M-M PROFESSOR ASSIGNMENT PROBLEM

Anirudh Jayakishan - 2021B3A70979G
Jai Bhobe - 2021B3A71024G
Varun Khadpe - 2021B3A72864G

November 2023

1 Problem formulation

In the given modified version of the popular assignment problem(in which there are n men, n jobs and numerical ratings are given for each man's performance one each job)(1), the task is as follows- Given the constraints regarding the professors and courses, the number of courses assigned must be maximized while also ensuring that no professor is unassigned. The constraints are as follows:
1) There are 3 categories of professors- x1(who can be assigned 0.5 course), x2(who can be assigned 1 course) and x3(who can be assigned 1.5 courses).
2) A professor is said to be assigned 0.5 course when he/she is sharing the course with another professor.
3) There are 4 types of courses- HD CDCs, HD electives, FD CDCs, FD electives.
4) All the CDCs must be assigned while some electives may be left unassigned.
5) Each professor has a preference list consisting of at least 12 courses(4 FD CDCs, 4 FD electives, 2 FD CDCs, 2 FD electives).
The approach used in this paper involves extending the Kuhn–Munkres algorithm and is elaborated upon in the following section.

2 Method adopted

As mentioned above, this problem statement deals with extending the typical $N-N$(one-one) assignment problem to an $M-M$(many-many) assignment.Here, multiple professors can take the same course, while the same course can also be taken by multiple professors. Although this has been dealt with in a previous paper(which modified the algorithm through means of backtracking)(3) we aim to propose an alternate solution which will be sub-optimal but will also satisfy all constraints. The algorithm also runs in polynomial time(2) and as such can be regarded as an efficient solution. It must also be noted that while the $Python$(which is the language the algorithm is coded in) library $scipy$ does contain a $linar\_sum\_assignment$ module which directly applies the Kuhn–Munkres algorithm on a given cost matrix, we chose to be more elaborate with the code to make it easier to understand

2.1 Kuhn–Munkres algorithm

The Kuhn–Munkres algorithm was developed as a means of finding an optimal solution for the N-N assignment problem. Let A be an $n \times n$ cost matrix with $a_{ij} > 0$ representing the cost of assigning agent i to task j.
(1) For all $i \in$ {1, 2, ..., n} and a_{ij} such that $j \in$ {1, 2, ..., n}, set $a_{ij} = a_{ij} - \min_j a_{ij}$. Similarly, for all j

$\in$ {1, 2, ..., n} and a_{ij} such that $i \in$ {1, 2, ..., n}, set $a_{ij} = a_{ij} - \min_i a_{ij}$. Let this new matrix be denoted as A_0.
(2) Find the number of lines, k, through both the rows and columns that ”cover” all of the zeros of A_0.
(3) If $k < n$, let a_0 be equal to the minimal element a_{ij} such that a_{ij} is not covered by any of the k lines. For all uncovered elements a_{ij}, let $a_{ij} = a_{ij} - a_0$, whereas for all twice covered elements let $a_{ij} = a_{ij} + a_0$. With the revised matrix, repeat Step 2.
(4) When $k \geq n$, construct a set of n independent zeros, Γ, where for all a_{ij}, if $a_{ij} \in \Gamma$, $a_{ij} = 1$, else $a_{ij} = 0$. This assignment output, composed of the n independent zeros, is the new (and final) matrix.

2.2 Modifying the algorithm

A few dissimilarities arise when trying to apply the Kuhn–Munkres algorithm to this problem. The first one being that rather than having multiple distinct professors, we are presented with multiple categories of professors with each category having multiple distinct professors.Furthermore, each category of professors can take up different amounts of workloads as mentioned above. In order to account for this each professor is "split" into multiple distinct professors(each of which can accept 0.5 courses) depending upon which category the professor belongs to(i.e. professors of category x1,x2,x3 will be "split" into 1,2 and 3 professors respectively).
The next dissimilarity is that a course can be divided between 2 professors. This is accounted for by "splitting" each course into 2 courses(each representing 0.5 of the actual course).
The final dissimilarity lies in the fact that the cost matrix A mentioned above may not be a square matrix when all electives are considered for allotment. This is remedied by only picking up the most generally preferred electives for the actual allotment process.
It is important to mention first and foremost that the solutions that the algorithm devised in this paper will produce will only be sub-optimal. The search for the optimal assignment is something that can be researched in a future paper but will not be dealt with here. The entire modified algorithm is explained in detail below:
(1) The preferences of each professor are assigned scores (with a higher score being given to a preference that’s lower on the list). These scores will represent the cost of taking that course and will range from 1 to k where k is the number of preferences filled by a particular professor($k >= 12$).
(2) A cost matrix is built with i rows and columns, where $i = n_1 + 2 \cdot n_2 + 3 \cdot n_3$. Here n_1, n_2, and n_3 represent the number of professors of categories x_1, x_2, and x_3 respectively.
(3) Since all the CDCs must be assigned, the cost matrix will contain all the CDCs.
(4) For an optimal assignment to be possible, the cost matrix must be a square matrix. Hence, if $2 \cdot m$ (where m is the number of the CDCs) is less than the i defined above, electives must be picked up as well to fill the cost matrix.
(5) The electives to be chosen for the cost matrix are selected in order of their preference scores, where if they have a lower preference (better understood as cost) score, they will be added to the cost matrix first. This is done until the cost matrix is a square matrix.
(6) The above-explained Kuhn–Munkres algorithm is then run on this cost matrix to obtain our final assignment.
(7) Multiple outputs are obtained by changing the electives that are used in the cost matrix.

3 Citations

(1)Kuhn, H. W. (1956). Variants of the Hungarian method for assignment problems. Naval Research Logistics Quarterly, 3, 253–258.
(2)Munkres, J. (1957, March). Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics, 5(1), 32–38.
(3)Zhu, H., Liu, D., Zhang, S., Zhu, Y., Teng, L., Teng, S. (2016). Solving the Many to Many assignment problem by improving the Kuhn–Munkres algorithm with backtracking. Theoretical Computer Science, 618, 30-41. https://doi.org/10.1016/j.tcs.2016.01.002