University Course Allotment

Anuj Shikarkhane, Jenil Shah, Yogesh Singla

November 2023

1 Introduction

The modern university course assignment system poses a unique challenge of optimizing faculty course allocations while respecting individual preferences and category-based constraints. This research project delves into developing an innovative assignment scheme within a department where professors are categorized into three distinct groups: "x1," "x2," and "x3," each with course loads 0.5, 1 and 1.5, respectively. Each professor submits their preference list, and no priority is considered among professors within the same category.

2 Objective

The primary objective of this research is to maximize the number of courses assigned to faculty members, aligning with their preferences and category-based constraints. The challenge lies in developing an assignment scheme that ensures courses are only assigned to faculty members if they appear in their preference list while respecting the category-based constraints.

3 Problem Statement

This study aims to refine the course assignment process at a university by optimizing how courses are allocated to faculty members. Faculty are grouped into three categories based on their course load capacity: 'x1' can teach half a course, 'x2' a full course, and 'x3' one and a half courses per semester. Faculty have the option to co-teach courses, which splits the course load equally between them, and each faculty member has a ranked list of preferred courses they wish to teach, with no priority given to any individual within the same category.

The goal is to create an assignment system that maximizes the alignment of course assignments with faculty preferences and workload capacities. The challenge is to assign courses exclusively to faculty members who have listed those courses as preferred. The uniqueness of the problem lies in the flexibility of course loads and the potential for adapting the system to better fit faculty preferences, such as modifying the maximum courses allowed or expanding the faculty categories for a more generalized solution.

4 Assumptions

1. The allocation starts with x1 category professors first, followed by x2 and x3.

2. If two profs have same preference for a course, then it will be allocated to the one who appears earlier in the serial order.

3. If it is not possible to allocate courses to a professor commensurate with their category weight, then the program allocates courses according to whatever is feasible to reach maximum allocation. For example, it can allocate just 1 course to professor of 1.5 course weight if it there is no other way to allocate 1.5 courses. This is not a shortcoming of the code, rather it's because of the limitations of the preference list submitted by the professor.

4. Our program doesn't segregate courses department wise. This means, for example, a CS professor can be allocated a Humanities course too. Any professor can submit preferences for any course, and any course can be

allocated to any professor.

5. CDC's are always allocated before elective courses.

5 Methodology

Professors are divided into 3 categories: x1, x2 and x3 and stored into an array list. Similarly, courses are divided into 2 categories: CDC and EL as array lists. Every course has property "courseAvailable" associated with it which indicates how much of the course is available initialised with default 1.

Methods for allocating courses to professors include various priorities which are as follows: 1. Professor wise 2. Category wise. 3. Course wise The method acquired by us does a more rigorous prioritisation to courses followed by category and professors.

We perform the assignment for x1 category professors first, followed by x2 and x3.

A loop iterates over all the CDC courses according to the ascending order of the serial number in which they appear. Then the program checks if that course is in the of category x1 professors. If the course is listed as 1st preference of only 1 professor, then it is immediately half allocated. If it is listed as 1st preference of multiple professors, then it is allocated to the two professors who comes earlier in the serial order of professors. Such a course is now fully allocated. Now consider the course that is half allocated. The program will then try to assign the course for x2 profs according to their preferences. If the half allocated course appears as 1st preference for multiple professors, then it will allocate to the one who comes earlier in the serial order. If It could not be allocated to any professor from x2, then it will try to do it from x3 and similar process as x2 will be followed. It is important to note that the design of our code is such that a course is either fully allocated or not allocated at all. If not allocated at all, it will be removed from the list of courses(unallocable).

The whole process is repeated for elective courses list.

6 Conclusion

We have followed a purely functional approach based on a rule based logic to solve this problem. The program makes an attempt to maximize the number of courses allocated while also rendering some courses as unallocable to any professor. No course can be left half allocated. It is either fully allocated or not allocated at all. The first attempt is to always allocate CDC's as they are the compulsory courses.

Our code is easy to understand, robust and can be suitably modified to accommodate new requirements and constraints.

In summary, the proposed optimization of the University Course Assignment System effectively balances faculty preferences with their teaching loads. The innovative categorization and shared teaching models promise enhanced satisfaction and equity. Future steps include fine-tuning the system based on empirical results and considering its applicability across various academic departments. This model paves the way for a more integrated and adaptable approach to academic course assignment.