Course Schedule Optimization

Mallika Khanna, Jun Hur, Jeremy Wei, Yong Gun Choe

December 18th, 2018

Table of Contents

ABSTRACT

This report investigates the optimal scheduling of classes based on the results from Faculty Course Evaluations (FCE). It focuses on designing and implementing an algorithm to minimize the workload that students carry throughout their undergraduate years while at the same time have them take enough challenging courses to make their degrees worthwhile. In order to be specific and more relevant, we narrowed down the student pool to those pursuing a general Mathematics degree in the Mellon College of Science. In other words, we wanted students in the General Mathematics major to get something useful out of their degrees while minimizing the stress imposed on them. Therefore, drawing a fine line between a class that is considered time consuming and useful was a key component in determining the worth of the classes. The first sections of the paper will discuss the overview of the problem and the process we used to approach the problem. Then, the paper delves into the actual solving of the optimization problem through Integer Programming and Greedy Algorithm approaches. The final parts of the paper outlines the results, potential errors, and further improvements that could be made should the project be taken beyond the scope of this class.

INTRODUCTION

Carnegie Mellon University is renowned for its heavy workload and rigorous class schedules; it is well documented that CMU students, especially those majoring in mathematics, spend a lot of time on their classes. In order to reduce the stress for these students and help them live a more productive college life, the goal is to come up with an optimal schedule that fulfills all the requirements for graduation while simultaneously maintaining the challenging nature of a mathematics degree. For the simplicity of the study, we will limit the mathematics curriculum to the general mathematics concentration. Our plan is to have students take all the required core classes, choose from a list of depth electives that give the mathematics degree some value and take the easiest general education electives since they are not as relevant, and contribute little to a mathematics degree. We will use a valuation system by using the overall scores from FCEs to evaluate the depth electives based on their worth and then use Integer Programming to determine which depth electives to take. We will then implement a Greedy Algorithm to choose the general education electives that required the least number of hours per week by looking at the results from FCEs. Overall, our goal is to come up with a concrete schedule that spans 8 semesters, which is equivalent to the length of a full academic undergraduate degree.

Courses & Requirements

Math depth electives
21-270
21-292
21-300
21-301
21-329
21-344
21-369
21-370
21-371
21-374
21-378
21-393
21-420
21-441
21-484

Table 1: Math Depth Electives.

Stats/Physics/CS depth electives					
36-303	33-339	15-213	15-349	15-411	15-465
36-304	33-340	15-214	15-351	15-412	15-466
36-309	33-341	15-251	15-354	15-415	15-487
36-314	33-342	15-259	15-355	15-418	15-491
36-315	33-353	15-292	15-365	15-421	15-492
36-326	33-355	15-295	15-381	15-423	15-494
36-350	33-398	15-300	15-382	15-437	
36-401	33-441	15-312	15-383	15-440	
36-402	33-444	15-313	15-385	15-441	
36-410	33-445	15-314	15-386	15-445	
36-428	33-446	15-316	15-387	15-449	
36-459	33-448	15-317	15-390	15-451	
36-490	33-456	15-322	15-392	15-458	
33-331	33-466	15-323	15-400	15-462	
33-332	33-467	15-330	15-405	15-463	
33-338	15-210	15-348	15-410	15-464	

Table 2: Statistics/Physics/Computer Science Depth Electives

TYPE	Core Classes	Math Depth Electives	Other Depth Electives (sample)	General Education Electives
COURSE #	21-120	21-270	36-350	A lot...
	21-122	21-292	36-401	
	21-127	21-300	36-410	
	21-228	21-301	33-331	
	21-241	21-329	33-341	
	36-225	21-344	15-210	
	21-259	21-369	15-213	
	21-260	21-370	15-214	
	21-341	21-371	15-251	
	21-355	21-374	15-300	
	21-356	21-378	15-410	
	21-373	21-393	15-411	
		21-420	15-412	
		21-441	15-415	
		21-484	15-418	
CHOICE	12 classes from 12	5 to 8 classes from 15	0 to 3 classes from 87	Combination of 114 units
METHOD	Must Take All	Value System	Value System	Greedy Algorithm

Table 3: Overview of Math Graduation Requirements

TECHNICAL ASSUMPTIONS

In order to tailor this problem into the scope of this course, there were several technical assumptions we had to consider. Most importantly, we only consider the perspective of a single Carnegie Mellon University student majoring in Mathematical Sciences, specifically general mathematics track. Furthermore, this student does not have any other additional major nor a minor so that we can consider only one set of graduation requirements. Addition-ally, the student does not have any courses from high school that can substitute for required courses in CMU (i.e. AP exam scores, IB-diploma programs). We will further assume that the student will always complete courses with a grade C or better, without dropping or retaking the course, so that the prerequisite requirements will always be met.

We also did not consider the flexibility of graduation requirements. For instance, we assumed that no classes could double count and satisfy multiple graduation requirements. Another assumption we made is that prerequisites must be completed before and could not be taken simultaneously as co-requisites. In addition, we disregarded any courses that are offered intermittently and courses that could be replaced by similar alternative with an approval from the academic advisor, as these courses are difficult to predict and account for.

There were more serious simplifications of course registration and graduation requirement logistics. Although it is not always possible, we assumed that the student will always be guaranteed to register for desired classes. This further entails that the student either

1. Has a reservation in the course,
2. Will always be removed from the waitlist
3. Will always be registered during course registration week

Additionally, we assume that all the courses the student wishes to take for a particular semester do not have any scheduling conflicts. These two assumptions are the two biggest assumptions that deviates from real situations. However, the fact that course registration and schedule conflicts are unpredictable makes the problem itself difficult to solve accurately.

Although the assumptions are not trivial, we claim that the simplified version of our problem is still meaningful in that the solution to this problem will give provide a general overview of what to expect during four years in CMU. The student can and should appropriately adapt to different circumstances and account for potential unforeseen circumstances prior to and during course registration.

Modeling The Data

Primary Model - Integer Programming

The first model we consider is an integer programming model. We used the Integer Programming model to schedule the core classes and the mathematical depth electives.

Variables:

$$x_{ij} = \begin{cases} 1 & \texttt{if class i is taken in semester j} \\ 0 & \texttt{otherwise} \end{cases}$$

$$h_i = \texttt{FCE hours}, s_i = \texttt{FCE scores}, u_i = \texttt{\# of units}$$

Sets:

$$F = \{\texttt{classes offered in Fall i.e. semesters 1,3,5,7}\}$$

$$S = \{\texttt{classes offered in Spring i.e. semesters 2,4,6,8}\}$$

$$F \cap S = \texttt{ classes offered in both fall and spring semesters}$$

$$\mathbb{C} = \{i : \texttt{class } i \in \texttt{Core Classes}\}$$

$$\mathbb{P} = \{(i, i') : \texttt{Class i' is a prerequisite for class i}\}$$

$$\mathbb{N} = \{(i, i', i'') : \texttt{substitutable classes}\}$$

Objective function: Minimize $\sum_i h_i x_{ij} \forall j \in \{1, 2, ..., 8\}$

Subject to:

1. $\sum_i x_{ij} \leq 1 \ \forall \ \texttt{classes} \ i$. This ensures each class is taken at most once.
2. $x_{ij} \leq x_{i'j'} \forall j' < j$ and $(i, i') \in \mathbb{P}$. This ensures prerequisites are satisfied.
3. $36 \leq \sum_i u_i x_{ij} \leq 54 \forall j$. This ensures we stay above the minimum and below the maximum number of units. Note: we don't allow for overloading.
4. $\sum_j x_{ij} + x_{i'j} + x_{i''j} \leq 1$. This ensures only 1 substitutable is taken.
5. $\sum_i s_i x_{ij} \geq S_{min}$ This ensures we surpass some minimum value for the courses as measured by FCE scores.
6. $\sum_j x_{ij} = 1 \forall i \in \mathbb{C}$. This ensures we take all the core classes.
7. $\sum_{i \notin F} x_{ij} = 0 \forall j \in 1, 3, 5, 7$. This ensures that in the Spring, no Fall courses are taken.
8. $\sum_{i \notin S} x_{ij} = 0 \forall j \in 2, 4, 6, 8$. This ensures that in the Fall, no Spring classes are taken.

Secondary Model - Greedy Algorithm

Since there are much more general education classes than core and depth elective classes using integer programming became infeasible. Thus, we turned to a greedy approach that would get us close to an optimal schedule for general education classes. We first greedily chose the general education classes that had the least FCE hours. Then, for each pair of classes, we swapped them with each other. If a swap produced a set of general education classes with less FCE hours, we repeated the swapping process on the more optimal schedule. We repeatedly did this swapping until we arrived at a schedule where swapping any pair of classes would not yield better result.

MODEL IMPLEMENTATION

The aforementioned integer programming model has around 800 variables because there are more than 120 classes excluding general education and each class has 8 variables, one for each semester. Microsoft Excel solver was used to implement the model which only has a limit of 200 variables. Thus, to create a sample solution, it was imperative to reduce the variables, thus three approaches:

1. Reduced the number of variables to make solution feasible. This was done by only considering 4 out of the 8 semesters and only scheduling one required General Education elective and another sample General Education elective.
2. Filtered elective classes by score value, specifically, 4.50. After, the classes were sorted, and then were included in list on which the greedy algorithm was applied.
3. After the two were applied, the problem was treated as an IP and then was solved on solver ensuring all sets and pre-requisites.

Core Classes	Units	FCE Hours	FCE Scores	Required
21120	10	12.8	3.718	1
21122	10	8.3	3.62	1
21127	9	11.9	4.33	1
21128	9	14	4.75	1
21228	9	9.6	4.54	1
21241	10	9.5	3.946	1
36225	9	9	4.22	1
21259	9	5.1	3.19	1
21260	9	10.5	4.115	1
21341	9	8.5	4.22	1
21355	9	7.2	4.29	1
21356	9	8.8	4	1
21373	9	7.7	3.97	1
21270	9	9.4	4.92	1
21292	9	7.7	3.11	1
21300	9	9.1	4.05	1
21301	9	9.1	4.68	1
21329	9	7.5	2.6	1
21369	12	8.9	3.96	1
21370	9	9	4.17	1
21371	9	9	4	1
21374	9	10	4.5	1
21378	9	9	4.86	1
21393	9	9	3.13	1
21420	9	11.2	4	1
21441	9	9	5	1
21484	9	9	4.33	1
76101	9	9	4.23	1
70100	9	7.3	4.56	1
Units Scheduled	55	47	54	45
FCE Hours	57.7	45.8	49.4	44.6

Table 4: Courses and Matching Data Representation

The above table represents the data of units and FCE scores for the selected classes which were used as variables in the model. It also shows the unit and requirement constraints. Since all these classes are offered in both semesters, the model did not require an additional Spring or Fall set pre-requisite.

RESULT & IMPLEMENTATION OF SOLUTION

The result of this model is a sample schedule for four semesters including math depth electives and general education requirements. The model was optimized for 29 classes, which cover more than half the units required. Since this model required a lot of local minimization (over global), it is essential to know that this is a floating dynamic model that will need iterations, till the best one is found. However, the current sample result is very promising.

Core Classes	Semester 1	Semester 2	Semester 3	Semester 4	Number of times class taken
21120	1	0	0	0	1
21122	0	1	0	0	1
21127	1	0	0	0	1
21128	0	0	0	0	0
21228	0	1	0	0	1
21241	0	1	0	0	1
36225	1	0	0	0	1
21259	0	0	1	0	1
21260	0	0	1	0	1
21341	0	0	1	0	1
21355	0	0	1	0	1
21356	0	0	0	1	1
21373	0	0	0	1	1
21270	0	1	0	0	1
21292	1	0	0	0	1
21300	0	0	1	0	1
21301	0	0	0	1	1
21329	0	0	0	0	0
21369	0	0	0	0	0
21370	0	1	0	0	1
21371	0	0	1	0	1
21374	0	0	0	1	1
21378	0	0	0	0	0
21393	0	0	0	0	0
21420	0	0	0	0	0
21441	0	0	0	0	0
21484	0	0	0	1	1
76101	1	0	0	0	1
70100	1	0	0	0	1

Semester 1	Semester 2	Semester 3	Semester 4
21-120	21-122	21-259	21-356
21-127	21-228	21-260	21-373
36-225	21-241	21-341	21-301
21-292	21-270	21-355	21-374
76-101	21-370	21-300	21-484
70-100		21-371	

Table 5: Solution Model

The Excel model has optimized over FCE hours with limits to units which can be scheduled and as can be seen below, FCE hours (197.5hrs) are lower than units schedules (201).

Scheduled FCE hours	197.5
Units Summed	201

Table 6: Units vs. FCE hours of our schedule

The above result is similar to the sample schedule mentioned on the CMU course website.

Potential Errors & Future Vision

Although our solution model has presented a meaning outcome, our solution is not the optimal solution to be used in real life because of the modifications we made to the original problem. The most deviating factor is the fact that we disregarded any potential class conflicts and waitlist probabilities. Such errors are impossible to deal with at this point, so we decided not to consider them at all. However, other more minor potential errors could be easier to deal with. We could have more carefully structured out which course can be used as a co-requisite. Then it would have been possible to output a better solution with certain courses taken simultaneously in one semester; one used as a co-requisite for another.

Another error that we might potentially face in the future comes from modifications to the graduation requirements. We have noticed that the course catalog has been updated every year. Although catalogs are not very different every year, there were some years where graduation requirement changed significantly. Because our solution model uses the most recent catalog, we might potentially encounter errors when the course catalog gets updated.

If we had more time, we would have chosen a different method to solve our integer programming problem. We faced few limitations of Microsoft Excel. Our problem was too big for Excel to handle. Alternatively, we could have looked into and learn how to use Mathematica. Then we would have been able to produce a full four year schedule of our problem. Additionally, we could have covered larger audience of our model if we also considered different concentrations and majors of other schools. Even further, our project can be expanded to be used in other universities too.

Another potential modification to our project is setting different criteria to optimize. For example, someone might be interested in simply minimizing the hours spent per week. Then, the FCE scores component of the model can be entirely excluded. The problem we would like to solve has variety of application and room for modification, and these were only a small portion of it.

Conclusion

By researching on the topic of optimizing class schedules based on Faculty Course Evaluations, we found a concrete solution to 4 semesters. Excel, due to variable restrictions, could not handle coming up with a schedule for a full undergraduate career (i.e. 8 semesters). Like we predicted, we found that the majority of classes that were chosen for the first 4 semesters were math courses. This reaffirms our goal of taking classes that are 100% relevant to the degree at hand and not taking any unnecessary classes. So, although further improvement and advancement is needed for a more in-depth scheduling, we have established a working algorithm that churns out a schedule that is both time efficient and valuable.

References

http://coursecatalog.web.cmu.edu/melloncollegeofscience/departmentofmathematicalsciences/

https://www.smartevals.com/reporting/StudentsSeeResults.aspx

https://www.math.cmu.edu/-af1p/Teaching/OR2/Notes/ILP.pdf