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

Hong Gan hg383

December 9, 2016

Abstract

The 33rd Conference on How to Ride the Lightning will be held in San Francisco on July 27, 2017. In the Conference these are 71 papers submitted to the conference, these paper are going to be reviews by 21 referees. Each referee has his/her own preference. In this paper we proposed a solution to the referee assignments problem by using Gurobi.

1 Introduction

I am on the organization committee for the 33rd Conference on How to Ride the Lightning to be held in San Francisco on July 27, 2017. I am asked to come up with an optimization model to assign the papers that have been submitted to the conference to the referees. There are 71 papers submitted to the conference. There are 21 referees to review these papers. Each paper needs to be reivewed by 3 referees. My solution is to find out how to schedule the paper assignment.

As a matter of fact, Not all referees are experts in the subject matter of all papers. Before doing the assignment, abstracts of the papers have been sent to the referees and the referees were asked to provide their preferences. Each referee will go through all the abstracts and rank them in 4 categories 'yes', 'no', 'maybe', 'conflict'.

"yes, I can definitely review this paper," "I can maybe review this paper," "no, I do not want to review this paper," "I have a conflict of interest with this paper, so it is unethical for me to review this paper." Ideally, i would like the referees to be assigned to the papers for which they said "yes, I can definitely review this paper" or "I can maybe review this paper." If it is unavoidable, then a referee may be assigned to a paper for which she said "no, I do not want to review this paper." However, I definitely do not want to assign to a paper to a referee for which she said "I have a conflict of interest with this paper, so it is unethical for me to review this paper."

2 Problem Analysing

This is typical scheduling and assignment problem. I have to assign each paper to 3 different referees. Since referees have their own preferences. Ideally, each referee should be assigned paper they feel most comforatble. However number of papers is far more than the number of referees. So it is impossible to satisfy every single one. But we can find a optimal solution to satisfied most of the referees.

First we need to find a way of evaluating how good our solution is. We assign int value to each category for scroing purpose. I come up a scoring table like below, if a paper is assgined to referee for which she is totally comforatble, we got 0 points added to the total score. If we a paper is assigned to referee of which she feel conflicted, we got 3 points added to the total score.

so for any solution, an according score will be computed to evaluate how good this solution is.

Category	Score
yes	0
maybe	1
no	2
conflict	3

Table 1: Score Table.

Now I have a way to tell how good our solution. Let's do some analysis to comfirm our strategy. First, according to requirement, each paper shoule be reviewed by 3 different referees. Second, each referees should review roughly the same number of papers. Last but not least, never assign a paper to a referee in conflict. This is really bad.

3 Modeling

3.1 Some Definition

In order to formulate the problem, I need to give out some definition. we will have a assignment matrix A_{71x21}, matrix A has 71 rows and 21 columns.

$$A_{i,j} = \begin{cases} 1 & \text{paper i is going to be assigned to referee j} \\ 0 & \text{Otherwise} \end{cases} \tag{1}$$

Remember we have collected referee's preference. We will convert the preference to a matrix P_{71x21},

$$P_{i,j} = \begin{cases} 0 & \text{referee j can definitely review paper i} \\ 1 & \text{referee j can maybe review paper i} \\ 2 & \text{referee j do not want to review paper i} \\ 3 & \text{referee j have a conflict with reviewing paper i} \end{cases} \tag{2}$$

3.2 Formulating the problem in compact form

3.2.1 Objective Function

Now we have assignment matrix and preference matrix, the preference matrix is read in through xls, and assignment matrix is decision variables. We can compute a total score by:

$$score = \sum_{i=1}^{71} \sum_{j=1}^{21} P_{i,j} \cdot A_{i,j} \tag{3}$$

The smaller the score is, the better the solution perform. So our objective is to minimize score.

3.2.2 Constraints

1. The first constraint is each paper should be reviewed by 3 referees, when i = 1,2,3,...71

$$\sum_{j=1}^{21} A_{i,j} = 3 \tag{4}$$

2. The second constraint is each referee should review roughly the same number of papers. It is impossible to let them review exactly the same, so here i just set some telerance threshold.

for any i,j in 1,2,3,...,21

$$|\sum_{k=1}^{71} A_{k,i} - \sum_{k=1}^{71} A_{k,j}| \leq 2 \tag{5}$$

3. Can not assign a paper to a referee that has a conflict.
for any i = 1 to 71, j = 1 to 21

$$P_{i,j} \cdot A_{i,j} \neq 3 \tag{6}$$

4 Experiment and Result

Programming this problem with formuation and solve it with gurobi. we get a score of 90. And the smaller the score is, the better the solution is.

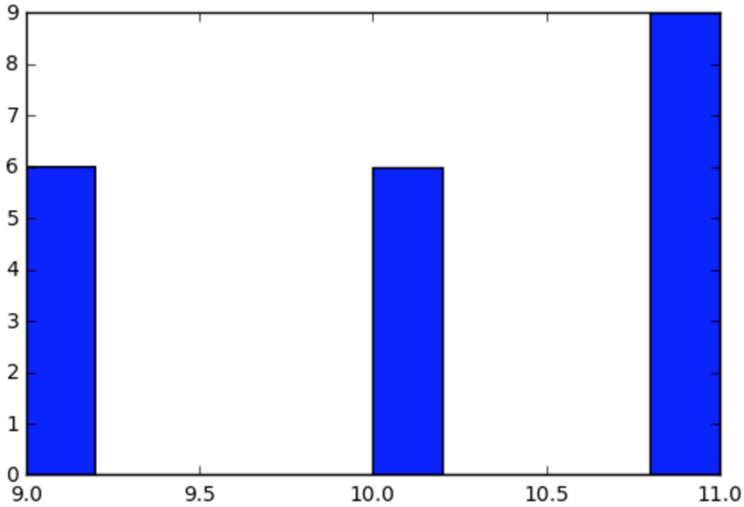


Figure 1: referee assignments histogram.

we can see that there are 6 referees who are assigned 9 papers. there are 6 referees who are assigned 10 papers, there are 9 referees who are assigned 11 papers. Although everyone do not review the same number of papers, but it is balanced.

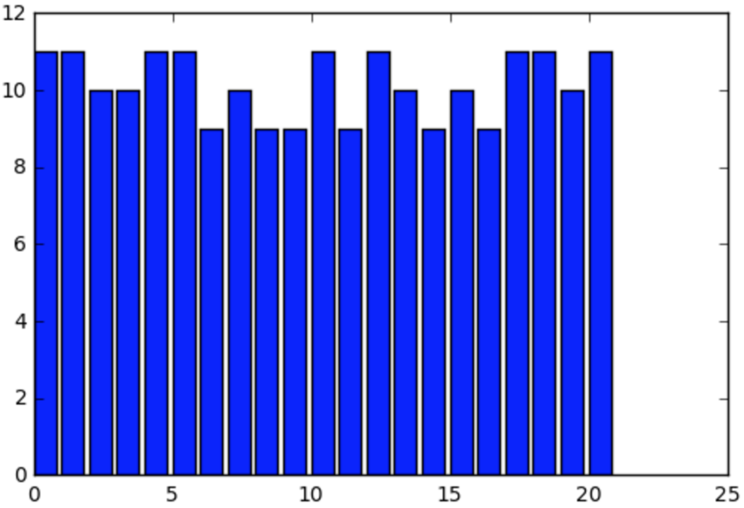
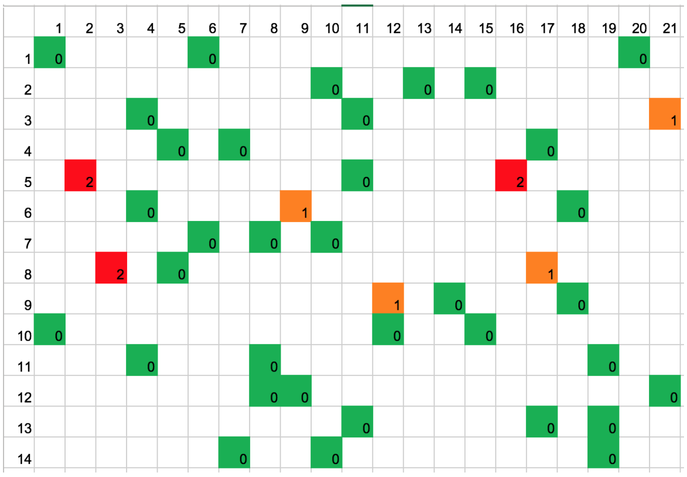


Figure 2: referee assignments bar chart.

below is the final assignment output of the first 14 papers, for the entire output, please see attached figure. In the output xls file, green grid mean the referee is assigned the paper that he definitely can review, yellow grid mean the referee maybe can review the paper. Red grid mean the referee is assigned paper that he do not want to review.

The output make sure in each row there is only 3 color grid, means that each paper need 3 referees to review. And there is no conflict grid, there is as least red grid as possible.

There is not a definite answer to this problem, because we set the tolerance of variance of number of paper each referee assigned to 2, and got the best score of 90. Obviously, if we set the this tolerance value a little bit bigger, we can reach better score. But it also means some referee will have to review 3 more papers than the other referee. So there is a trade off here.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	🟩 0					🟩 0														🟩 0	
2										🟩 0			🟩 0		🟩 0						
3				🟩 0							🟩 0										🟧 1
4					🟩 0		🟩 0										🟩 0				
5		🟥 2									🟩 0					🟥 2					
6				🟩 0					🟧 1									🟩 0			
7						🟩 0		🟩 0		🟩 0											
8			🟥 2		🟩 0												🟧 1				
9												🟧 1		🟩 0				🟩 0			
10	🟩 0											🟩 0			🟩 0						
11				🟩 0				🟩 0											🟩 0		
12								🟩 0	🟩 0												🟩 0
13											🟩 0						🟩 0		🟩 0		
14							🟩 0			🟩 0									🟩 0		

Figure 3: referee assignments output.

5 Apprendix

```
%matplotlib inline
from gurobipy import *
import xlrd
from numpy import *
from matplotlib import pyplot as plt

# there are 71 papers and 21 referees

noPapers = 71
noReferees = 21

# read data from xlsx

workbook = xlrd.open_workbook('paper_preferences.xlsx')
worksheet = workbook.sheet_by_index(2)
preferences = [[0 for _ in range(noReferees)] for __ in range(noPapers)]

for i in range(1, noPapers+1):
    for j in range(1, noReferees+1):
        preferences[i-1][j-1] = worksheet.cell_value(i,j)

map = { 'yes':0, 'maybe':1, 'no':2, 'conflict':3 }
for i in range(noPapers):
    for j in range(noReferees):
        preferences[i][j] = map[preferences[i][j]]
```

```
# create a model
model = Model('assignment')

# create decision variables and store them in the array assignments
assignments = [[0 for _ in range(noReferees)] for __ in range(noPapers)]
for i in range(noPapers):
    for j in range(noReferees):
        curVar = model.addVar( vtype = GRB.BINARY, \
                               name = "x" + "("+ str(i) + "," + str(j) + ")")
        assignments[i][j] = curVar
model.update()


# constraits
# constrait 1 : each paper gets reviewed by 3 referees
# sum up each rows of assignments equals exactly to 3

for i in range(noPapers):
    constExpr = LinExpr()
    for j in range(noReferees):
        curVar  = assignments[i][j]
        constExpr += 1 * curVar
    model.addConstr( lhs = constExpr , sense = GRB.EQUAL, rhs = 3, \
                     name = "p" + str(i) )

# constrain 2 : each referee should review roughly the same number of papers.
index = 0
for i in range(noReferees-1):
    for j in range(i+1, noReferees):
        constExpr = LinExpr()
        index += 1
        for k in range(noPapers):
            left = assignments[k][i]
            right = assignments[k][j]
            constExpr += 1 * left
            constExpr += -1 * right
            model.addRange( constExpr, -2.0, 2.0, "Range" + str(index) )

# constrain 3 : don't assign the conflict paper to referee
index = 0
for i in range(noPapers):
    for j in range(noReferees):
        index += 1
        constExpr = LinExpr()
        constExpr += preferences[i][j] * assignments[i][j]
        model.addConstr( lhs = constExpr , sense = GRB.LESS_EQUAL, rhs = 2, \
                     name = "uu" + str(index) )

model.update()
model.write( filename = "Output.lp" )

model.optimize()
```