Benjamin Tate
5/21/2017
CS 325 -- Section 400
Homework #4

1.

a. Input:

```
max dc
ST
    dg = 0
    da - df <= 5
    da - dh <= 4
    db - da <= 8
    db - df <= 7
    db - dh <= 9
    dc - db <= 4
    dc - df <= 3
    dd - dc <= 3
    dd - de <= 9
    dd - dg <= 2
    de - db <= 10
    de - dd <= 25
    de - df <= 2
    df - da <= 10
    df - dd <= 18
    dg - de <= 7
    dh - dg <= 3
END
```

Output:

```
LP OPTIMUM FOUND AT STEP      6

       OBJECTIVE FUNCTION VALUE

       1)      16.00000

 VARIABLE        VALUE          REDUCED COST
       DC        16.000000          0.000000
       DG         0.000000          0.000000
       DA         4.000000          0.000000
       DF        13.000000          0.000000
       DH         3.000000          0.000000
       DB        12.000000          0.000000
       DD         0.000000          0.000000
       DE         0.000000          0.000000


      ROW   SLACK OR SURPLUS     DUAL PRICES
       2)         0.000000          1.000000
       3)        14.000000          0.000000
       4)         3.000000          0.000000
       5)         0.000000          0.000000
       6)         8.000000          0.000000
       7)         0.000000          1.000000
       8)         0.000000          1.000000
       9)         0.000000          0.000000
      10)        19.000000          0.000000
      11)         9.000000          0.000000
      12)         2.000000          0.000000
      13)        22.000000          0.000000
      14)        25.000000          0.000000
      15)        15.000000          0.000000
      16)         1.000000          0.000000
      17)         5.000000          0.000000
      18)         7.000000          0.000000
      19)         0.000000          1.000000

 NO. ITERATIONS=       6
```

Shortest path distance G → C: **16**

b. Input:

```
max da + db + dc + dd + de + df + dg + dh
ST
    dg = 0
    da - df <= 5
    da - dh <= 4
    db - da <= 8
    db - df <= 7
    db - dh <= 9
    dc - db <= 4
    dc - df <= 3
    dd - dc <= 3
    dd - de <= 9
    dd - dg <= 2
    de - db <= 10
    de - dd <= 25
    de - df <= 2
    df - da <= 10
    df - dd <= 18
    dg - de <= 7
    dh - dg <= 3
END
```

Output:

```
LP OPTIMUM FOUND AT STEP      5

       OBJECTIVE FUNCTION VALUE

       1)      76.00000

  VARIABLE        VALUE          REDUCED COST
        DA         7.000000          0.000000
        DB        12.000000          0.000000
        DC        16.000000          0.000000
        DD         2.000000          0.000000
        DE        19.000000          0.000000
        DF        17.000000          0.000000
        DG         0.000000          0.000000
        DH         3.000000          0.000000

       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         0.000000          8.000000
        3)        15.000000          0.000000
        4)         0.000000          3.000000
        5)         3.000000          0.000000
        6)        12.000000          0.000000
        7)         0.000000          2.000000
        8)         0.000000          1.000000
        9)         4.000000          0.000000
       10)        17.000000          0.000000
       11)        26.000000          0.000000
       12)         0.000000          1.000000
       13)         3.000000          0.000000
       14)         8.000000          0.000000
       15)         0.000000          1.000000
       16)         0.000000          2.000000
       17)         3.000000          0.000000
       18)        26.000000          0.000000
       19)         0.000000          6.000000

 NO. ITERATIONS=       5
```

Shortest path distance G → A: **7**
Shortest path distance G → B: **12**
Shortest path distance G → D: **2**
Shortest path distance G → E: **19**
Shortest path distance G → F: **17**
Shortest path distance G → G: **0**
Shortest path distance G → H: **3**

2.

a. S, P, B_1, B_2 = # of silk, polyester, blend 1, and blend 2 ties to produce per month

Profits per tie:

Silk = 6.70 - .75 - 20*.125 = 3.45
Poly = 3.55 - .75 - 6*.08 = 2.32
B1 = 4.31 - .75 - 6*.05 - 9*.05 = 2.81
B2 = 4.81 - .75 - 6*.03 - 9*.07 = 3.25

Max $3.45S + 2.32P + 2.81B_1 + 3.25B_2$

ST
$.125S \le 1000$
$.08P + .05B_1 + .03B_2 \le 2000$
$.05B_1 + .07B_2 \le 1250$
$6000 \le S \le 7000$
$10000 \le P \le 14000$
$13000 \le B_1 \le 16000$
$6000 \le B_2 \le 8500$
$S, P, B_1, B_2 \ge 0$

Converting for use with Matlab's linprog:

Min $-3.45S - 2.32P - 2.81B_1 - 3.25B_2$

ST
$.125S \le 1000$
$.08P + .05B_1 + .03B_2 \le 2000$
$.05B_1 + .07B_2 \le 1250$
$-S \le -6000$
$S \le 7000$
$-P \le -10000$
$P \le 14000$
$-B_1 \le -13000$

B_1 <= 16000
$-B_2$ <= -6000
B_2 <= 8500
$-S, -P, -B_1, -B_2$ <= 0

b. *(Using Matlab)*
Input commands:
f = [-3.45;-2.32;-2.81;-3.25]
A = [.125 0 0 0; 0 .08 .05 .03; 0 0 .05 .07; -1 0 0 0; 1 0 0 0; 0 -1 0 0; 0 1 0 0; 0 0 -1 0; 0 0 1 0; 0 0 0 -1; 0 0 0 1; -1 0 0 0; 0 -1 0 0; 0 0 -1 0; 0 0 0 -1]
b = [1000;2000;1250;-6000;7000;-10000;14000;-13000;16000;-6000;8500;0;0;0;0]
linprog(f,A,b)
Output:

```
>> x = linprog(f,A,b)
Warning: Your current settings will run a different algorithm ('dual-simplex') in
a future release.

> In linprog (line 204)
Optimization terminated.

x =

   1.0e+04 *

    0.7000
    1.3625
    1.3100
    0.8500
```

c. Optimal number of ties to maximize profit:
7,000 Silk, 13,625 Polyester, 13,100 Blend 1, and 8,500 Blend 2

3.

a. To turn this problem into a linear program, I will first consider our variables to be each possible route that a fridge can take from plant to retailer, where X_{ijk} is the number of refrigerators that are shipped from P_i to W_j, and finally from W_j to R_k. This means that, in total, there will be 36 variables. By adding the total cost of shipping along each route, we can obtain the following objective function:
min $15X_{111}+ 16X_{112}+ 17X_{113}+ 20X_{114}+ 27X_{123}+ 23X_{124}+ 25X_{125}+ 29X_{126}+ 16X_{211}+ 17X_{212}+ 18X_{213}+ 21X_{214}+ 20X_{223}+ 16X_{224}+ 18X_{225}+ 22X_{226}+ 18X_{311}+ 19X_{312}+ 20X_{313}+ 23X_{314}+ 20X_{323}+ 16X_{324}+ 18X_{325}+ 22X_{326}+ 23X_{334}+ 21X_{335}+ 21X_{336}+ 15X_{337}+ 26X_{423}+ 22X_{424}+ 24X_{425}+ 28X_{426}+ 22X_{434}+ 20X_{435}+ 20X_{436}+ 14X_{437}$
This program will then be subject to the constraints of supply for each plant and demand for each retailer, as well as nonnegativity constraints for each of the plants. Finally, there must be nonnegativity constraints for each of the 36 variables. In other words, adjusted for convenient use with Matlab's linprog funciton, the constraints are as follows:
$X_{111}+X_{112}+X_{113}+X_{114}+X_{123}+X_{124}+X_{125}+X_{126}<=150$
$X_{211}+X_{212}+X_{213}+X_{214}+X_{223}+X_{224}+X_{225}+X_{226}<=450$
$X_{311}+X_{312}+X_{313}+X_{314}+X_{323}+X_{324}+X_{325}+X_{326}+X_{334}+X_{335}+X_{336}+X_{337}<=250$
$X_{423}+X_{424}+X_{425}+X_{426}+X_{434}+X_{435}+X_{436}+X_{437}<=150$
$-X_{111}-X_{211}-X_{311}<=-100$
$-X_{112}-X_{212}-X_{312}<=-150$
$-X_{113}-X_{123}-X_{213}-X_{223}-X_{313}-X_{323}-X_{423}<=-100$
$-X_{114}-X_{124}-X_{214}-X_{224}-X_{314}-X_{324}-X_{334}-X_{424}-X_{434}<=-200$
$-X_{125}-X_{225}-X_{325}-X_{335}-X_{425}-X_{435}<=-200$
$-X_{126}-X_{226}-X_{326}-X_{336}-X_{426}-X_{436}<=-150$
$-X_{337}-X_{437}<=-100$
$-X_{111}-X_{112}-X_{113}-X_{114}-X_{123}-X_{124}-X_{125}-X_{126}<=0$
$-X_{211}-X_{212}-X_{213}-X_{214}-X_{223}-X_{224}-X_{225}-X_{226}<=0$
$-X_{311}-X_{312}-X_{313}-X_{314}-X_{323}-X_{324}-X_{325}-X_{326}-X_{334}-X_{335}-X_{336}-X_{337}<=0$
$-X_{423}-X_{424}-X_{425}-X_{426}-X_{434}-X_{435}-X_{436}-X_{437}<=0$

b. *(Using Matlab)*
Input commands:
f = [15;16;17;20;27;23;25;29;16;17;18;21;20;16;18;22;18;19; 20;23;20;16;18;22;23;21;21;15;26;22;24;28;22;20;20;14]
A = [1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;

```
        0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1;
        -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 -1 0 -1 0 0 0 0 0 -1 0 -1 0 0 0 0 0 -1 0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0;
        0 0 0 -1 0 -1 0 0 0 0 0 -1 0 -1 0 0 0 0 0 -1 0 -1 0 0 -1 0 0 0 0 -1 0 0 -1 0 0 0;
        0 0 0 0 0 0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 -1 0 0 -1 0 0 0 0 -1 0 0 -1 0 0;
        0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 -1 0 0 -1 0 0 0 0 -1 0 0 -1 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 -1;
        -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 -1 -1 -1 -1 -1 -1 -1;
        -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1 0;
        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 -1]
b =     [150;450;250;150;-100;-150;-100;-200;-200;-150;-100;
        0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0]
linprog(f,A,b)
```

Output:

```
>> x = linprog(f,A,b)
Warning: Your current settings will run a different algorithm ('dual-simplex') in
a future release.

> In linprog (line 204)
Optimization terminated.

x =

   42.9135
   75.5004
   31.5862
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
   57.0865
   74.4996
   68.4138
    0.0000
    0.0000
  130.1754
  119.8246
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
   69.8246
   80.1754
    0.0000
    0.0000
    0.0000
   60.5991
   39.4009
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
    0.0000
   89.4009
   60.5991
```

c. Optimal shipping routes:
From the above linear program, we get the following number of fridges to travel each full route from plant to retailer, where X_{ijk} represents the route from $P_i \rightarrow W_j \rightarrow R_k$:

Route	**# of fridges per week**
X_{111}	42.9135
X_{112}	75.5004
X_{113}	31.5862
X_{211}	57.0865
X_{212}	74.4996
X_{213}	68.4138
X_{224}	130.1754
X_{225}	119.8246

X_{324}	69.8246
X_{325}	80.1754
X_{336}	60.5991
X_{337}	39.4009
X_{436}	89.4009
X_{437}	60.5991

However, the results in the above table aren't very informative, since fridges don't go directly from a plant to a retailer but instead stop at a warehouse in the middle. To make more sense of it, we need to split the routes into their two separate parts (from plant to warehouse, and then from warehouse to retailer). We can do this by adding the total number of fridges that go from a given plant or warehouse to a given warehouse or retailer in the table above. For example: $Y_{11} = X_{111} + X_{112} + X_{113} = 42.9135 + 75.5004 + 31.5862$. Doing so gives the following results, where Y_{ij} represents the number of fridges shipped from P_i to W_j each week, and where Z_{jk} represents the number of fridges shipped from W_j to R_k each week:

Route	**# of fridges per week**
Y_{11}	150
Y_{21}	200
Y_{22}	250
Y_{32}	150
Y_{33}	100
Y_{43}	150
Z_{11}	100
Z_{12}	150
Z_{13}	100
Z_{24}	200
Z_{25}	200
Z_{36}	150
Z_{37}	100

<u>Total minimum cost</u> = $15X_{111} + 16X_{112} + 17X_{113} + 20X_{114} + 27X_{123} + 23X_{124} + 25X_{125} + 29X_{126} + 16X_{211} + 17X_{212} + 18X_{213} + 21X_{214} + 20X_{223} + 16X_{224} + 18X_{225} + 22X_{226} + 18X_{311} + 19X_{312} + 20X_{313} + 23X_{314} + 20X_{323} + 16X_{324} + 18X_{325} + 22X_{326} + 23X_{334} + 21X_{335} + 21X_{336} + 15X_{337} + 26X_{423} + 22X_{424} + 24X_{425} + 28X_{426} + 22X_{434} + 20X_{435} + 20X_{436} + 14X_{437}$
$= 15(42.9135) + 16(75.5004) + 17(31.5862) + 16(57.0865) + 17(74.4996) + 18(68.4138) + 16(130.1754) + 18(119.8246) + 16(69.8246) + 18(80.1754) + 21(60.5991) + 15(39.4009) + 20(89.4009) + 14(60.5991)$
≈ $17,100