Management Science Group #7, Project #1: Transshipment

Ruby Pittman

Jessica Lu

Arijay Mehta

Alex Mursalim

Taha Chaudhry

March 6, 2020

We did not give or receive any assistance on this project. 









**Step 1: Define your variables and parameters**

* Sets:
  + Let I be the set of sources indexed by i
  + Let J be the set of transhipment points indexed by j
  + Let K be the set of buyers indexed by k
* Parameters:
  + Let Cij be the cost that is due to shipping all source goods i in I to transhipment points j in J
  + Let Cjk be the cost that is due to shipping goods j in J at the transhipment point to buyers k in KS
  + Let Si  is the set of supplier capacity from i in I
  + Let Dk is the set of demand capacity from k in K
* Variables:
  + Let Xij be the amount we want to send sources i in I to the transhipment points j in J
  + Let Xjk be the amount we want to send transhipment points j in J to demand points k in K

**Step 2: Formulate the linear programming model problem for above case study.**

Objective Function: Min CijXij + CjkXjk

S.T

Xij ≤ Ii ∀ i∈I

Xjk ≥ Kk ∀ j∈J

Xij = Xjk ∀ j∈J

Xij, Xjk ≥ 0, ∀ i∈I, j∈J, k∈K

**Step 3: Write AMPL code for the model from step 2 and paste your codes (.mod file, .dat file, .run file) after the formulation.**

**Code with Variables I, J, K**

**Mod File**

set I ordered;

set J ordered;

set K ordered;

param cs {i in I, j in J};

param cd {j in J, k in K};

param s{i in I};

param d{k in K};

var x {i in I, j in J} >= 0;

var y {j in J, k in K} >= 0;

minimize z: sum{i in I, j in J} cs[i,j] \* x[i,j] + sum{j in J, k in K} cd[j,k] \* y[j,k];

subject to c1 {i in I} : sum{j in J} x[i,j] <= s[i];

subject to c2 {j in J} : sum{i in I} x[i,j] - sum{k in K} y[j,k] = 0;

subject to c3 {k in K} : sum{j in J} y[j,k] >= d[k];

**Data File**

set I := 1,2;

set J := 3,4,5;

set K := 6,7,8;

param cs: 3 4 5 :=

1 16 10 12

2 15 14 17;

param cd: 6 7 8 :=

3 6 8 10

4 7 11 11

5 4 5 12;

param: s :=

1 300

2 300;

param: d :=

6 200

7 100

8 300;

**Run File**

option solver cplexamp;

solve;

display z;

display {i in I, j in J} x[i,j];

display {j in J, k in K} y[j,k];

**Step 4: Solve your codes from step 3 using CPLEX solver. Take a screenshot of your results and attach it after the code.**



