Linear Programming

Homework 1

Nick Morris

2/22/16

Proposed Model

Formulation:

```
set N;    # set of all nodes (consumers, facilities, wastes)
set A := N cross N;    # set of arcs
set M;    # set of materials (product, waste)
set C;    # set of consumers
set F;    # set of facilities
set W;    # set of wastes

param net{N,M} default 0;    # net flow of material at a node
param d{(i,j) in A} default 0;    # arc distance
param cap{(i,j) in A} default 0;    # arc material flow capacity
param seta{(i,j) in A} := 500 + (0.30 * d[i,j]);    # cost to use (setup) an arc
param setf{F};    # cost to setup a facility
param setw{W};    # cost to setup a waste
param con{F};    # conversion constant (how much waste is produced for a unit of product)
param c{M};    # material travel unit cost

var openf{F} binary;    # a facility is/isnt setup
var openw{W} binary;    # a waste is/isnt setup
var opena{(i,j) in A} binary;    # an arc is/isnt setup
var x{(i,j) in A, M} >= 0;    # material flow on an arc

minimize Cost: sum{i in F}(setf[i] * openf[i]) + sum{i in W}(setw[i] * openw[i])
               + sum{(i,j) in A: i < 11 and j < 11}(seta[i,j] * opena[i,j])
               + sum{(i,j) in A, k in M}(c[k] * d[i,j] * x[i,j,k]);

s.t. Demand{i in C, k in M}: sum{(n,i) in A}(x[n,i,k]) - sum{(i,j) in A}(x[i,j,k])
                             = net[i,k];
s.t. Production{i in F, k in M}: sum{(n,i) in A}(x[n,i,k]) - sum{(i,j) in A}(x[i,j,k])
                                 >= net[i,k] * openf[i];
s.t. Disposal{i in W, k in M}: sum{(n,i) in A}(x[n,i,k]) - sum{(i,j) in A}(x[i,j,k])
                               <= net[i,k] * openw[i];
s.t. Capacity{(i,j) in A}: sum{k in M}(x[i,j,k]) <= cap[i,j] * opena[i,j];
s.t. Conversion{i in F}: sum{(i,j) in A}(x[i,j,2]) = sum{(i,j) in A}(con[i] * x[i,j,1]);
```

Description:

Sets:

There are 6 total sets to represent all consumers, pathways, facilities, waste disposals, and materials involved in this decision process. It is important to note that there are unique nodes for each consumer, facility, and waste disposal. There is an arc between each relevant consumer node and facility node pair, with a capacity equivalent to the facility production capacity, and with a distance of zero, to represent that the facility and consumer are sharing the same location but are two separate objects. The same arc design corresponds to each relevant consumer node and waste node pairing as well.

Parameters:

All of the parameters described in the problem statement are all included. It is important to note that the "net" parameter takes on negative and positive values to represent how much a consumer node demands (positive value for product, zero value for waste), how much a facility node can supply (negative values for product and waste), and how much a waste node can receive (zero value for product, positive value for waste).

Variables:

There are four variables, three of which are binary to represent the three decisions on whether or not to open a facility, to open a waste disposal, and to use an arc. The fourth variable is a non-negative real to represent how much material flow (product or waste) is on an arc, and in which direction on the arc.

Objective Function:

The objective function is a cost function with four components: the total facility setup cost, the total waste disposal setup cost, the arc usage cost, and the arc transportation cost. The third component in the objective function states that there is no arc usage cost for any arc connected to a facility node or waste node. This is because these arcs are artificial, and don't actually exist with regards to cost.

Constraints:

The first constraint "Demand" is an equality constraint stating that the total inflow at a node, subtracted by the total outflow at a node, must be equivalent to the net flow value (ie. demand) for every consumer node and every material type.

The second constraint "Production" is an inequality constraint stating that the total inflow at a node, subtracted by the total outflow at a node, must be no smaller than the net flow value (ie. production capacity) for every facility node and every material type. The net flow value for facilities is negative for each material type, therefore if the material flow variables "`var x[i, j, k]`" took on values that were cumulatively smaller than the net flow value, then this would represent a facility giving more than its production capacity allows.

The third constraint "Disposal" is an inequality constraint stating that the total inflow at a node, subtracted by the total outflow at a node, must be no larger than the net flow value (ie. waste capacity) for every waste node and every material type. The net flow value for waste disposals is positive for waste material, therefore if the material flow variables "`var x[i, j, k]`" took on values that were cumulatively larger than the net flow value, then this would represent a waste disposal accepting more waste than its capacity allows.

The fourth constraint "Capacity" is an inequality constraint stating that the cumulative material flow on an arc must be no larger than its capacity, if the arc is decided to be used, for every arc.

The fifth constraint "Conversion" is an equality constraint stating that the total outflow of waste material must be proportional to the total outflow of product material, via the "`param con[i]`" value for every facility.

AMPL Output:

```
ampl: include network.run;
CPLEX 12.6.3.0: optimal integer solution; objective 17663085.65
678 MIP simplex iterations
0 branch-and-bound nodes
No basis.
x [*,*,1]
:      1     2         3         4          5         6     7     8     9   10 :=
1      0     0         0         0          0        70     0     0     0    0
2      0     0         0       49.9348      0         0     0     0     0    0
3    150     0         0       43.6957      0       150   100     0     0    0
4      0     0         0         0          0         0     0    20     0    0
5      0   149.935     0         1.36957    0         0     0   150     0    0
6      0     0         0         0          0         0     0     0   110    0
7      0     0         0         0          0         0     0     0     0    0
8      0     0         0         0          0         0     0     0     0   85
9      0     0         0         0          0         0     0     0     0   20
10     0     0         0         0          0         0     0     0     0    0
11     0     0       563.696     0          0         0     0     0     0    0
12     0     0         0         0        391.304     0     0     0     0    0
13     0     0         0         0          0         0     0     0     0    0
14     0     0         0         0          0         0     0     0     0    0
15     0     0         0         0          0         0     0     0     0    0
16     0     0         0         0          0         0     0     0     0    0
17     0     0         0         0          0         0     0     0     0    0
18     0     0         0         0          0         0     0     0     0    0

:   11  12  13  14  15  16  17  18    :=
1    0   0   0   0   0   0   0   0
2    0   0   0   0   0   0   0   0
3    0   0   0   0   0   0   0   0
4    0   0   0   0   0   0   0   0
```

```
5    0   0   0   0   0   0   0   0
6    0   0   0   0   0   0   0   0
7    0   0   0   0   0   0   0   0
8    0   0   0   0   0   0   0   0
9    0   0   0   0   0   0   0   0
10   0   0   0   0   0   0   0   0
11   0   0   0   0   0   0   0   0
12   0   0   0   0   0   0   0   0
13   0   0   0   0   0   0   0   0
14   0   0   0   0   0   0   0   0
15   0   0   0   0   0   0   0   0
16   0   0   0   0   0   0   0   0
17   0   0   0   0   0   0   0   0
18   0   0   0   0   0   0   0   0

 [*,*,2]
:    1           2           3         4          5         6           7    8 :=
1    0           0           0         0          0         0.0652174   0    0
2    0.0652174   0           0         0          0         0           0    0
3    0           0           0         6.36957    0         0          50    0
4    0           0           0         0          0         0           0   65
5    0           0.0652174   0        58.6304     0         0           0    0
6    0           0           0         0          0         0           0    0
7    0           0           0         0          0         0           0    0
8    0           0           0         0          0         0           0    0
9    0           0           0         0          0         0           0    0
10   0           0           0         0          0         0           0    0
11   0           0          56.3696    0          0         0           0    0
12   0           0           0         0         58.6957    0           0    0
13   0           0           0         0          0         0           0    0
14   0           0           0         0          0         0           0    0
15   0           0           0         0          0         0           0    0
16   0           0           0         0          0         0           0    0
17   0           0           0         0          0         0           0    0
18   0           0           0         0          0         0           0    0

:    9           10          11  12  13  14  15  16  17    18       :=
1    0            0           0   0   0   0   0   0   0     0
2    0            0           0   0   0   0   0   0   0     0
3    0            0           0   0   0   0   0   0   0     0
4    0            0           0   0   0   0   0   0   0     0
5    0            0           0   0   0   0   0   0   0     0
6    0.0652174    0           0   0   0   0   0   0   0     0
7    0           50           0   0   0   0   0   0   0     0
8    0           65           0   0   0   0   0   0   0     0
9    0            0.0652174   0   0   0   0   0   0   0     0
10   0            0           0   0   0   0   0   0   0   115.065
11   0            0           0   0   0   0   0   0   0     0
```

```
12   0           0          0   0   0   0   0   0   0     0
13   0           0          0   0   0   0   0   0   0     0
14   0           0          0   0   0   0   0   0   0     0
15   0           0          0   0   0   0   0   0   0     0
16   0           0          0   0   0   0   0   0   0     0
17   0           0          0   0   0   0   0   0   0     0
18   0           0          0   0   0   0   0   0   0     0

:  openf openw    :=
11    1     .
12    1     .
13    0     .
14    0     .
15    0     .
16    .     0
17    .     0
18    .     1

opena [*,*]
:    1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18   :=
1    0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
2    1   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0
3    1   0   0   1   0   1   1   0   0   0   0   0   0   0   0   0   0   0
4    0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   1   0   0
5    0   1   0   1   0   0   0   1   0   0   0   0   0   0   0   0   0   0
6    0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0
7    0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
8    0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0
9    0   0   0   0   0   0   0   0   0   1   0   0   0   0   0   0   1   0
10   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1
11   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
12   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0   0
13   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0   0
14   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0   0
15   0   0   0   0   0   0   0   1   0   0   0   0   0   0   0   0   0   0
16   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
17   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
18   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
```

Appendix

Data File:

```
set N :=
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18;

set M :=
1
2;

set C :=
1
2
3
4
5
6
7
8
9
10;

set F :=
11
12
13
14
15;
```

```
set W :=
16
17
18;

param net :=
1      1      80
2      1      100
3      1      120
4      1      75
5      1      90
6      1      110
7      1      100
8      1      85
9      1      90
10     1      105
11     1      -650
12     1      -450
13     1      -800
14     1      -350
15     1      -500
11     2      -65
12     2      -67.5
13     2      -160
14     2      -43.75
15     2      -87.5
16     2      100
17     2      140
18     2      120;

param d :=
2      1      300
3      1      400
6      1      100
1      2      300
3      2      500
4      2      100
5      2      100
1      3      400
2      3      500
4      3      500
6      3      300
7      3      300
2      4      100
3      4      500
5      4      300
7      4      400
8      4      200
```

```
2       5       100
4       5       300
8       5       300
1       6       100
3       6       300
7       6       200
9       6       100
3       7       300
4       7       400
6       7       200
9       7       400
10      7       500
4       8       200
5       8       300
10      8       100
6       9       100
7       9       400
10      9       200
7       10      500
8       10      100
9       10      200;

param cap :=
2       1       150
3       1       150
6       1       150
1       2       150
3       2       150
4       2       150
5       2       150
1       3       150
2       3       150
4       3       150
6       3       150
7       3       150
2       4       150
3       4       150
5       4       150
7       4       150
8       4       150
2       5       150
4       5       150
8       5       150
1       6       150
3       6       150
7       6       150
9       6       150
3       7       150
```

```
4      7      150
6      7      150
9      7      150
10     7      150
4      8      150
5      8      150
10     8      150
6      9      150
7      9      150
10     9      150
7      10     150
8      10     150
9      10     150
11     3      650
12     5      450
13     6      800
14     7      350
15     8      500
4      16     100
9      17     140
10     18     120;

param setf :=
11     4000000
12     3000000
13     5000000
14     2000000
15     2500000;

param setw :=
16     5000000
17     7000000
18     6000000;

param con :=
11     0.1
12     0.15
13     0.2
14     0.125
15     0.175;

param c :=
1      15
2      10;
```