Operations Research

Midterm Exam

October 27, 2017

1. Consider linear programming problem

$$\text{maximize } z = 3\,x_1 + 2\,x_2 + x_3$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &\le 2\,, \\ 3\,x_1 + x_2 &\le 1\,, \\ x_1, x_2, x_3 &\ge 0\,. \end{aligned}$$

   (a) Write out the *dual* problem to this LP.

   (b) Solve the dual problem using the graphical method.

   (c) Which constraints in the dual formulation are binding?

   (d) Which variables in the primal formulation are basic?

   (e) Using your answer to (d), verify your solution by a direct computation on the *primal* problem.

(5+10+3+3+4)

2. Consider the linear programming problem

$$\text{maximize } z = 2\,x_1 + x_2$$

subject to

$$\begin{aligned} -x_1 + x_2 &\le 1\,, \\ x_1 - 2\,x_2 &\le 2\,, \\ x_1, x_2 &\ge 0\,. \end{aligned}$$

Determine the solvability of this problem and find the solution, if it exists, *using the simplex method.* (20)

3. You are tasked with setting up an image processing pipeline for a medical laboratory. Images come from a database in LIFF, a format specifically designed for microscope image processing. The lab would like to make these images available in PNG format via their web server. Unfortunately, you do not have an LIFF-to-PNG converter. However, you have access to a large set of image converters which take the following average times (in milliseconds) to convert an image. Missing format converters are indicated by a dash.

Source Format	Destination Format				
	LIFF	**PNG**	**JPG**	**BPG**	**BMP**
LIFF	–	–	800	700	300
PNG	500	–	600	–	–
JPG	–	900	–	700	–
BPG	700	300	100	–	600
BMP	–	600	200	200	–

Find the most efficient way to realize the conversion from LIFF to PNG. Illustrate your solution by drawing an appropriate network representation of the problem.

(20)

4. The Pyomo program on the last page is showing a min-cost flow problem similar to "Distribution Unlimited" Example from Hillier and Lieberman.

   (a) Draw the network that is described by the data in the code.

   (b) Management is considering investing in expanding the transportation capacities on any of the currently capacity-constrained routes. Which route should be invested in first, assuming that it costs the same on all routes to per unit increase of the maximal capacity, and that the unit cost of transportation will not change?

   (c) The route from the distribution center DC to warehouse W1 is cut due to road damage. Is it possible to still supply the warehouses? If not, how many units can you still deliver to the warehouses? Which problem did you solve to find out?

   (d) One truck can be loaded with 10 units of the product. Thus, it is desirable that the optimal solution will be in multiples of 10. Which property of the data ensures that this is possible?

(10+5+5+5)

In [1]:
```
from pyomo.environ import *
from pyomo.opt import *
opt = solvers.SolverFactory("glpk")
```

In [2]:
```
b = {'F1':50,
     'F2':40,
     'DC':0,
     'W1':-30,
     'W2':-60}

C = {('F1','F2'):200,
     ('F1','DC'):400,
     ('F1','W1'):100,
     ('F2','DC'):300,
     ('DC','W1'):100,
     ('DC','W2'):200,
     ('W1','W2'):200,
     ('W2','W1'):100}

U = {('DC','W2'):40,
     ('F1','W1'):20,
     ('W2','W1'):20}

N = list(b.keys())
A = list(C.keys())
V = list(U.keys())
```

In [3]:
```
model = ConcreteModel()
model.f = Var(A, within=NonNegativeReals)

def flow_rule(model, n):
    InFlow  = sum(model.f[i,j] for (i,j) in A if j==n)
    OutFlow = sum(model.f[i,j] for (i,j) in A if i==n)
    return InFlow + b[n] == OutFlow

model.flow = Constraint(N, rule=flow_rule)

def capacity_rule(model, i, j):
    return U[i,j] >= model.f[i,j]

model.capacity = Constraint(V, rule=capacity_rule)

model.cost = Objective(expr = sum(model.f[a]*C[a] for a in A), sense=minimize
)
```

In [4]:
```
model.dual = Suffix(direction=Suffix.IMPORT)
results = opt.solve(model)
model.f.get_values()
```

Out[4]:
```
{('DC', 'W1'): 30.0,
 ('DC', 'W2'): 40.0,
 ('F1', 'DC'): 30.0,
 ('F1', 'F2'): 0.0,
 ('F1', 'W1'): 20.0,
 ('F2', 'DC'): 40.0,
 ('W1', 'W2'): 20.0,
 ('W2', 'W1'): 0.0}
```

In [5]:
```
for (i,j) in V:
    print ((i,j), model.dual[model.capacity[(i,j)]])
```

```
('DC', 'W2') -100.0
('F1', 'W1') -400.0
('W2', 'W1') 0.0
```