Referee Assignments for the 33rd Conference on How to Ride the Lightning

Jingxu Gao, jg2238

December 9, 2016

Abstract

Assignment problem with respect to accessible constraints and load balance is a widely encountered problem in real life and has gained much attention. In this paper, I will address the referee assignment problem of papers submitted to a conference. An integer program is formulated and I leverage the Gurobi package to solve the problem. We additionally discuss some more about the different formulations for load balance. Finally, we will evaluate the solution in terms of both how well the balance is achieved and the overall satisfaction of all the referees. The report is made of two parts, the second part. general report for manager, is meant for people without knowledge into operation research and the third part, which is the technical detail for scientist, is meant for who would like to understand the model and possibly use it in their applications.

1 Introduction

The background of the problem is that for the 33rd Conference on How to Ride the Lightning, there are 71 papers submitted to the conference and21 referees to review these papers. Each paper needs to be reviewed by 3 referees. And we hope that each referee should review roughly the same number of papers so as to be fair to referees.

2 General Report for Manager

2.1 Define Decision in need of making and related constraints

The problem we here address is to decide which paper should be assigned to which referee.

And we encounter certain constraints and goals, that is, first mandatorily, we cannot assign a paper to a referee if he/she announces that he/she has a conflict interest, and secondly, a paper should be assigned to exactly 3 referees, and thirdly, we hope to achieve fair assignment as much as possible in terms of number of papers assigned to each referee, lastly, though not explicitly required, we hope that referee can read the papers they announce they want to read.

2.2 Ways to approach the problem

Here I first work on finding the potential feasible solutions by restricting the first, second, and third constraints I mentioned in the previous section. I will gradually reduce the maximum difference of numbers of papers to read by any two referees and see how small the difference could be.

And once the "feasible region" is small enough, we want to consider the forth one, to maximize referees' satisfaction, which is not explicitly required.

2.3 Result proposed

Referee 1 is assigned to paper [1, 23, 34, 39, 45, 46, 47, 48, 49, 69] Referee 2 is assigned to paper [16, 24, 26, 33, 37, 44, 62, 63, 66, 70] Referee 3 is assigned to paper [15, 18, 35, 40, 47, 50, 51, 53, 57, 68] Referee 4 is assigned to paper [6, 8, 11, 27, 41, 50, 58, 59, 61, 71] Referee 5 is assigned to paper [3, 4, 8, 28, 31, 53, 59, 61, 63, 66] Referee 6 is assigned to paper [1, 6, 7, 30, 31, 32, 49, 57, 58, 62, 64] Referee 7 is assigned to paper [4, 14, 19, 23, 25, 36, 46, 52, 57, 64] Referee 8 is assigned to paper [3, 7, 11, 12, 37, 40, 41, 58, 63, 71] Referee 9 is assigned to paper [12, 20, 21, 22, 29, 38,

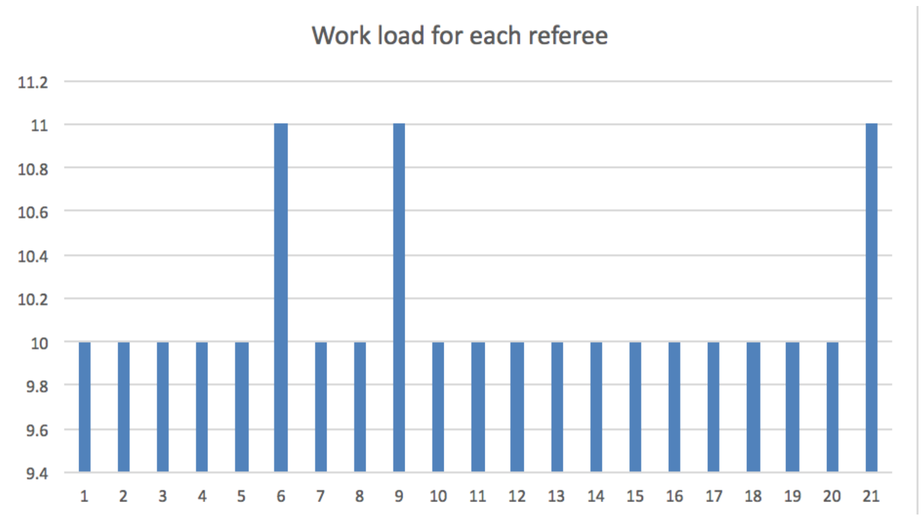


Figure 1: work load for each referee

Denote x_{ij} =1 if paper i is assigned to referee j. Denote u_{ij} as as the utility referee $j$from reading paper i, detailed value will be assigned later according to "yes", "no", "maybe", for preference but u_{ij}=0 is fixed for "conflict". Denote r as the balance ratio, which restricts the work load ratio for each two pair of referees can not exceed r.

$$\begin{aligned} max \quad & z = \sum_i \sum_j u_{ij} x_{ij} \\ s.t. \quad & \sum_j x_{ij} = 3, \ for\ all\ i \\ & x_{ij} \leq u_{ij}, \ for\ all\ i, \ j \\ & \sum_i x_{im} \leq r * \sum_j x_{in}, \ for\ all\ m, \ n \end{aligned}$$

Figure 2: notation and model formulation

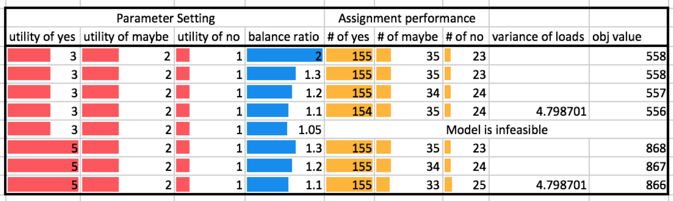
43, 46, 60, 65] Referee 10 is assigned to paper [2, 5, 7, 12, 14, 18, 25, 40, 47, 54, 62] Referee 11 is assigned to paper [5, 13, 17, 19, 28, 51, 53, 55, 59, 60] Referee 12 is assigned to paper [10, 20, 21, 27, 30, 36, 42, 44, 54, 56] Referee 13 is assigned to paper [2, 18, 24, 30, 35, 45, 52, 68, 69, 70] Referee 14 is assigned to paper [9, 10, 15, 26, 33, 34, 42, 48, 61, 64] Referee 15 is assigned to paper [2, 5, 10, 17, 33, 41, 43, 45, 48, 54] Referee 16 is assigned to paper [15, 16, 21, 29, 31, 34, 38, 43, 55, 56, 67] Referee 17 is assigned to paper [3, 4, 8, 9, 13, 22, 24, 32, 35, 42] Referee 18 is assigned to paper [6, 9, 20, 22, 29, 32, 38, 39, 56, 65] Referee 19 is assigned to paper [11, 13, 14, 23, 25, 49, 52, 55, 67, 68] Referee 20 is assigned to paper [1, 16, 17, 19, 27, 36, 39, 60, 65, 71] Referee 21 is assigned to paper [26, 28, 37, 44, 50, 51, 66, 67, 69, 70]

Figure 1 shows the balance we achieved, which is very good as all the referees will need to read either 10 or 11.

3 Technical Detail for Scientist

3.1 Notation and model formulation

See figure 2.



Parameter Setting				Assignment performance				
utility of yes	utility of maybe	utility of no	balance ratio	# of yes	# of maybe	# of no	variance of loads	obj value
3	2	1	2	155	35	23		558
3	2	1	1.3	155	35	23		558
3	2	1	1.2	155	34	24		557
3	2	1	1.1	154	35	24	4.798701	556
3	2	1	1.05	Model is infeasible				
5	2	1	1.3	155	35	23		868
5	2	1	1.2	155	34	24		867
5	2	1	1.1	155	33	25	4.798701	866

Figure 3: result comparison of different settings (*corrected: variance=0.128751)

3.2 Experiment and performance analysis

I tried different parameter settings. Intuitively, the lower the balance ratio is , the better the assignment is; the higher the utility value difference of "yes", "no", "maybe"is, the better the assignment is.

As time is limited, we tried a limited amount of different settings by intuition and sequentially decided the next setting in the procedure. (* better experiment design)

Since we give different settings different utilities, the objective value is of non-sense to us. I do the evaluation of different assignments according to 1) number of yes, 2) range and variance of loads for different referee

The result is in Figure 3.

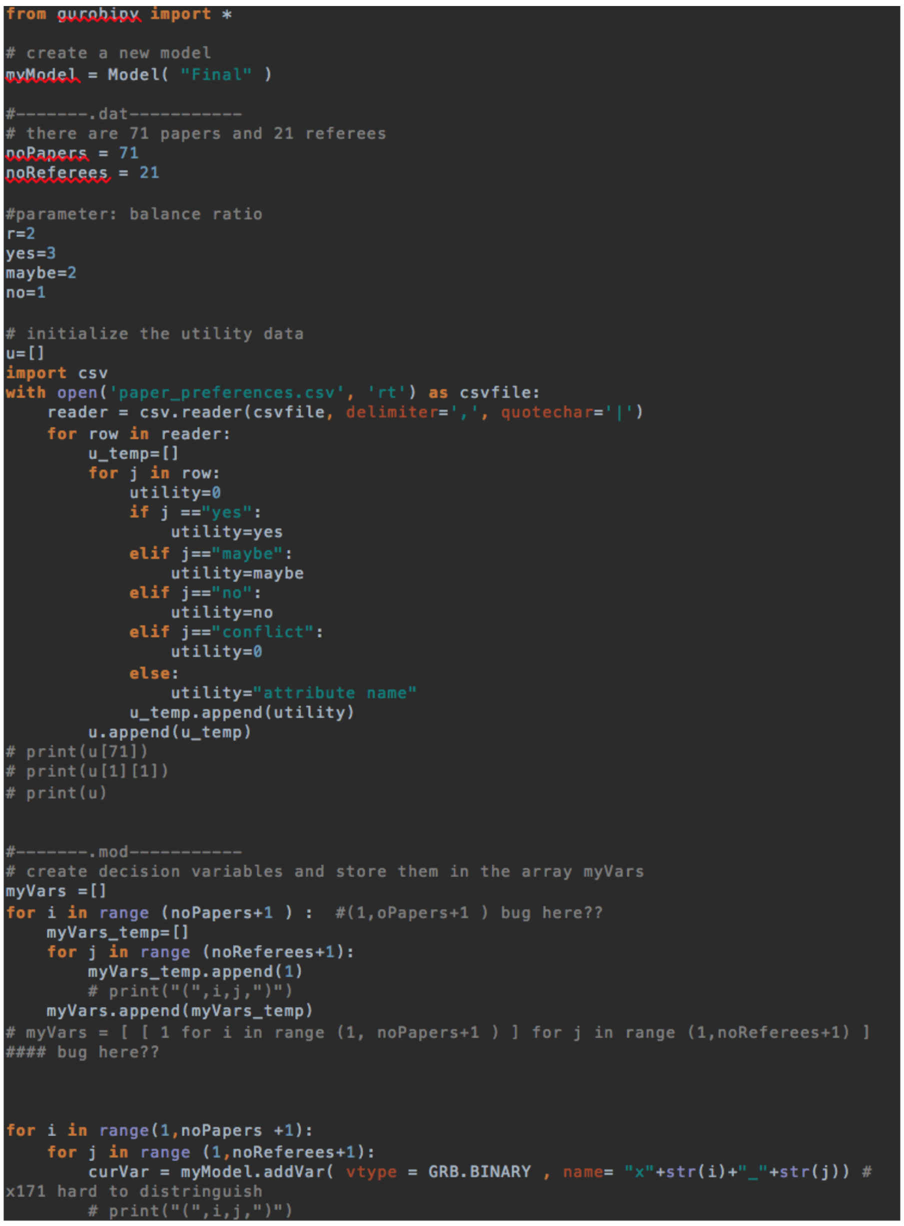
We chose the best settings and achieved the best assignment according to all the performance evaluation parameters, by intuition(* see Future research section for multi-objective optimization).

Since we care more about fairness than making referees happier for reading the papers they want, we decided to choose the (3,2,1) for "yes", "no", "maybe", 1.1 for balance ratio. Also we notice that this is the same for (5,2,1) for "yes", "no", "maybe", 1.1 for balance ratio, which means that the balance load constraints have been very tight. (* we can do duality and sensitivity analysis on this in the future to approach this in detail.)

Here is the detailed solution, the [i,j] here represents paper i assigned to referee j.

(5,2,1) for "yes", "no", "maybe", 1.1 for balance ratio

[[’1’, ’1’], [’1’, ’6’], [’1’, ’20’], [’2’, ’10’], [’2’, ’13’], [’2’, ’15’], [’3’, ’5’], [’3’, ’8’], [’3’, ’11’], [’4’, ’5’], [’4’, ’7’], [’4’, ’17’], [’5’, ’10’], [’5’, ’11’], [’5’, ’20’], [’6’, ’4’], [’6’, ’6’], [’6’, ’18’], [’7’, ’6’], [’7’, ’8’], [’7’, ’10’], [’8’, ’4’], [’8’, ’5’], [’8’, ’17’], [’9’, ’12’], [’9’, ’14’], [’9’, ’18’], [’10’, ’12’], [’10’, ’15’], [’10’, ’19’], [’11’, ’4’], [’11’, ’8’], [’11’, ’19’], [’12’, ’8’], [’12’, ’9’], [’12’, ’10’], [’13’, ’11’], [’13’, ’17’], [’13’, ’19’], [’14’, ’7’], [’14’, ’10’], [’14’, ’19’], [’15’, ’3’], [’15’, ’14’], [’15’, ’16’], [’16’, ’2’], [’16’, ’16’], [’16’, ’20’], [’17’, ’8’], [’17’, ’16’], [’17’, ’20’], [’18’, ’3’], [’18’, ’10’], [’18’, ’13’], [’19’, ’7’], [’19’, ’11’], [’19’, ’20’], [’20’, ’9’], [’20’, ’12’], [’20’, ’18’], [’21’, ’6’], [’21’, ’9’], [’21’, ’16’], [’22’, ’9’], [’22’, ’15’], [’22’, ’18’], [’23’, ’1’], [’23’, ’7’], [’23’, ’20’], [’24’, ’3’], [’24’, ’13’], [’24’, ’17’], [’25’, ’7’], [’25’, ’10’], [’25’, ’19’], [’26’, ’2’], [’26’, ’14’], [’26’, ’21’], [’27’, ’4’], [’27’, ’12’], [’27’, ’20’], [’28’, ’2’], [’28’, ’5’], [’28’, ’21’], [’29’, ’9’], [’29’, ’15’], [’29’, ’18’], [’30’, ’6’], [’30’, ’13’], [’30’, ’15’], [’31’, ’5’], [’31’, ’6’], [’31’, ’16’], [’32’, ’6’], [’32’, ’17’], [’32’, ’18’], [’33’, ’2’], [’33’, ’14’], [’33’, ’15’], [’34’, ’1’], [’34’, ’14’], [’34’, ’16’], [’35’, ’3’], [’35’, ’13’], [’35’, ’17’], [’36’, ’3’], [’36’, ’4’], [’36’, ’7’], [’37’, ’2’], [’37’, ’8’], [’37’, ’21’], [’38’, ’9’], [’38’, ’16’], [’38’, ’18’], [’39’, ’1’], [’39’, ’18’], [’39’, ’20’], [’40’, ’3’], [’40’, ’8’], [’40’, ’10’], [’41’, ’8’], [’41’, ’12’], [’41’, ’16’], [’42’, ’12’], [’42’, ’14’], [’42’, ’17’], [’43’, ’9’], [’43’, ’12’], [’43’, ’17’], [’44’, ’2’], [’44’, ’12’], [’44’, ’21’], [’45’, ’1’], [’45’, ’13’], [’45’, ’15’], [’46’, ’1’], [’46’, ’7’], [’46’, ’9’], [’47’, ’1’], [’47’, ’10’], [’47’, ’14’], [’48’, ’1’], [’48’, ’14’], [’48’, ’15’], [’49’, ’1’], [’49’, ’6’], [’49’, ’19’], [’50’, ’4’], [’50’, ’6’], [’50’, ’21’], [’51’, ’3’], [’51’, ’11’], [’51’, ’21’], [’52’, ’11’], [’52’, ’13’], [’52’, ’19’], [’53’, ’3’], [’53’, ’5’], [’53’, ’13’], [’54’, ’10’], [’54’, ’12’], [’54’, ’15’], [’55’, ’11’], [’55’, ’16’], [’55’, ’19’], [’56’, ’12’], [’56’, ’17’], [’56’, ’18’], [’57’, ’3’], [’57’, ’6’], [’57’, ’7’], [’58’, ’4’], [’58’, ’6’], [’58’, ’8’], [’59’, ’5’], [’59’, ’7’], [’59’, ’11’], [’60’, ’4’], [’60’, ’9’], [’60’, ’11’], [’61’, ’4’], [’61’, ’5’], [’61’, ’14’], [’62’, ’2’], [’62’, ’6’], [’62’, ’15’], [’63’, ’2’], [’63’, ’5’], [’63’, ’17’], [’64’, ’6’], [’64’, ’7’], [’64’, ’14’], [’65’, ’9’], [’65’, ’18’], [’65’, ’20’], [’66’, ’2’], [’66’, ’5’], [’66’, ’21’], [’67’, ’16’], [’67’, ’19’], [’67’, ’21’], [’68’, ’3’], [’68’, ’13’], [’68’, ’19’], [’69’, ’1’], [’69’, ’13’], [’69’, ’21’], [’70’, ’2’], [’70’, ’11’], [’70’, ’21’], [’71’, ’4’], [’71’, ’8’], [’71’, ’20’]]



```
from gurobipy import *

# create a new model
myModel = Model( "Final" )

#-------.dat----------
# there are 71 papers and 21 referees
noPapers = 71
noReferees = 21

#parameter: balance ratio
r=2
yes=3
maybe=2
no=1

# initialize the utility data
u=[]
import csv
with open('paper_preferences.csv', 'rt') as csvfile:
    reader = csv.reader(csvfile, delimiter=',', quotechar='|')
    for row in reader:
        u_temp=[]
        for j in row:
            utility=0
            if j =="yes":
                utility=yes
            elif j=="maybe":
                utility=maybe
            elif j=="no":
                utility=no
            elif j=="conflict":
                utility=0
            else:
                utility="attribute name"
            u_temp.append(utility)
        u.append(u_temp)
# print(u[71])
# print(u[1][1])
# print(u)


#-------.mod----------
# create decision variables and store them in the array myVars
myVars =[]
for i in range (noPapers+1 ) :  #(1,oPapers+1 ) bug here??
    myVars_temp=[]
    for j in range (noReferees+1):
        myVars_temp.append(1)
        # print("(",i,j,")")
    myVars.append(myVars_temp)
# myVars = [ [ 1 for i in range (1, noPapers+1 ) ] for j in range (1,noReferees+1) ]
#### bug here??



for i in range(1,noPapers +1):
    for j in range (1,noReferees+1):
        curVar = myModel.addVar( vtype = GRB.BINARY , name= "x"+str(i)+"_"+str(j)) #
x171 hard to distringuish
        # print("(",i,j,")")
```

Figure 4: code (1/3)

4 Future research

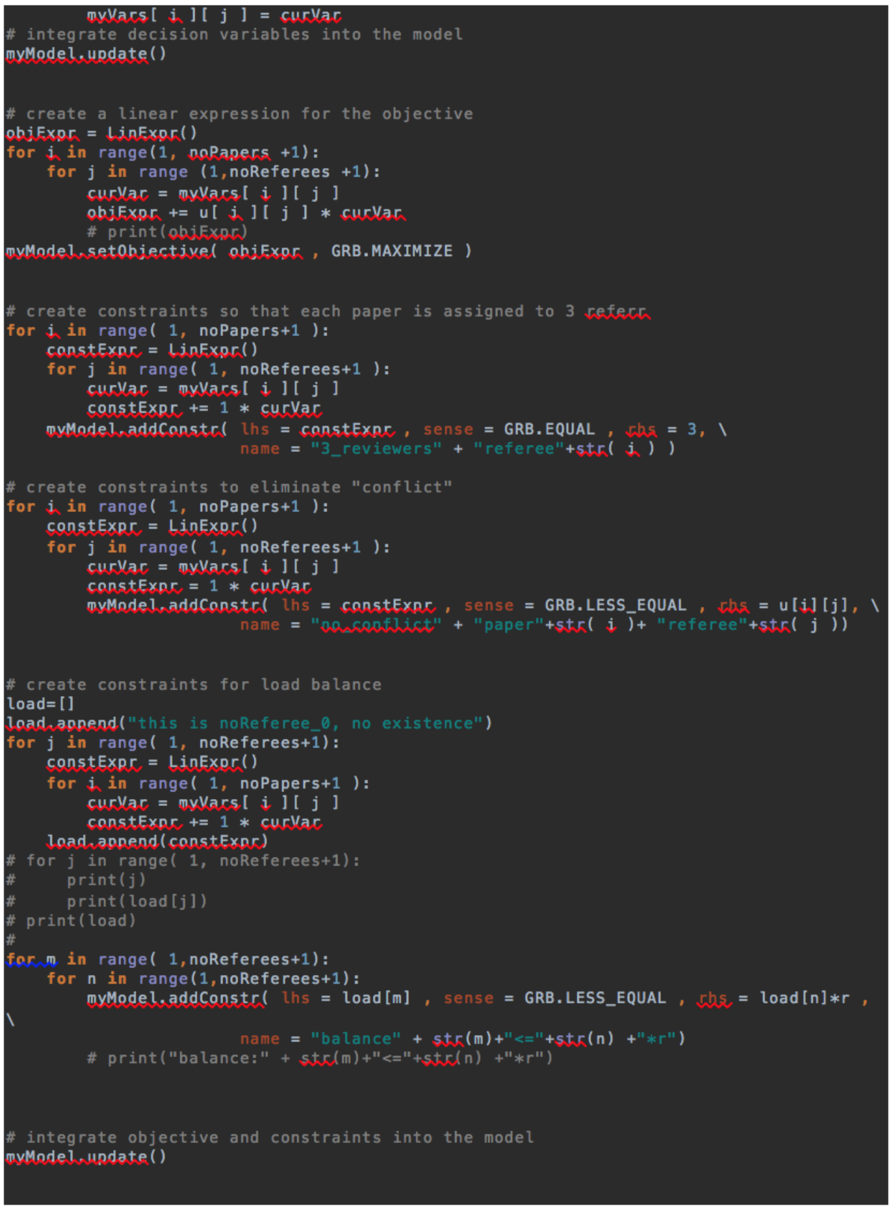
Here we are faced with the compromise problem as better work load better will result in more "maybe" and "no". Here we intuitively decide on the best assignment (see sentences marked with * in the previous section). It's up to future research to try other models like "putting workload balance into objective function as a penalty" (multi-objective optimization), to better quantify this compromise by adding weights for different sub objective.

5 Appendix

In this part we will include the codes in python. See figure 4,5,6.

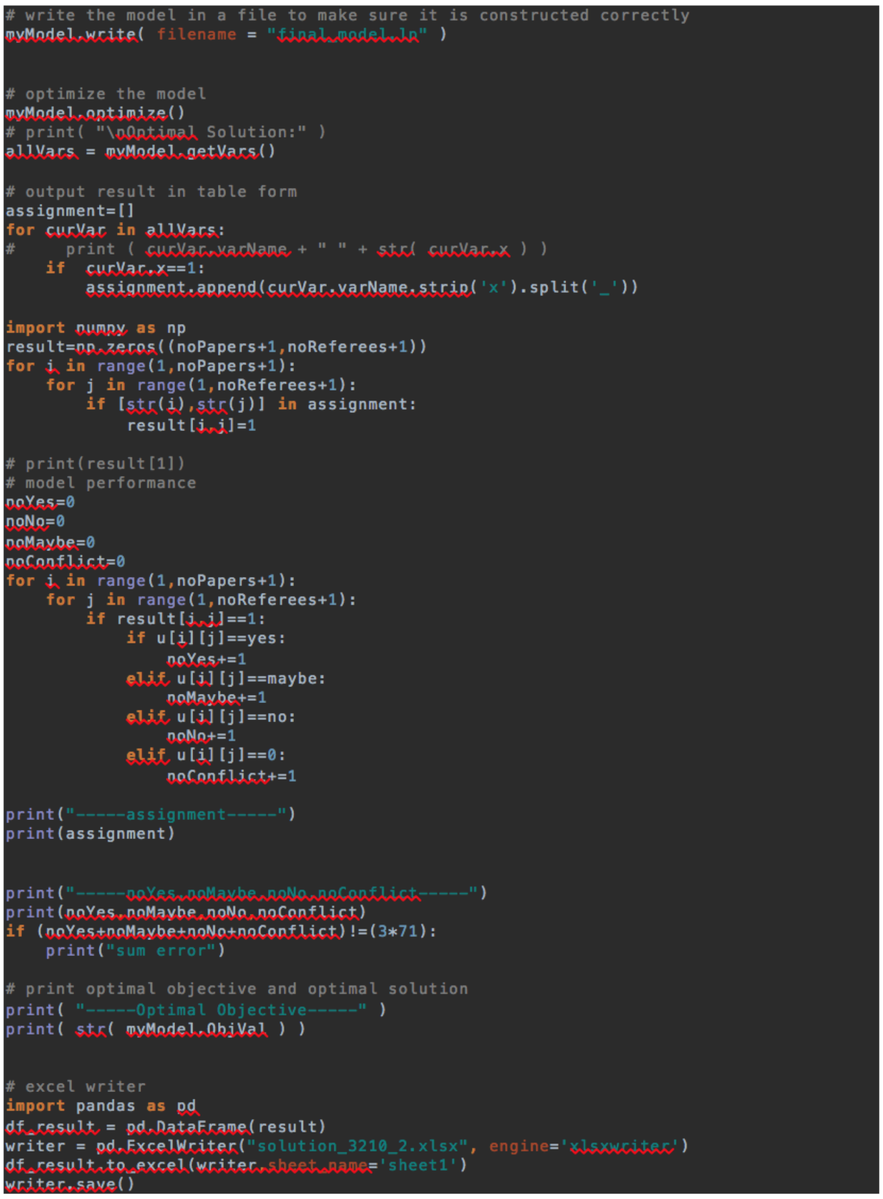
plus the codes we use to generate the solution statement.

for j in range(1,noReferees+1): temp=[] for i in range(1,noPapers+1): if result[i,j]==1: temp.append(i) print("Referee", j, "is assigned to paper" ,temp)



```
        myVars[ i ][ j ] = curVar
# integrate decision variables into the model
myModel.update()


# create a linear expression for the objective
objExpr = LinExpr()
for i in range(1, noPapers +1):
    for j in range (1,noReferees +1):
        curVar = myVars[ i ][ j ]
        objExpr += u[ i ][ j ] * curVar
        # print(objExpr)
myModel.setObjective( objExpr , GRB.MAXIMIZE )


# create constraints so that each paper is assigned to 3 referr
for i in range( 1, noPapers+1 ):
    constExpr = LinExpr()
    for j in range( 1, noReferees+1 ):
        curVar = myVars[ i ][ j ]
        constExpr += 1 * curVar
    myModel.addConstr( lhs = constExpr , sense = GRB.EQUAL , rhs = 3, \
                       name = "3_reviewers" + "referee"+str( i ) )

# create constraints to eliminate "conflict"
for i in range( 1, noPapers+1 ):
    constExpr = LinExpr()
    for j in range( 1, noReferees+1 ):
        curVar = myVars[ i ][ j ]
        constExpr = 1 * curVar
        myModel.addConstr( lhs = constExpr , sense = GRB.LESS_EQUAL , rhs = u[i][j], \
                       name = "no_conflict" + "paper"+str( i )+ "referee"+str( j ))


# create constraints for load balance
load=[]
load.append("this is noReferee_0, no existence")
for j in range( 1, noReferees+1):
    constExpr = LinExpr()
    for i in range( 1, noPapers+1 ):
        curVar = myVars[ i ][ j ]
        constExpr += 1 * curVar
    load.append(constExpr)
# for j in range( 1, noReferees+1):
#     print(j)
#     print(load[j])
# print(load)
#
for m in range( 1,noReferees+1):
    for n in range(1,noReferees+1):
        myModel.addConstr( lhs = load[m] , sense = GRB.LESS_EQUAL , rhs = load[n]*r ,
\
                       name = "balance" + str(m)+"<="+str(n) +"*r")
        # print("balance:" + str(m)+"<="+str(n) +"*r")



# integrate objective and constraints into the model
myModel.update()
```

Figure 5: code (2/3)



```python
# write the model in a file to make sure it is constructed correctly
myModel.write( filename = "final_model.lp" )


# optimize the model
myModel.optimize()
# print( "\nOptimal Solution:" )
allVars = myModel.getVars()

# output result in table form
assignment=[]
for curVar in allVars:
#     print ( curVar.varName + " " + str( curVar.x ) )
    if  curVar.x==1:
        assignment.append(curVar.varName.strip('x').split('_'))

import numpy as np
result=np.zeros((noPapers+1,noReferees+1))
for i in range(1,noPapers+1):
    for j in range(1,noReferees+1):
        if [str(i),str(j)] in assignment:
            result[i,j]=1

# print(result[1])
# model performance
noYes=0
noNo=0
noMaybe=0
noConflict=0
for i in range(1,noPapers+1):
    for j in range(1,noReferees+1):
        if result[i,j]==1:
            if u[i][j]==yes:
                noYes+=1
            elif u[i][j]==maybe:
                noMaybe+=1
            elif u[i][j]==no:
                noNo+=1
            elif u[i][j]==0:
                noConflict+=1

print("-----assignment-----")
print(assignment)


print("-----noYes,noMaybe,noNo,noConflict-----")
print(noYes,noMaybe,noNo,noConflict)
if (noYes+noMaybe+noNo+noConflict)!=(3*71):
    print("sum error")

# print optimal objective and optimal solution
print( "-----Optimal Objective-----" )
print( str( myModel.ObjVal ) )


# excel writer
import pandas as pd
df_result = pd.DataFrame(result)
writer = pd.ExcelWriter("solution_3210_2.xlsx", engine='xlsxwriter')
df_result.to_excel(writer,sheet_name='sheet1')
writer.save()
```

Figure 6: code (3/3)

6 No references at the moment