COSC364 Assignment 2

George Drummond(53243258),
Ryan Cox(64656394)

May 27, 2017

Abstract

The following pertains to COSC364-17S1 Assignment 2. This was a joint work by George Drummond and Ryan Cox in accordance with the course requirements of COSC364-17S1 and is a result of **equal contribution (50%/50%).**

Contents

1   Problem formulation

We wish to formulate an optimization problem for generic values of X, Y and Z (with $Y > 3$) such that the load on all transit nodes is balanced. Here we outline the governing mathematical expressions concerning the objective function, the decision variables and all constraints.

Notation: $[X] = \{1, 2, 3, ..., X\}$.

1.1   Decision and auxiliary variables

Let u_{ik} be the amount of flow on the link between a given source node S_i and transit node T_k. Likewise let v_{kj} be the amount of flow on the link between a given transit node T_k and destination node D_j.

Therefore, letting x_{ikj} be the part of the demand volume between source node S_i and destination node D_j that is routed through transit node T_k, we achieve

$$u_{ik} + v_{kj} = x_{ikj} \tag{1}$$

$\forall i \in [X], k \in [Y], j \in [Z]$.

Also, the total traffic flow into a transit node is equal to the total traffic flow out of the node, achieving the following balance.

$$\sum_{i=1}^{X} u_{ik} = \sum_{j=1}^{Z} v_{kj} \tag{2}$$

$\forall k \in [Y]$.

1.2   Objective function

The goal of this linear program is to balance the load (total *incoming* traffic) across the transit nodes $T_1, T_2, ..., T_Y$. As the load on a given transit node l_k is simply the sum of the flows from all source nodes to T_k, we achieve

$$l_k = \sum_{i=1}^{X} u_{ik}, \quad \forall k \in [Y]$$

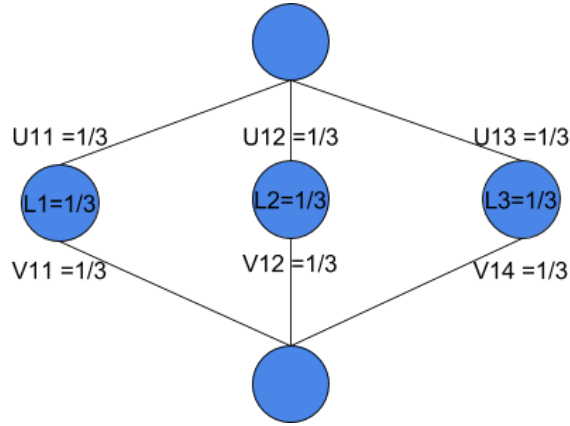As demonstrated in the small example of figure 1.

Figure 1: X=1,Y=4,Z=1

At this stage, it may be tempting to define the objective function l as *equal* to all l_k, $k \in [Y]$ in order to achieve a common minimum load. This however, would actually lead to an infeasible problem for certain cases such as shown in figure2 below in which all l_k cannot be equal.
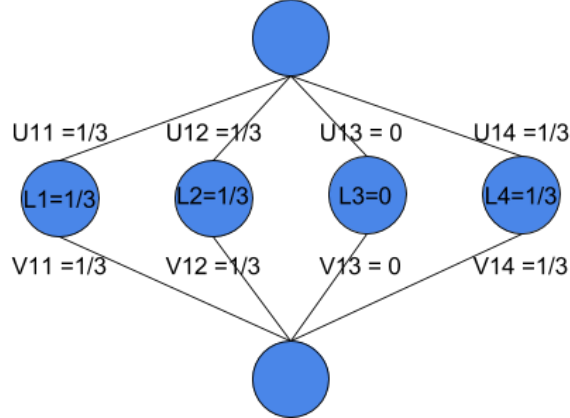

Figure 2: X=1,Y=4,Z=1

To *balance* this load across the transit nodes, we define our objective function l by

$$l = \max l_k = \max \sum_{i=1}^{X} u_{ik}, \quad \forall k \in [Y]$$

Proceeding in this fashion acts to minimise the greatest load on a transit node. The piecewise linear interpretation of this is as follows.

$$l \geq \sum_{i=1}^{X} u_{ik}, \quad \forall k \in [Y] \tag{3}$$

1.3   Demand constraints

We now turn our attention to demand constraints and the global requirement that each demand volume shall be split over exactly three different paths, such that each path gets an equal share of the demand volume.

let w_{ikj} be the indicator variable for the path between source node S_i and destination node D_j through transit node T_k, taking the value 1 if path $S_i->T_k->Z_j$ carries data and 0 otherwise. With our 3 path restriction, we achieve

$$\sum_{k=1}^{Y} w_{ikj} = 3, \quad \forall i \in [X], j \in [Z] \tag{4}$$

Now to consider the demand volume between source node S_i and destination node D_j. Firstly, we have that this demand volume is simply the sum of the parts of the demand volume through each transit router $T_k, \quad k \in [Y]$.

$$\sum_{k=1}^{Y} x_{ikj} = h_{ij} = i + j$$

However, we also have that x_{ikj} will be non-zero if and only if there is flow on path $S_i->T_k->Z_j$ ($w_{ikj} = 1$). As this occurs for precisely 3 x_{ikj} (say $x_{ik_1j}, x_{ik_2j}, x_{ik_3j}$) and the demand h_{ij} is shared *equally* across the paths that do possess flows, we achieve the following.

$$x_{ikj} = \left\{ \begin{array}{ll} \frac{(i+j)}{3}, & \text{if } w_{ikj} = 1 \\ 0, & \text{if } w_{ikj} = 0 \end{array} \right\}$$

Or explicitly.

$$x_{ikj} = w_{ikj}\frac{h_{ij}}{3} = \frac{(i+j)}{3}w_{ikj}, \quad \forall i \in [X], k \in [Y], j \in [Z] \tag{5}$$

1.4   Additional constraints

Capacity:
$\forall i \in [X], k \in [Y], j \in [Z]$, we have,

$$u_{ik} \leq c_{ik} \tag{6}$$

$$v_{kj} \leq d_{kj} \tag{7}$$

Non-negativity:

$\forall i \in [X], k \in [Y], j \in [Z]$, we have,

$$x_{ikj} \geq 0 \tag{8}$$
$$u_{ik} \geq 0 \tag{9}$$
$$v_{kj} \geq 0 \tag{10}$$
$$l \geq 0 \tag{11}$$

1.5 Model summary

The constraints generated by equations 1 through to 11 suffice to fully describe our optimisation problem.

2 Results

3   Appendix

3.1   LP generation source file

```python
1  import sys
2
3
4  def print_aux(X,Y,Z):
5      aux = ""
6      for i in range(1,X+1):
7          for k in range(1,Y+1):
8              for j in range(1,Z+1):
9                  aux += "u{0}{1} + v{1}{2} - x{0}{1}{2} = 0\n".format(i,
   k,j)
10
11      print(aux,end="")
12
13      for k in range(1,Y+1):
14          mysum = ""
15          for i in range(1,X+1):
16              mysum += "+ u{}{} ".format(i,k)
17          for j in range(1,Z+1):
18              mysum += "- v{}{} ".format(k,j)
19
20          mysum += "= 0"
21          print(mysum[2:])
22
23
24  def print_objective_constraints(X,Y,Z):
25      for k in range(1,Y+1):
26          con = "1 - "
27          for i in range(1,X+1):
28              con += "u{}{} - ".format(i,k)
29
30          con = con[:-2] + ">= 0"
31
32          print(con)
33
34  def print_demand(X,Y,Z):
35      '''prints demand constraints'''
36      demand = ""
37      for i in range(1,X+1):
38          for j in range(1,Z+1):
39              mysum1 = mysum2 = mysum3 = ""
40              for k in range(1,Y+1):
41                  demand += "x{0}{1}{2} - {3}w{0}{1}{2} = 0\n".format(i,k
   ,j,(i+j)/3)
42                  mysum1 += "w{}{}{} + ".format(i,k,j)
43                  mysum2 += "w{0}{1}{2} x{0}{1}{2} + ".format(i,k,j)
44                  mysum3 += "x{}{}{} + ".format(i,k,j)
45
```

```python
46
47                    mysum1 = mysum1[:-2] + "= "
48                    mysum2 = mysum2[:-2] + "= "
49                    mysum3 = mysum3[:-2] + "= "
50
51                    mysum1 += "3"
52                    mysum2 += str(i + j)
53                    mysum3 += str(i + j)
54
55                    demand += mysum1 + "\n" #+ mysum2 + "\n" + mysum3+ "\n"
56
57
58
59        print(demand, end="")
60
61
62  def print_capp(X,Y,Z):
63        capp = ""
64        for i in range(1,X+1):
65              for k in range(1,Y+1):
66                    capp += "u{0}{1} - c{0}{1} <= 0\n".format(i,k)
67
68
69
70        for k in range(1,Y+1):
71              for j in range(1,Z+1):
72                    capp += "v{0}{1} - d{0}{1} <= 0\n".format(k,j)
73
74
75        print(capp, end="")
76
77
78  def print_integer(X,Y,Z):
79        integer = ""
80        for i in range(1,X+1):
81              for k in range(1,Y+1):
82                    for j in range(1,Z+1):
83                          integer += "w{}{}{}\n".format(i,k,j)
84        print(integer, end="")
85
86  def print_nonneg(X,Y,Z):
87        nonneg = ""
88        for i in range(1,X+1):
89              for k in range(1,Y+1):
90                    nonneg += "0 <= u{}{}\n".format(i,k)
91                    for j in range(1,Z+1):
92
93                          nonneg += "0 <= x{}{}{}\n".format(i,k,j)
94
95        for k in range(1,Y+1):
96              for j in range(1,Z+1):
97                    nonneg += "0 <= v{}{}\n".format(k,j)
```

```python
 98
 99        nonneg += "0 <= l\n"
100
101        print(nonneg, end="")
102
103
104 def main():
105        (X,Y,Z) = sys.argv[1:4]
106        X = int(X)
107        Y = int(Y)
108        Z = int(Z)
109
110        print("X = {}, Y = {}, Z = {}".format(X,Y,Z))
111        print("Minimize")
112        print("  l")
113        print("Subject to")
114        print_aux(X,Y,Z)
115        print_objective_constraints(X,Y,Z)
116        print_demand(X,Y,Z)
117        print_capp(X,Y,Z)
118        print("Bounds")
119        print_nonneg(X,Y,Z)
120        print("Integer")
121        print_integer(X,Y,Z)
122        print("End")
123
124
125
126 main()
```

3.2 LP File for X = 3, Y = 2 and Z = 4

```
 1 X = 3, Y = 2, Z = 4
 2 Minimize
 3  l
 4 Subject to
 5 u11 + v11 − x111 = 0
 6 u11 + v12 − x112 = 0
 7 u11 + v13 − x113 = 0
 8 u11 + v14 − x114 = 0
 9 u12 + v21 − x121 = 0
10 u12 + v22 − x122 = 0
11 u12 + v23 − x123 = 0
12 u12 + v24 − x124 = 0
13 u21 + v11 − x211 = 0
14 u21 + v12 − x212 = 0
15 u21 + v13 − x213 = 0
16 u21 + v14 − x214 = 0
17 u22 + v21 − x221 = 0
18 u22 + v22 − x222 = 0
19 u22 + v23 − x223 = 0
20 u22 + v24 − x224 = 0
21 u31 + v11 − x311 = 0
22 u31 + v12 − x312 = 0
23 u31 + v13 − x313 = 0
24 u31 + v14 − x314 = 0
25 u32 + v21 − x321 = 0
26 u32 + v22 − x322 = 0
27 u32 + v23 − x323 = 0
28 u32 + v24 − x324 = 0
29 u11 + u21 + u31 − v11 − v12 − v13 − v14 = 0
30 u12 + u22 + u32 − v21 − v22 − v23 − v24 = 0
31 l − u11 − u21 − u31 >= 0
32 l − u12 − u22 − u32 >= 0
33 x111 − 0.6666666666666666 w111 = 0
34 x121 − 0.6666666666666666 w121 = 0
35 w111 + w121 = 3
36 x112 − 1.0 w112 = 0
37 x122 − 1.0 w122 = 0
38 w112 + w122 = 3
39 x113 − 1.3333333333333333 w113 = 0
40 x123 − 1.3333333333333333 w123 = 0
41 w113 + w123 = 3
42 x114 − 1.6666666666666667 w114 = 0
43 x124 − 1.6666666666666667 w124 = 0
44 w114 + w124 = 3
45 x211 − 1.0 w211 = 0
46 x221 − 1.0 w221 = 0
47 w211 + w221 = 3
48 x212 − 1.3333333333333333 w212 = 0
49 x222 − 1.3333333333333333 w222 = 0
50 w212 + w222 = 3
```

```
51   x213 − 1.6666666666666667 w213 = 0
52   x223 − 1.6666666666666667 w223 = 0
53   w213 + w223 = 3
54   x214 − 2.0 w214 = 0
55   x224 − 2.0 w224 = 0
56   w214 + w224 = 3
57   x311 − 1.3333333333333333 w311 = 0
58   x321 − 1.3333333333333333 w321 = 0
59   w311 + w321 = 3
60   x312 − 1.6666666666666667 w312 = 0
61   x322 − 1.6666666666666667 w322 = 0
62   w312 + w322 = 3
63   x313 − 2.0 w313 = 0
64   x323 − 2.0 w323 = 0
65   w313 + w323 = 3
66   x314 − 2.3333333333333335 w314 = 0
67   x324 − 2.3333333333333335 w324 = 0
68   w314 + w324 = 3
69   u11 − c11 <= 0
70   u12 − c12 <= 0
71   u21 − c21 <= 0
72   u22 − c22 <= 0
73   u31 − c31 <= 0
74   u32 − c32 <= 0
75   v11 − d11 <= 0
76   v12 − d12 <= 0
77   v13 − d13 <= 0
78   v14 − d14 <= 0
79   v21 − d21 <= 0
80   v22 − d22 <= 0
81   v23 − d23 <= 0
82   v24 − d24 <= 0
83   Bounds
84   0 <= u11
85   0 <= x111
86   0 <= x112
87   0 <= x113
88   0 <= x114
89   0 <= u12
90   0 <= x121
91   0 <= x122
92   0 <= x123
93   0 <= x124
94   0 <= u21
95   0 <= x211
96   0 <= x212
97   0 <= x213
98   0 <= x214
99   0 <= u22
100  0 <= x221
101  0 <= x222
102  0 <= x223
```

```
103  0 <= x224
104  0 <= u31
105  0 <= x311
106  0 <= x312
107  0 <= x313
108  0 <= x314
109  0 <= u32
110  0 <= x321
111  0 <= x322
112  0 <= x323
113  0 <= x324
114  0 <= v11
115  0 <= v12
116  0 <= v13
117  0 <= v14
118  0 <= v21
119  0 <= v22
120  0 <= v23
121  0 <= v24
122  0 <= l
123  Integer
124  w111
125  w112
126  w113
127  w114
128  w121
129  w122
130  w123
131  w124
132  w211
133  w212
134  w213
135  w214
136  w221
137  w222
138  w223
139  w224
140  w311
141  w312
142  w313
143  w314
144  w321
145  w322
146  w323
147  w324
148  End
```