param revenue {OUTPUT} > 0;
param in_max {INPUT} >= 0;
param out_max {OUTPUT} >=0;

param q_in {INPUT} >= 0;
param q_out_min {OUTPUT} >= 0;

var x {INPUT, OUTPUT} >= 0;

# changed to negative to output mps solvable
maximize profit: sum{i in INPUT, j in OUTPUT} -revenue[j]*x[i,j];
```

```
subject to ins{i in INPUT}:
  sum{j in OUTPUT} x[i,j]<=in_max[i];

subject to outs{j in OUTPUT}:
  sum{i in INPUT} x[i,j]<=out_max[j];

subject to quality{j in OUTPUT}:
  sum{i in INPUT} q_in[i]*x[i,j]==q_out_min[j]*sum{i in INPUT} x[i,j];
```

```
set INPUT := a b;
set OUTPUT := c d e f;

param: revenue out_max q_out_min :=
  c 15 12 0.1
  d 21 12 0.13
  e 25 12 0.16
  f 30 12 0.2 ;

param: in_max q_in :=
  a 10 0.4
  b 20 0 ;
```

```
model pooling.mod
data pooling.dat

option solver cplex;
option cplex_options 'timing=1';
option presolve 0;
option show_stats 1;


problem pooling: x, profit, ins, outs, quality;


expand pooling;
write mpooling;

solve;
display x;
printf{i in INPUT} "%s %d\n", i, sum{j in OUTPUT} x[i,j];
printf{j in OUTPUT} "%s %d\n", j, sum{i in INPUT} x[i,j];
```

```
maximize profit:
        15*x['a','c'] + 21*x['a','d'] + 25*x['a','e'] + 30*x['a','f'] +
        15*x['b','c'] + 21*x['b','d'] + 25*x['b','e'] + 30*x['b','f'];

subject to ins['a']:
        x['a','c'] + x['a','d'] + x['a','e'] + x['a','f'] <= 10;

subject to ins['b']:
        x['b','c'] + x['b','d'] + x['b','e'] + x['b','f'] <= 20;

subject to outs['c']:
        x['a','c'] + x['b','c'] <= 12;

subject to outs['d']:
        x['a','d'] + x['b','d'] <= 12;
```

```
subject to outs['e']:
        x['a','e'] + x['b','e'] <= 12;

subject to outs['f']:
        x['a','f'] + x['b','f'] <= 12;

subject to quality['c']:
        0.3*x['a','c'] - 0.1*x['b','c'] >= 0;

subject to quality['d']:
        0.27*x['a','d'] - 0.13*x['b','d'] >= 0;

subject to quality['e']:
        0.24*x['a','e'] - 0.16*x['b','e'] >= 0;

subject to quality['f']:
        0.2*x['a','f'] - 0.2*x['b','f'] >= 0;


8 variables, all linear
10 constraints, all linear; 24 nonzeros
        10 inequality constraints
1 linear objective; 8 nonzeros.

CPLEX 12.6.0.0: timing=1

Times (seconds):
Input =  0.004
Solve =  0
Output = 0
CPLEX 12.6.0.0: optimal solution; objective 630
10 dual simplex iterations (2 in phase I)
x :=
a c   0
a d   3.9
a e   4.8
a f   1.3
b c   0
b d   8.1
b e   7.2
b f   1.3
;
a 10
b 16
c 0
d 12
e 12
f 2
```