Discrete Structures of Computer Science
Project–Faculty Assignment

Aditya Chaudhary(2022A7PS0622G)
Pray Raskapoorwala(2022A7PS1239G)
Aditya Gupta(2022A7PS0090G)

28 November 2023

1 Problem Statement

The problem at hand involves the faculty allocation problem,which aims to allocate courses to faculty based on their preferences and workload while also making sure that all the courses have been alloted and **no course remains empty,if the course is a Compoulsary Disciplinary Elective(CDC),while it may or may not be alloted if it is an elective** Through the allocation,we need to make sure that maximum courses are alloted according to the preferences of the professor. Some of the given constraints are:

- The professors are divided into three categories-The first category can take 0.5(half) of a course,the second category professors can take 1(full) course and the third category can take 1.5 courses(i.e. either half,full or 1.5).
- There is no prioritization between professors in the same category

2 Assumptions

- The preferences of the professors are multiplied by 2 since we cannot compare two **floating point numbers** for equality.
- We are assuming that all CDC's are alloted,and the sum of all the workloads of the professors should be equal to the total number of courses available,i.e.

$$\sum_i \textbf{Workload}(i) = \textbf{Total number of courses}$$

3 Approach

:

The approach we used was **linear programming and making constraints from Matrix Manipulation**. The steps include:

We have given two codes for CDCs and Electives as there is a slight change in their constraints.
We assumed there are n professors and each professor gives preference for n courses involves some CDC's and electives
To begin with a smaller testcase,we assumed a matrix of order 4 wherein the preferences for the courses were listed in the corresponding rows,i.e.

Preference Matrix= $\begin{bmatrix} Prof/Course & CourseA & CourseB & CourseC & CourseD \\ Prof1 & 1 & 1 & 0 & 0 \\ Prof2 & 0 & 2 & 2 & 0 \\ Prof3 & 3 & 0 & 0 & 3 \\ Prof4 & 0 & 2 & 2 & 0 \end{bmatrix}$

- – The courses are represented by A, B, C, and D, and the preferences are given by:
  - $*$ 1 if a professor of the first category has given preference for the course.
  - $*$ 2 if a professor of the second category has given preference for the course.
  - $*$ 3 if a professor of the third category has given preference for the course.
  - $*$ 0 if a professor does not give preference for the course.

- Then we multiplied the above matrix with:

$$VariableMatrix = \begin{bmatrix} w1 & w2 & w3 & w4 \\ x1 & x2 & x3 & x4 \\ y1 & y2 & y3 & y4 \\ z1 & z2 & z3 & z4 \end{bmatrix}$$

- For the above matrix,we introduced some dummy variables, which are:(**Note: All the variables will only take integer value for solution)**
  - W_i for faculty i giving preference to course A,
  - X_i for faculty i giving preference to course B,
  - Y_i for faculty i giving preference to course C,
  - Z_i for faculty i giving preference to course D.

  Subject to the constraint: $\sum_{j=1}^{4} A_{i,j} = 2$ for each row i.

- The obtained product is $\begin{bmatrix} w1 & w2 & w3 & w4 \\ x1 & x2 & x3 & x4 \\ y1 & y2 & y3 & y4 \\ z1 & z2 & z3 & z4 \end{bmatrix}$
- Then,we formulated some linear equations to satisfy the given constraints and provided a solution.
- The equations are:
- $w1 + w2 = 1$

- $x2 + x3 = 2$
- $z2 + z3 = 2$
- $y1 + y4 = 3$
- $w1 + y1 = 2$
- $w2 + x2 + z2 = 2$
- $y4 = 2$
- $x3 + z3 = 2$
- These equations can be solved using Linear Algebra; there are multiple libraries in Python to solve linear Algebra Equations using different Constraints (Eg-Pulp). For this project, we are using Google OR-Tools library: To Install Google Or-Tools:

  python3 -m pip install -U –user ortools

- Then, inputs for the number of courses and the professors' preferences were taken as run-time user input. The input is given at runtime, but we have introduced a jpg representing the user input taken for the particular case to read that input easily. Correspondingly, the output.txt file was made to assign the courses to the professors and to find the number of solutions(if the optimal solution is possible).
- We then modeled this logic using some libraries in the ORTools to get all combinations and also to find if the solution is optimal or not

4 Crash Test

- The crash test refers to a test case where the code would not work optimally.
- Here, the crash test would be a case where all the courses have not been allotted to a professor or a case where the total number of courses taken by a professor exceeds his course limit.
- Also, the code takes some time to run if the number of CDCs/Electives is more than 10.
- Another aspect where the code might not work would be invalid inputs,i.e., the user might enter invalid inputs(e.g., a string instead of an integer in the course load of the professor). Now, we examine some cases:

4.1 Crash Test 1

```
Enter the value for n: 3
Enter course number 0: CSE
Enter course number 1: History
Enter course number 2: Maths
Enter faculty ID: Faculty 1
Enter faculty course limit: 3
Does faculty Faculty 1 prefer course 1? (Y/N): Y
Does faculty Faculty 1 prefer course 2? (Y/N): Y
Does faculty Faculty 1 prefer course 3? (Y/N): Y
Enter faculty ID: Faculty 2
Enter faculty course limit: 3
Does faculty Faculty 2 prefer course 1? (Y/N): N
Does faculty Faculty 2 prefer course 2? (Y/N): Y
Does faculty Faculty 2 prefer course 3? (Y/N): N
Enter faculty ID: Faculty 3
Enter faculty course limit: 2
Does faculty Faculty 3 prefer course 1? (Y/N): Y
Does faculty Faculty 3 prefer course 2? (Y/N): N
Does faculty Faculty 3 prefer course 3? (Y/N): N
Status = INFEASIBLE
Number of solutions found: 0
```

- As we can see the status of our solution is INFEASIBLE and we cannot get any possible combinations.This is because the **workload has exceeded the total number of courses(which contradicts our assumption and hence infeasible)**

4.2 Crash Test 2

```
pray@Prays-MacBook-Pro DISCO PROJECT % /usr/local/bin/python3 "/Users/pray/Documents/DISCO PR
de/CDC.py"
Enter the value for n: 4
Enter course number 0: A
Enter course number 1: B
Enter course number 2: C
Enter course number 3: D
Enter faculty ID: Fac_1
Enter faculty course limit: d
Traceback (most recent call last):
  File "/Users/pray/Documents/DISCO PROJECT/CDC Python Code/CDC.py", line 63, in <module>
    faculty_Courselimit.append(int(input("Enter faculty course limit: ")))
                               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
ValueError: invalid literal for int() with base 10: 'd'
```

Here,we can see that the input added is a string instead of an integer,so we are getting an error

5 Consistency Report

5.1 Testing for CDC's

5.1.1 Case 1

For the given testcase:

- $\begin{bmatrix} Professor/Course & CSF214-LCS & CSF222-DISCO & CSF215-DD & CSF213-OOP \\ Professor1 & 1 & 1 & 0 & 0 \\ Professor2 & 0 & 2 & 2 & 0 \\ Professor3 & 3 & 0 & 0 & 3 \\ Professor4 & 0 & 2 & 2 & 0 \end{bmatrix}$ we got **3 optimal solutions**,which is consistent with the answer we got while physically iterating and generating the outputs.Those solutions are

```
Status = OPTIMAL
Number of solutions found: 3
```

Status of the output

output.csv

```
FACULTY1 will take Course :
CS214 LCS
FACULTY2 will take Course :
CS215 DD
FACULTY3 will take Course :
CS214 LCS CS213 OOP
FACULTY4 will take Course :
CS222 DISCO

FACULTY1 will take Course :
CS214 LCS
FACULTY2 will take Course :
CS222 DISCO CS215 DD
FACULTY3 will take Course :
CS214 LCS CS213 OOP
FACULTY4 will take Course :
CS222 DISCO CS215 DD

FACULTY1 will take Course :
CS214 LCS
FACULTY2 will take Course :
CS222 DISCO
FACULTY3 will take Course :
CS214 LCS CS213 OOP
FACULTY4 will take Course :
CS215 DD
```

All possible combinations

Figure 1: Case of 4 CDC's

5.1.2 Case 2

We now try to take a more complex example where the number of professors and courses are 8 and try to find the solution:

		CS 214 LCS	CS 222 DISCO	CS 215 DD	CS 213 OOP	CS F211 DSA	CS F241 MUP	CS F212 DB	CS F301 PoPL
Faculty 1	1	N	N	N	Y	N	Y	Y	Y
Faculty 2	2	Y	N	Y	Y	N	Y	N	N
Faculty 3	3	Y	N	Y	N	N	N	Y	Y
Faculty 4	3	Y	N	Y	N	Y	N	Y	N
Faculty 5	2	N	Y	N	N	Y	Y	N	Y
Faculty 6	1	N	Y	N	Y	N	Y	N	Y
Faculty 7	2	Y	Y	N	Y	N	N	N	Y
Faculty 8	2	N	N	Y	Y	N	Y	Y	N

Input matrix

```
Status = OPTIMAL
Number of solutions found: 8066
```

```
120565   Faculty 1 will take Course :
120566   CS 213 OOP
120567   Faculty 2 will take Course :
120568   CS 215 DD CS F241 MUP
120569   Faculty 3 will take Course :
120570   CS 214 LCS CS F212 DBMS
120571   Faculty 4 will take Course :
120572   CS 215 DD CS F211 DSA
120573   Faculty 5 will take Course :
120574   CS 222 DISCO
120575   Faculty 6 will take Course :
120576   CS 213 OOP
120577   Faculty 7 will take Course :
120578   CS F301 PoPL
120579   Faculty 8 will take Course :
120580   CS F241 MUP CS F212 DBMS
120581
```

Hence,we see that as the number of professors increases the total possible course assignments also increase.

5.2 Testing for electives

-The constraint to be kept in mind is that a few electives may not be assigned to any of the professors.

5.2.1 Case 1

Considering 2 electives:

$$\begin{bmatrix} Professor/Course & BITSF312-NeuralNetworksandFuzzyLogic & BITSF343-FuzzyLogicandApplications \\ Professor1 & 2 & 0 \\ Professor2 & 0 & 3 \end{bmatrix}$$

```
Faculty 1 will take Course :

Faculty 2 will take Course :


Faculty 1 will take Course :

Faculty 2 will take Course :
BITS F343 FUZZY LOGIC & APPL

Faculty 1 will take Course :
BITS F312 NEURAL NET & FUZZY LOGIC
Faculty 2 will take Course :


Faculty 1 will take Course :
BITS F312 NEURAL NET & FUZZY LOGIC
Faculty 2 will take Course :
BITS F343 FUZZY LOGIC & APPL
```

The output obtained is:

```
Status = OPTIMAL
Number of solutions found: 4
```

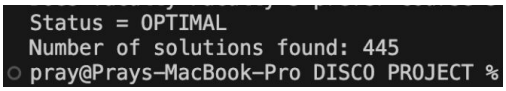
5.2.2 Case 2

Here, we introduce another example of 5 electives.

		BITS F311 In	BITS F312 NEU	BITS F343 FU	BITS F364 HU	BITS F386 QU
Faculty 1	1	N	N	Y	Y	N
Faculty 2	2	N	N	Y	Y	N
Faculty 3	3	Y	Y	N	N	N
Faculty 4	3	Y	N	Y	N	Y
Faculty 5	2	N	Y	N	N	Y

Input matrix

```
4742    Faculty 1 will take Course :
4743
4744    Faculty 2 will take Course :
4745    BITS F343 FUZZY LOGIC & APPL BITS F364 HUMAN COMP INTERACTION
4746    Faculty 3 will take Course :
4747    BITS F311 Image Processing BITS F312 NEURAL NET & FUZZY LOGIC
4748    Faculty 4 will take Course :
4749    BITS F343 FUZZY LOGIC & APPL
4750    Faculty 5 will take Course :
4751    BITS F312 NEURAL NET & FUZZY LOGIC BITS F386 QUANTUM INFO & COMPUTING
4752
4753    Faculty 1 will take Course :
4754
4755    Faculty 2 will take Course :
4756    BITS F364 HUMAN COMP INTERACTION
4757    Faculty 3 will take Course :
4758    BITS F311 Image Processing BITS F312 NEURAL NET & FUZZY LOGIC
4759    Faculty 4 will take Course :
4760    BITS F343 FUZZY LOGIC & APPL
4761    Faculty 5 will take Course :
4762    BITS F312 NEURAL NET & FUZZY LOGIC BITS F386 QUANTUM INFO & COMPUTING
```



6 Conclusion

Hence,we see that the code is consistent for a variety of testcases for both CDC's as well as Electives. We would like to thank Prof. Snehanshu Saha for this opportunity to do this project which made us understand many new things and improved our problem solving skills