Incorporating student course preferences into the course scheduling problem

Nishant Dash
Senior, Croton-Harmon High School
Croton on Hudson, NY 10520

Advised by: Amir Ali Ahmadi, Princeton University

October 30, 2021

Abstract

We study the problem of scheduling classes in four-year universities, and also four-year high schools in the United States. This problem is often formalized as the University Course Scheduling Problem (UCSP). There are many research papers on the UCSP, some of which use integer programming (IP) models, e.g., Daskalaki, Birbas, and Housos (2004). In U.S. universities, student preferences for courses are often not explicitly taken into account when constructing class schedules. A 2016 survey of over 700 undergraduate colleges and universities by the American Association of Collegiate Registrars and Admissions Officers (AACRAO) notes that faculty availability and time block popularity were among the most popular factors driving the overall course scheduling process, while "data collected from student plans of study" was the least-influential factor. We compare course scheduling outcomes when student preferences are explicitly taken into account while constructing course schedules and when student preferences are ignored. The quality of course schedules is measured by the number of students getting their first and second-choice electives across three scenarios. In the first, classes are assigned to periods, and teachers to classes while maximizing teacher preferences for teaching times. Students are then assigned to courses in a first-come-first-serve manner. In the second scenario, after course scheduling is completed as in the first scenario, students are assigned to courses while maximizing student preferences for electives. In the third scenario, courses are scheduled so as to maximize student preferences while ignoring teacher preferences. For each scenario, we define an IP model and solve it with CPLEX. We show, using a simplified model of course scheduling constraints, that if student preferences are taken into account, then a course schedule can be constructed that gives students many more of their preferred courses than if the course schedules are constructed without taking preferences into account. The increase is about 20% in a simulated medium-sized department with roughly 20 teachers, 32 courses, and 150 students across four years.

1 Introduction

We study the problem of scheduling classes in four-year universities, and also four year high schools in the United States. This problem is often formalized as the University Course Scheduling Problem (UCSP) or the University Course Timetabling Problem (UCTP). University course scheduling has been studied a lot and there are many research papers on this topic, for example, Daskalaki, Birbas, and Housos [8], Hertz and Robert [10], and Algethami and Laesanklang [1]. The first and third papers above give integer programming models for this problem and use standard solvers such as IBM-CPLEX [11] to solve the models. This problem is also called the Course Timetabling Problem; see Boland et. al. [3]. The UCSP is considered to be a computationally hard problem, as variants of it include the Timetable Design problem which is known to be NP-hard [9]. However, some variants are not hard to solve [13, 5]. For practical issues in university course scheduling and best practices, see the survey [2] by Hanover Research. Another article that surveys common approaches used in universities is [6]. Though the UCSP has been widely studied, it has been recognized (Pillay [12]) that scheduling in four year high schools is less well studied. Pillay [12] surveys research on school timetabling/scheduling.

In U.S. universities, student preferences for courses are often not explicitly taken into account when constructing class schedules. See for example the FAQ on course scheduling at Rutgers University [7]. Instead, university departments look at historical course enrollments to understand student interest in courses, and also take into account faculty preferences and room requirements and availabilities while scheduling courses. A Hanover Research survey [2] states that the American Association of Collegiate Registrars and Admissions Officers (AACRAO) surveyed over 700 undergraduate colleges and universities. According to Hanover Research, the AACRAO

> found that the most popular factors in the overall scheduling process included faculty availability (90.7 percent), time block popularity (76.5 percent), and courses scheduled at the same time from year to year (71.4 percent) (Figure 1.1).8 Notably, the least-influential factor in scheduling for undergraduate students is "driven by data collected from student plans of study," suggesting that relatively few institutions consider student data and plans of study when making course scheduling decisions.

In this work, we study course scheduling when student preferences are explicitly taken into account while constructing course schedules. We note that this is the practice in the Croton-Harmon High School in Westchester County, NY. In this school, high school students give a plan of study (list of courses) at the end of every year. For highly desired electives, students list them in order of preference. The school attempts to satisfy these plans of study and give students their desired courses. From among the ordered list of preferred electives, the school attempts to assign the most preferred elective to a student, and if not possible, the second most preferred elective and so on.

We will compare course schedules where student preferences for electives are taken into account and those where they are not. We make a number of simplifying assumptions regarding room availability and requirements and faculty constraints. We believe our simplified model still captures many constraints used in universities and four-year U.S. high schools. We show, using our simplified model, that if student preferences are taken into account, then a course schedule can be constructed that gives students many more of their preferred courses than if the course schedules are constructed without taking preferences into account.

We define three optimization models. The first model assigns classes to periods, and teachers to classes in a way that scheduling conflicts are taken into account while also maximizing teacher preferences for times at which they teach. The second assigns students to already scheduled classes in such a way that student preferences for classes are maximized. If one runs the second model using the solution of the first, then one obtains a complete scheduling solution, while maximizing teacher preferences. However, in many universities it is common for students to come forward with a list of electives , sorted by preference order, and then classes are given in a first-come first-serve manner. We also test an algorithm to implement such a first-come first-serve assignment. Finally, we write a combined model which creates a complete schedule, along with teacher-course and student-course assignments. The objective function is the maximization of student preferences.

We compare the student preference scores in the three scenarios:

1. Teacher preferences are maximized while scheduling classes, and student preferences are completely ignored. After classes have been scheduled, students are assigned to courses on a first-come first-serve basis. (model 1 + algorithm)

2. Teacher preferences are maximized while scheduling classes, and student preferences are completely ignored. After classes have been scheduled, an optimal allocation of students to courses is computed. (model 1, then model 2)

3. A combined model is solved to schedule classes, assign teachers and students to courses, while maximizing student preferences and ignoring teacher preferences.

We will demonstrate that student preference scores increase from the first to the third scenario (as would be expected). Our main observation is that this increase can be very significant.

2 Integer Programming Models

We will next explain the equations for the three models that we create. We first describe the constraints we incorporate in these models, and then the assumptions behind them, and then the simplifications we make over common practice in order to get a common model for high school and university settings.

In US universities it is common to have courses scheduled either on Monday, Wednesday, and Friday, or on Tuesday and Thursday. We assume that all days are identical in our models which is closer to how US high schools operate. Mandatory classes are often split into multiple sections, however

Symbol	Description
G	set of all grades
S	set of all students
C	set of all courses
T	set of all teachers
D	set of all days
P	set of all periods in a day
i	letter used to denote student index
j	letter used to denote course index
k	letter used to denote teacher index
g	letter used to denote grade index
C_k	set of courses taught by teacher k
T_j	set of teachers that can teach course j
M	set of all mandatory courses
M_g	set of mandatory courses for grade g
E	set of all elective courses
E_g	set of elective courses for grade g
mC	maximum number of courses per teacher
mSC	maximum number of courses per student
Cr_j	number of credits for course j

Table 1: Notation used to describe optimization models

we assume there is only one section and all students in a grade attend the same mandatory class during the same period in the same room. We do not take into account room assignment considerations, but we assume there is an enrollment capacity for each course. We assume that teachers teach up to 2 courses which is a common workload in US universities but not in US high schools. We assume that students only have the option to take elective courses for their year/grade. Mandatory courses for a grade cannot clash and electives cannot clash with mandatory classes but can clash with each other. Teachers cannot be assigned conflicting courses. Multiple periods of the same course cannot be assigned in a day. All courses across the four years are of one type, and teachers can teach only one type, for example, a teacher could teach English for all grades, but cannot teach History. We introduce some notation needed for our models in Table 1.

2.1 Model 1

We have a binary variable $u_{k,j}$ which is 1 if and only if teacher k is assigned to course j. We have a binary variable $x_{j,l,m}$ which is 1 if and only if course j is assigned to period m on day l. We have a binary variable $v_{k,j,l,m}$ which indicates if teacher k has to teach course j on period m on day l.

$$\max \quad \sum_{k\in T}\sum_{j\in C}\sum_{l\in D}\sum_{m\in P} \text{tScore}(k,m) v_{k,j,l,m} \tag{1}$$

$$\text{S.t.} \quad \sum_{j\in C_k} u_{k,j} \leq mC \quad \forall k, \tag{2}$$

$$\sum_{k\in T_j} u_{k,j} = 1 \quad \forall j, \tag{3}$$

$$u_{k,j_1} + u_{k,j_2} + x_{j_1,l,m} + x_{j_2,l,m} \leq 3$$
$$\forall k, j_1 \neq j_2 \in C_k, l \in D, m \in P, \tag{4}$$

$$v_{k,j,l,m} - u_{k,j} \leq 0 \quad \forall k, j \in C_k, l \in D, m \in P, \tag{5}$$

$$v_{k,j,l,m} - x_{j,l,m} \leq 0 \quad \forall k, j \in C_k, l \in D, m \in P, \tag{6}$$

$$\sum_{m\in P} x_{j,l,m} \leq 1 \quad \forall j \in C, l \in D, \tag{7}$$

$$x_{j_1,l,m} + x_{j_2,l,m} \leq 1 \quad \forall g \in G, j_1 \neq j_2 \in M_g, l \in D, m \in P, \tag{8}$$

$$\sum_{l\in D}\sum_{m\in P} x_{j,l,m} = Cr_j \quad \forall j \in C, \tag{9}$$

$$x_{j,l,m}, v_{k,j,l,m}, u_{k,j} \in \{0,1\}. \tag{10}$$

The first constraint gives an upper bound on number of courses taught by each teacher. The second constraint enforces the condition that each course must be taught by one teacher. The third constraint says that if a teacher is assigned to courses $j1$ and $j2$, the courses cannot be assigned to the same period. The next two constraints enforce the condition that variable v is 1 if a teacher is assigned to a course and the course is assigned to a time period. In other words, it indicates if a teacher is assigned to a time period. The next constraint does not allow a course to be assigned to two or more periods in the same day. The next constraint says that at most one mandatory course for a grade can be assigned in a period. The final constraint says that each course must be assigned to as many periods as the number of credits.

2.2 Model 2

The set of all available courses for student i is SC_i. The set of elective courses requested by student i is E_i. The set of all courses assigned to day l and period p is $AC_{l,p}$. The capacity of a course j is CAP_j. We have a binary variable $y_{i,j}$ which is 1 if and only if student i is assigned to course j.

$$\max \quad \sum_{i \in S} \sum_{j \in E_i} \text{sScore}(i, j) y_{i,j} \tag{11}$$

$$\text{Subject to} \quad \sum_{j \in SC_i} y_{i,j} \leq mSC \quad \forall i \in S, \tag{12}$$

$$\sum_{j \in AC_{l,p}} y_{i,j} \leq 1 \quad \forall i \in S, l \in D, m \in P, \tag{13}$$

$$\sum_{j \in M_g} y_{i,j} = |M_g| \quad \text{for all students } i \text{ in grade } g, \tag{14}$$

$$\sum_{i \in S} y_{i,j} \leq CAP_j \quad \forall j \in C, \tag{15}$$

$$y_{i,j} \in \{0, 1\}. \tag{16}$$

The first constraint limits the number of courses each student can take. The second constraint says that a student cannot be assigned to two or more courses during the same period. The third constraint ensures that each student takes all mandatory courses for their grade. The next constraint says that the number of students assigned to each course cannot exceed that course's capacity.

2.3 Combined Model

In the combined model we include all constraints from Model 1 and Model 2 apart from constraint 13 which is replaced my constraint 17. The objective can be a linear combination of objectives for Model 1 and 2, though we mostly use the objective from Model 2.

$$y_{i,j1} + y_{i,j2} + x_{j1,l,m} + x_{j2,l,m} \leq 3 \quad \forall i \in S, j1 \neq j2 \in C, l \in D, m \in P \quad (17)$$

3 Experiments

3.1 Data

In this problem we assume that there are 5 days in a week, there are 7 equal length periods in a day, the number of credits for a course is equal to the number of periods taught per week, and we assume that the schedule repeats weekly so the scheduling problem is effectively solved for a 5 day period. We randomly generated data sets of two different sizes.

In the first group of data sets we have between 100 to 150 students, 32 courses (8 per grade, each grade with 4 mandatory and 4 electives), elective courses have a capacity between 5 and 20 students, students can take up to 3 electives, there are 16 to 24 teachers, each teacher teaches at most 2 courses.

In the second group of data sets we have between 400 to 600 students, 40 courses (10 per grade, each grade with 3 mandatory and 7 electives), elective courses have a capacity between 15 and 30 students, students can take up to 4 electives, there are 20 to 30 teachers, each teacher teaches at most 2 courses.

Each student has a list of electives sorted by preference order. These preferences are encoded by scores with the most preferred elective getting a score of 4, the second most preferred elective gets a score of 3, the third gets a score of 2, and the remaining electives get a score of 1. When students are assigned to courses, the combined student preference score, is calculated by adding up the scores of the electives each student is assigned to.

The teachers have a preference order for the periods in a day in which they would like to teach. The teachers have 1 of 3 preference orders for the periods in which they wish to teach. The first preference order ranks periods in decreasing order from morning to afternoon. The second order is the reverse of the first preference (last period is most preferred and the first period is least preferred). The last preference order gives a higher preference to the middle of the day. These preference orders are encoded by giving a score of 10 to the most preferred period, 9 to the second most preferred period and so on. The combined teacher preference score is calculated by adding up the scores of the periods for each teacher.

We create 5 randomly generated instances of the data sets where we chose uniformly between possible choices. For example, in the first group of data sets, the number of students can be any number between 100 and 150 with equal probability. Each student has a list of elective courses, sorted in order of preference.

3.2 Computational Results

We ran model 1 with a time limit of 5 minutes and did not impose a time limit for models 2 and 3. Model 1 seems to be difficult to solve to optimality even for the relatively small scheduling problem in our setting. Even after 5 minutes, the integrality gap is on the order of 5-6%.

In Table 2 we give our results on the 5 smaller data sets. The results for the first data set are given in column 2, and for the remaining data sets in subsequent columns. We group the results by scenario type. For each scenario, we first give the combined student preference score, followed by the number of students who got their first choice elective, and then the second choice elective and so on.

In scenario 1, students choose courses by a first come first serve policy (the first arriving student gets all his/her courses in preference order assuming they don't conflict, subsequent students get courses if there is capacity). Recall that in scenario 2, students are assigned to courses to maximize student preference score after teachers and courses have been scheduled, where as in scenario 3, course scheduling is done along with student assignment while maximizing student preference scores. We see that for all random instances, the scenario 1 student preference score is less than the score for scenario 2 and the score for scenario 2 is less that the score for scenario 3.

The first observation suggests that allocating students via an optimization formulation can get a better combined preference score than a first come first serve policy. As an example, consider data set 3, where the overall score is higher in scenario 2 than in scenario 1 and many more students get their first choice course (the first come first serve policy will tend to reward early students with all their course choices barring conflicts while penalizing later students). The second observation suggests that taking student preferences into account while scheduling courses yields better student preference scores than if courses are scheduled first. Note that for data set 4, the difference

Scenario 1 Score	499	442	527	554	483
Choice 1:	84	99	84	100	92
Choice 2:	40	19	49	41	25
Choice 3:	20	7	20	14	17
Choice 4:	3	3	4	3	6
Scenario 2 Score	564	475	600	568	506
Choice 1:	117	96	134	106	98
Choice 2:	27	28	19	40	29
Choice 3:	7	3	3	12	12
Choice 4:	1	1	1	0	3
Scenario 3 Score	574	612	601	686	598
Choice 1:	122	102	135	106	98
Choice 2:	24	50	18	65	57
Choice 3:	5	23	3	31	17
Choice 4:	4	8	1	5	1

Table 2: Results of experiment 1

in scores between scenario 2 and 3 is very significant. In this data set, even though the same number of students get their most preferred course in both scenarios, many more students get their second most preferred course in scenario 3 (65 vs 40).

In the next table, we give our results for the larger data set. We note that the differences between scenario 2 and 3 are not as stark as in our previous group of data sets. However, scenario 2 scores are consistently much higher than scenario 1 scores.

4 Conclusions

Even with our simplified model, it is clear that taking student preferences into account while scheduling courses can yield a much better result for the students, depending on the data. In other words, if scheduling was "driven by data collected from student plans of study," as suggested in the Hanover Research quote from page 2, the outcome for students would be much better than current practice. Furthermore, even after course schedules have been created, one can get much better assignments of courses to students if one

Scenario 1 Score	1622	1670	1640	1785	1781
Choice 1:	239	288	259	289	264
Choice 2:	156	131	147	153	164
Choice 3:	86	54	67	68	98
Choice 4:	26	17	29	34	37
Scenario 2 Score	1894	1903	1996	2096	2148
Choice 1:	384	428	490	468	461
Choice 2:	107	63	12	72	100
Choice 3:	18	1	0	4	2
Choice 4:	1	0	0	0	0
Scenario 3 Score	1917	1906	1996	2098	2152
Choice 1:	389	430	490	468	464
Choice 2:	113	62	12	74	98
Choice 3:	11	0	0	2	1
Choice 4:	0	0	0	0	0

Table 3: Results of experiment 2

solves a course assignment problem for students instead of assigning them in a first come first serve manner.

Acknowledgements

We would like to thank Professor Amir Ali Ahmadi of Princeton who suggested the specific question studied in this document: how much impact does taking student preferences into account have on the overall course scheduling process.

References

[1] H. Algethami, W. Laesanklang, A Mathematical Model for Course Timetabling Problem with Faculty-Course Assignment Constraints, IEEE Access, DOI: 10.1109/ACCESS.2021.3103495, 2016.

[2] Best Practices in Course Scheduling, Hanover Research, 2018. https://www.cmich.edu/colleges/se/Documents/Hanover Research - Best Practices in Course Scheduling.pdf

[3] N. Boland, B. D. Hughes, L. T. G. Merlot, and P.J. Stuckey, New integer linear programming approaches for course timetabling, *Computers and Operations Research* **35** (7), 2209–2233 (2008).

[4] E. K. Burke, D. de Wera, and J. F. Kingston, Applications of timetabling, in *Handbook of graph theory*, pp. 445–474, Chapman and Hall, London (2004).

[5] M. W. Carter, C. A. Tovey, When is the classroom assignment problem hard?, *Operations Research* **40**(S1), 28–29 (1992).

[6] Class scheduling (aka Timetabling) practices and technology, American Association of Collegiate Registrars and Admissions Officers, 2016. https://www.aacrao.org/docs/default-source/research-docs/aacrao-sept-2016-class-scheduling-practices-60-second-survey-report.pdf

[7] CourseAtlas FAQs, Rutgers University. https://scheduling.rutgers.edu/courseatlas/faq.

[8] S. Daskalaki, T. Birbas, E. Housos, An integer programming formulation for a case study in university timetabling, *European Journal of Operational Research* **153** 117–135 (2004).

[9] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman (1979).

[10] A. Hertz, V. Robert, Constructing a course schedule by solving a series of assignment type problems, *European Journal of Operational Research* **108**, 585–603 (1998).

[11] IBM ILOG CPLEX optimizer, https://www.ibm.com/analytics/cplex-optimizer.

[12] N. Pillay, A survey of school timetabling research, *Annals of Operations Research* **218**, 261–293 (2014).

[13] A L. Lovelace, On the Complexity of Scheduling University Courses, Master's thesis, California Polytechnic State University, San Luis Obispo, 2010.