Final Project Report:

Academic Scheduling Optimization for USC Marshall School of Business

DSO 570: The Analytics Edge: Data, Models, and Effective Decisions

Professor Peng Shi

May 12th, 2020

By: Anni Cai, Yura Shakhnazarian, Shringar Sharan, Dakota Wu, Vivian Yang

Table of Contents

1. Executive Summary

In spring of 2020, our team of consultants were hired by the Assistant Dean of Institutional Research and Academic Administration at the USC Marshall School of Business to investigate the possibility of applying optimization to improve the current course scheduling system. We were tasked to create an algorithm that can optimize the current classroom allocation process for 7 departments and 22 academic programs in the USC Marshall with quantifiable gains.

Our team has chosen to focus the scope of our investigation specifically on the initial allocation of graduate-level classrooms. We analyzed the current scheduling process and identified some weaknesses that we planned to tackle. The greatest inefficiencies we identified are the use of historical schedules in initial allocation and the failure to take into account faculty preferences. With focus on satisfying professor preferences, we created an algorithm using Gurobi that can be easily applied by the course scheduling team at USC Marshall.

As a result, we were able to completely eliminate the use of historical schedules and factor in faculty requests during the initial allocation of time slots. Based on simulations of faculty preferences, our algorithm satisfied 100% of faculties' top choices, resulting in a 38.89% improvement in faculty satisfaction scores. Although our algorithm is not ready to be implemented immediately due to the limited scope of our investigation, we recommend the school administrators to invest in more resources to build a fully automated scheduling system based on our prototype algorithm. We believe the end result can not only increase faculty satisfaction but also decrease the amount of ad hoc requests on schedule changes and shorten the entire scheduling process.

2. Opportunity for Improvement

We have identified four major areas of weaknesses and inefficiencies through evaluating the current scheduling process as described by the school administrators and analyzing the datasets provided. The problems we identified include:

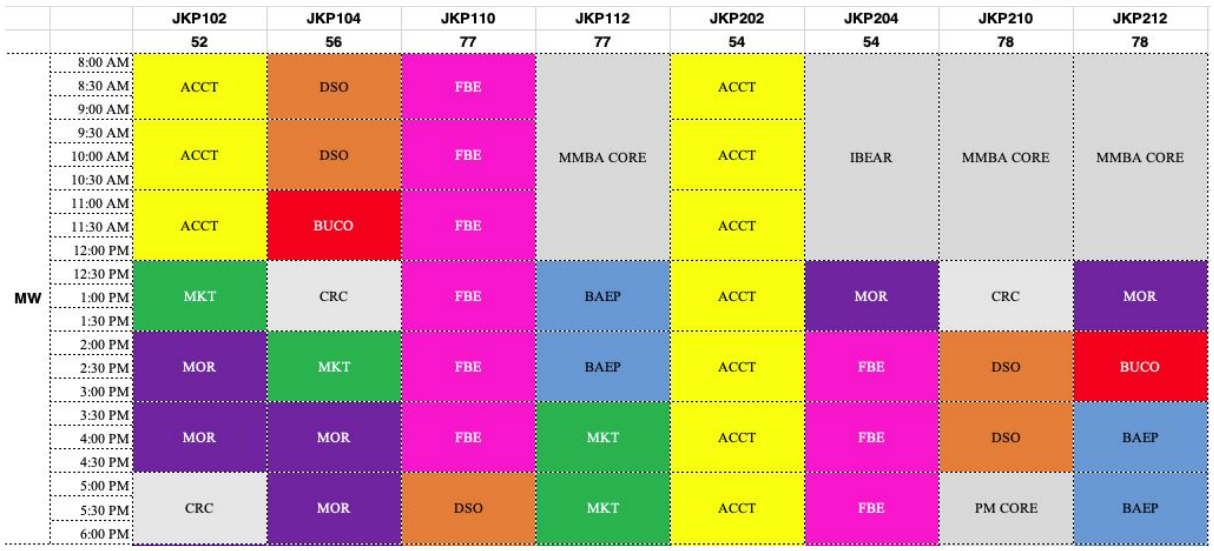
i. the inefficient use of historical schedules for initial allocation of time slots;
ii. the lack of consideration for faculty preferences;
iii. the lack of consideration for student preferences;
iv. the inefficient use of allocated classroom space.

For the purpose of this project, our team has decided to focus on proposing a new solution that addresses the first two problems identified with quantifiable improvements. Each problem is described in more detail and illustrated with data analyses below.

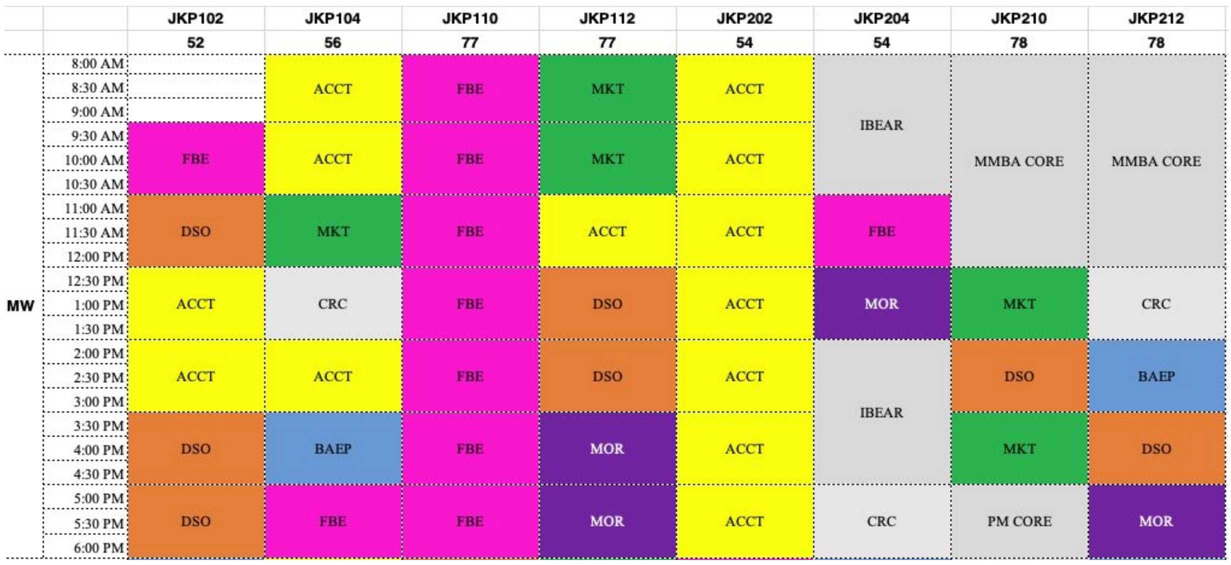
2.1 Use of Historical Schedules

Currently, the initial allocation of time slots from the scheduling team is determined solely based on the historical allocation for courses from that department. This is not an efficient step in the scheduling process as it assumes the courses scheduled each semester are generally the same, when in reality many factors such as the electives offered and faculty staffed can change each semester. By allocating initial time slots that are identical to the schedule of the previous semester, no information relevant to the current semester is taken into account, which decreases the efficiency of the whole scheduling system from the very start.

Below are the respective initial schedule allocations for 2019 Fall and 2020 Spring, which are consecutive semesters. According to the current system, the initial allocation of the 2020 spring classrooms are exactly the same as the 2019 final allocations. However, as we can see from the initial 2019 spring classroom allocations that over 50% of the classroom allocations have changed from the beginning of phase I to the end of phase II. This shows that the initial allocation is at most 50% efficient and that it is likely that a significant amount of ad hoc adjustments were made to reach the final schedule.



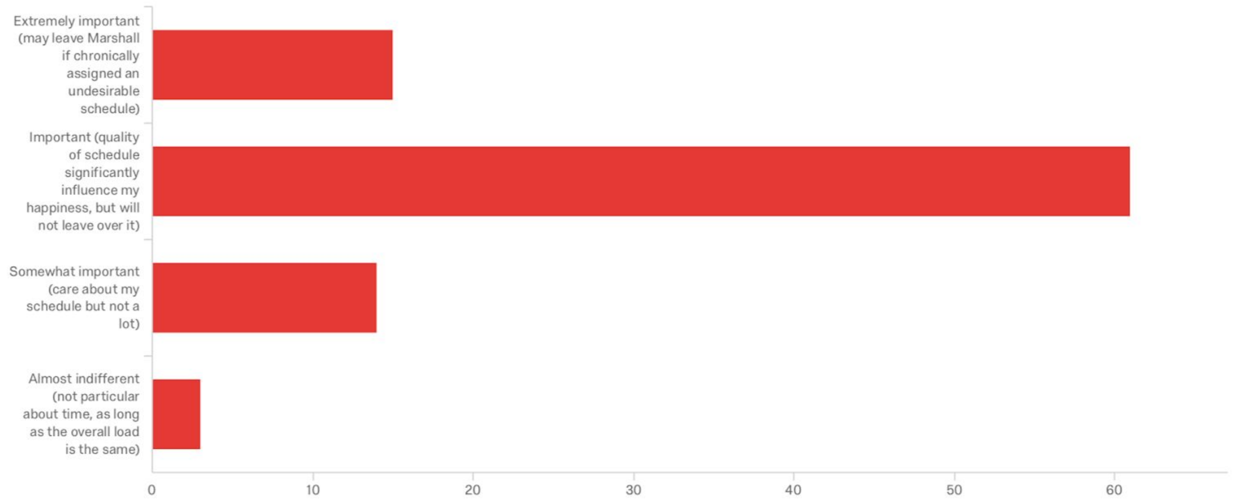
		JKP102	JKP104	JKP110	JKP112	JKP202	JKP204	JKP210	JKP212
		52	56	77	77	54	54	78	78
	8:00 AM								
	8:30 AM	ACCT	DSO	FBE		ACCT			
	9:00 AM								
	9:30 AM								
	10:00 AM	ACCT	DSO	FBE	MMBA CORE	ACCT	IBEAR	MMBA CORE	MMBA CORE
	10:30 AM								
	11:00 AM								
	11:30 AM	ACCT	BUCO	FBE		ACCT			
	12:00 PM								
	12:30 PM								
MW	1:00 PM	MKT	CRC	FBE	BAEP	ACCT	MOR	CRC	MOR
	1:30 PM								
	2:00 PM								
	2:30 PM	MOR	MKT	FBE	BAEP	ACCT	FBE	DSO	BUCO
	3:00 PM								
	3:30 PM								
	4:00 PM	MOR	MOR	FBE	MKT	ACCT	FBE	DSO	BAEP
	4:30 PM								
	5:00 PM								
	5:30 PM	CRC	MOR	DSO	MKT	ACCT	FBE	PM CORE	BAEP
	6:00 PM								



		JKP102	JKP104	JKP110	JKP112	JKP202	JKP204	JKP210	JKP212
		52	56	77	77	54	54	78	78
	8:00 AM 8:30 AM 9:00 AM		ACCT	FBE	MKT	ACCT	IBEAR	MMBA CORE	MMBA CORE
	9:30 AM 10:00 AM 10:30 AM	FBE	ACCT	FBE	MKT	ACCT			
	11:00 AM 11:30 AM 12:00 PM	DSO	MKT	FBE	ACCT	ACCT	FBE		
MW	12:30 PM 1:00 PM 1:30 PM	ACCT	CRC	FBE	DSO	ACCT	MOR	MKT	CRC
	2:00 PM 2:30 PM 3:00 PM	ACCT	ACCT	FBE	DSO	ACCT	IBEAR	DSO	BAEP
	3:30 PM 4:00 PM 4:30 PM	DSO	BAEP	FBE	MOR	ACCT		MKT	DSO
	5:00 PM 5:30 PM 6:00 PM	DSO	FBE	FBE	MOR	ACCT	CRC	PM CORE	MOR

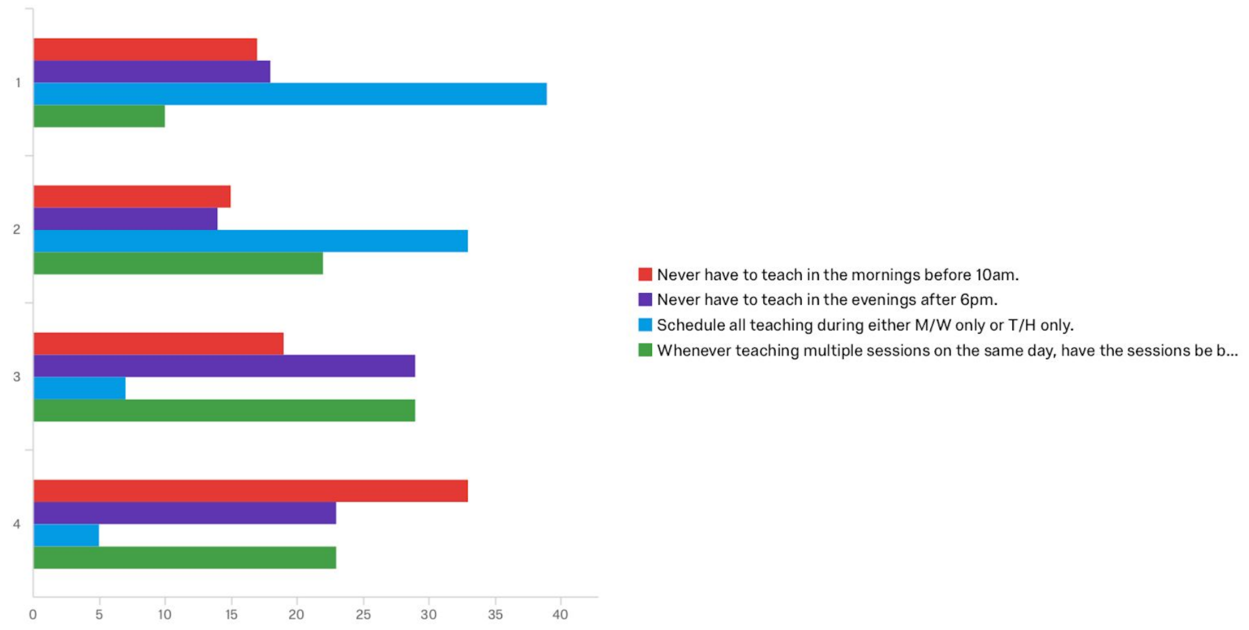
2.2 Faculty Preferences

Faculty preferences are not taken into account during the scheduling process thus creating a need to adjust schedules on an ad-hoc basis after the initial allocation.



#	Field	Choice Count	
1	Extremely important (may leave Marshall if chronically assigned an undesirable schedule)	16.13%	15
2	Important (quality of schedule significantly influence my happiness, but will not leave over it)	65.59%	61
3	Somewhat important (care about my schedule but not a lot)	15.05%	14
4	Almost indifferent (not particular about time, as long as the overall load is the same)	3.23%	3
			93

By analysing the faculty preferences survey report above, we found that almost 82% of faculty considers obtaining a preferable teaching time as very important and important.



#	Field	1		2		3		4		Total
1	Never have to teach in the mornings before 10am.	20.24%	17	17.86%	15	22.62%	19	39.29%	33	84
2	Never have to teach in the evenings after 6pm.	21.43%	18	16.67%	14	34.52%	29	27.38%	23	84
3	Schedule all teaching during either M/W only or T/H only.	46.43%	39	39.29%	33	8.33%	7	5.95%	5	84
4	Whenever teaching multiple sessions on the same day, have the sessions be back to back.	11.90%	10	26.19%	22	34.52%	29	27.38%	23	84

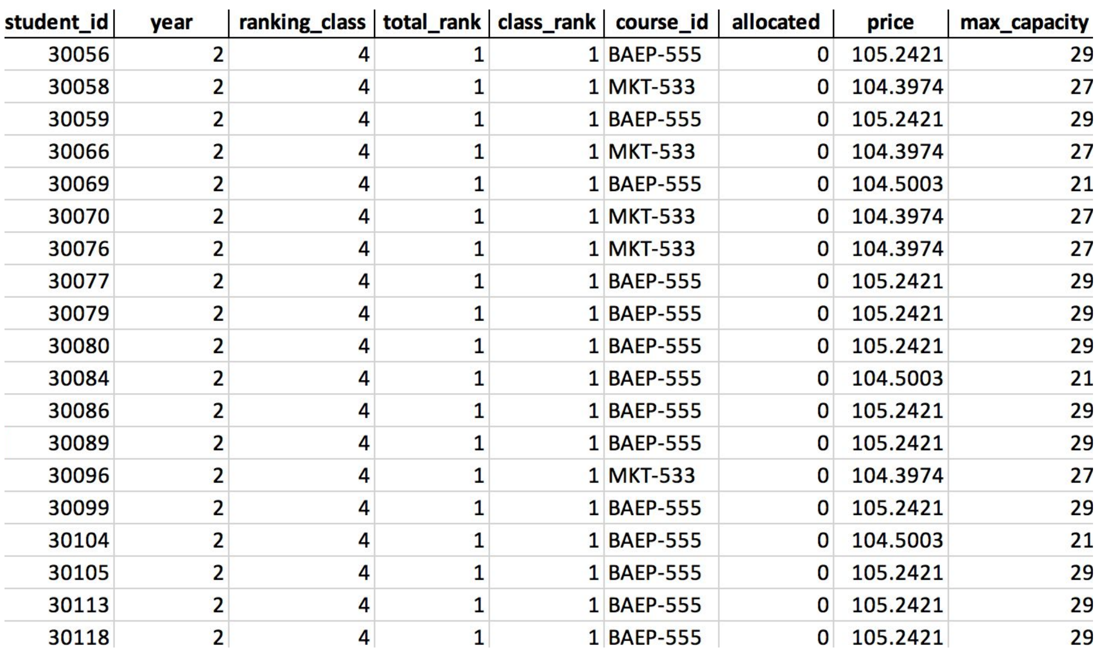
Moreover, when asked to rate the above mentioned pattern in teaching schedule in order of importance (1 being the most important and 4 being the least important), the most popular choice for most important was 'Schedule all teaching during either M/W only or T/H only' with 46.43% (39) votes out of 84 total faculty surveyed.

2.3 Student Preferences

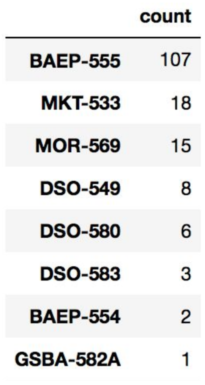
In Spring 2018, USC Marshall experimented with using a new course selection system called CourseMatch, in which students indicated their preference over all courses, and a special algorithm determined the assignment. For various reasons, USC did not continue using the system for subsequent semesters, but the data collected for this semester gave insights on student preferences for full-time MBA, part-time MBA and IBEAR students.

We analysed the student preferences of the MBA students and found that many students are not able to get courses of their choices because of the restriction on the size of the classrooms and hence the allowed registrations.

Here we have shown a sample of 20 students out of 165 students who were not allocated their highest ranked course of choice. There are 13 such courses with the following distributions.



student_id	year	ranking_class	total_rank	class_rank	course_id	allocated	price	max_capacity
30056	2	4	1	1	BAEP-555	0	105.2421	29
30058	2	4	1	1	MKT-533	0	104.3974	27
30059	2	4	1	1	BAEP-555	0	105.2421	29
30066	2	4	1	1	MKT-533	0	104.3974	27
30069	2	4	1	1	BAEP-555	0	104.5003	21
30070	2	4	1	1	MKT-533	0	104.3974	27
30076	2	4	1	1	MKT-533	0	104.3974	27
30077	2	4	1	1	BAEP-555	0	105.2421	29
30079	2	4	1	1	BAEP-555	0	105.2421	29
30080	2	4	1	1	BAEP-555	0	105.2421	29
30084	2	4	1	1	BAEP-555	0	104.5003	21
30086	2	4	1	1	BAEP-555	0	105.2421	29
30089	2	4	1	1	BAEP-555	0	105.2421	29
30096	2	4	1	1	MKT-533	0	104.3974	27
30099	2	4	1	1	BAEP-555	0	105.2421	29
30104	2	4	1	1	BAEP-555	0	104.5003	21
30105	2	4	1	1	BAEP-555	0	105.2421	29
30113	2	4	1	1	BAEP-555	0	105.2421	29
30118	2	4	1	1	BAEP-555	0	105.2421	29

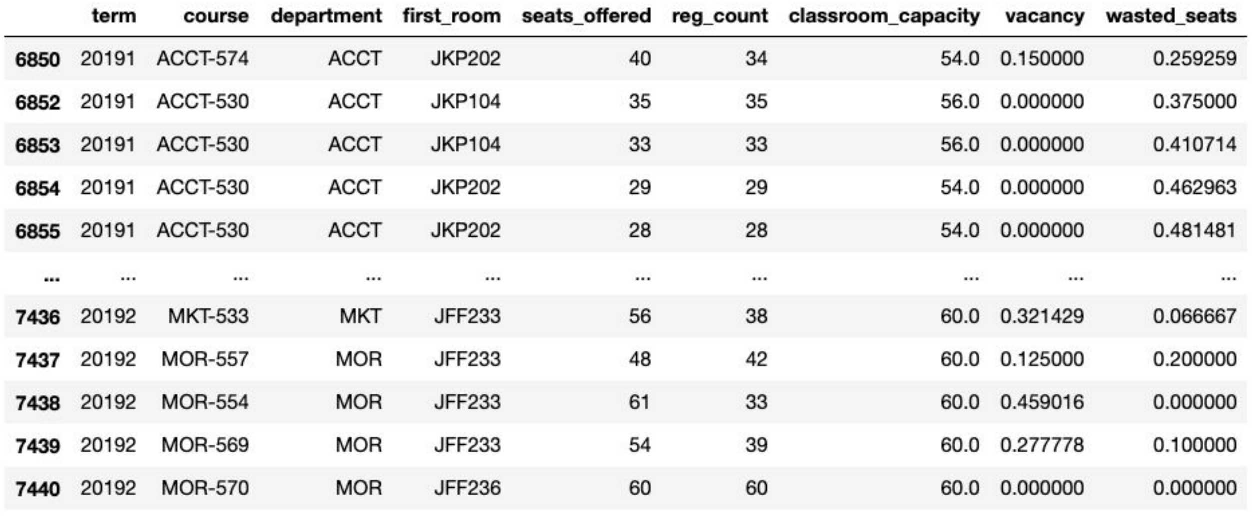


	count
BAEP-555	107
MKT-533	18
MOR-569	15
DSO-549	8
DSO-580	6
DSO-583	3
BAEP-554	2
GSBA-582A	1

We can see from the table that BAEP-555 is the most popular and also the highest course among MBA students which could not be allocated to 107 students.

2.4 Occupancy of Classrooms

Lastly, we also evaluated the efficiency of current classroom schedules by computing two additional metrics: classroom vacancy rates and wasted seats. We used the file with past schedules to evaluate the current occupancy of classrooms, filtering data only for graduate classes in the year of 2019. The classroom vacancy rate is calculated by dividing the number of registration ("reg_count") by the total classroom capacity. The rate of wasted seats is calculated by dividing the number of seats offered by the professor ("seats_offered") by the total classroom capacity. The metrics aim to evaluate if classroom space is wasted when small classes are assigned large rooms, or when there's a significant amount of unused seats due to low registration rates for certain classes. The added features for each course are displayed in the following DataFrame.



	term	course	department	first_room	seats_offered	reg_count	classroom_capacity	vacancy	wasted_seats
6850	20191	ACCT-574	ACCT	JKP202	40	34	54.0	0.150000	0.259259
6852	20191	ACCT-530	ACCT	JKP104	35	35	56.0	0.000000	0.375000
6853	20191	ACCT-530	ACCT	JKP104	33	33	56.0	0.000000	0.410714
6854	20191	ACCT-530	ACCT	JKP202	29	29	54.0	0.000000	0.462963
6855	20191	ACCT-530	ACCT	JKP202	28	28	54.0	0.000000	0.481481
...	...	...	...	...	...	...	...	...	...
7436	20192	MKT-533	MKT	JFF233	56	38	60.0	0.321429	0.066667
7437	20192	MOR-557	MOR	JFF233	48	42	60.0	0.125000	0.200000
7438	20192	MOR-554	MOR	JFF233	61	33	60.0	0.459016	0.000000
7439	20192	MOR-569	MOR	JFF233	54	39	60.0	0.277778	0.100000
7440	20192	MOR-570	MOR	JFF236	60	60	60.0	0.000000	0.000000

From the filtered data, the average classroom vacancy rate is 20.59% and the average rate of wasted seats is 18.85%. Additionally, almost 10% of classes have a classroom vacancy rate over 50%. We believe the data illustrates there's room for improvement in better utilizing the available classroom resources.

The inefficiencies identified summarizes most of the difficulties voiced by the current scheduling team. Two issues that we did not directly evaluate is the need for ad-hoc changes made in the current phase II of scheduling and the overall length of the scheduling process. As we do not have enough information to forecast and simulate ad-hoc requests or the overall scheduling timeline, we've chosen not to take into consideration these particular issues directly. However, we believe that once a more systematic algorithm that considers faculty preferences are adopted, there will naturally be a decrease in ad-hoc requests and also a decrease in the overall length of the scheduling process.

3. Optimization Methodology

Once we have identified the four inefficiencies in the current system, we took a deeper dive into the available data and chose to focus on the first two inefficiencies discussed. Our goal was to take into account faculty preferences in the initial allocation of schedules for departments so that ad hoc requests are minimized. This also eliminates the need for departments to fill in the initial allocation of time slots with courses as courses will be automatically scheduled based on the number of units, class duration, class frequency and faculty preferences. The specifics of our algorithm, including the input and out data, the decision variables, objectives and constraints, and the tool execution are described in more detail in the following subsections.

3.1 Input Data

The input data should be an Excel workbook of .xlsx file type, containing three sheets: ‘course’, ‘classroom’, and ‘timeslot’.

The first sheet, ‘course’, should list the information on offered course sections for a given semester. The relevant information include course section number, first instructor, class length, meeting frequency, seats offered and which part of the semester is the course offered. (first half, second half or full semester)

In addition, for convenience, we recommend combining faculty preference surveys with course information. Basically we ask professors to rank their preferences for morning classes (before 10am), evening classes (after 6pm) and other time slots (between 10 am and 6pm), with a ranking of 3 as the most desirable, and 1 as the least desirable. Then in the ‘course’ sheet, we create fields corresponding to different time slots, and record different instructors’ preference ranking so that we could easily refer to them when using the optimization tool.

The second sheet, ‘classroom’, should list the information on all available-for-use classrooms, including classroom name and room capacity. Although the capacity constraint is not defined and integrated in the latest version of our optimization tool, it is highly recommended that the capacity data is provided, so that future implementations can account for the new constraint.

The third and final sheet, ‘timeslot’, should list the information on all available-for-use during all working days (between Monday and Friday) time slots with corresponding time slot indices.

3.2 Optimization Output

The output of our optimization tool is an Excel workbook of .xlsx file type, containing only one sheet: ‘Schedule’. The ‘Schedule’ sheet serves as an easy-to read timetable representation of course section assignments to the respective classroom during the respective time slot on the respective day of week. The timetable has two layers of header, with the upper layer header containing the set of available classrooms, and the lower layer header containing the set of days of week for each of the classrooms in the upper layer. Rows of this timetable are the set of time slots, and a course section should be assigned to a particular cell in the timetable, corresponding to a specific time slot with some specific day and classroom.

3.3 Decision Variables, Objectives, and Constraints

Decision Variables

Our major decision variable is whether to assign a course section to a particular day with a particular classroom and time slot. Other decision variables include whether to schedule a course section on Monday, Tuesday, Wednesday, Thursday or Friday, whether to schedule a course section on Monday and Wednesday if the class is supposed to meet twice a week, and whether to schedule a course section to a specific classroom or a specific time slot. In total, we have 9 decision variables.

Objective
The objective of our scheduling optimization tool is to maximize faculty preference score.

Constraints
Our optimization tool schedules classes based on constraints regarding course length, course meeting frequency, which part of the semester the course is offered and professor availability.

Course length: The number of time slots that a course is assigned to is based on the course length. For example, a 90-minute class can at most be assigned to 1 time slot in a day, and a 180-minute class can at most be assigned to 2 time slots in a day.

Course frequency: The number of days that a course is assigned to is based on the course frequency. For example, a class that meets once a week can at most be assigned to 1 day in a week, and a class that meets twice a week can at most be assigned to 2 days in a week. In addition, since classes that meet twice a week are usually scheduled either on Monday & Wednesday or Tuesday & Thursday, we add another constraint to follow this scheduling convention.

Semester: We add constraints to make sure that courses in the first half or second half of the semester do not conflict with full semester courses. However, overlapping slots are allowed between half-semester courses.

Professor availability: We add a constraint to make sure that for each professor, their different courses are not scheduled in the same time slot.

Some additional constraints are created for more organized scheduling. For example, each combination of classroom, day and time slot can only have 1 course section assigned. Each section should be assigned the same classroom and time slot across the week. We have come up with another two constraints that have not been integrated in the latest version of our optimization tool, but should be considered for future work. They are:

i. the class size must be under the capacity of the classroom that the course is assigned to;

ii. 180-min classes must be assigned to 2 back-to-back time slots on the same day.

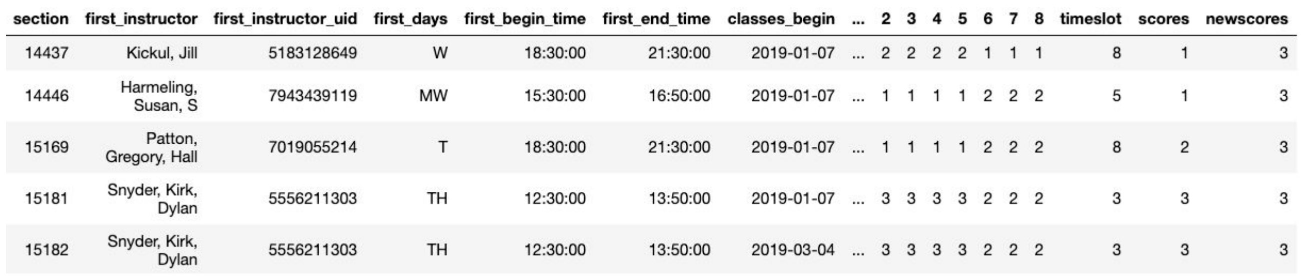
3.4 Tool Execution

The tool is designed for users with no experience in programming, and thus can be executed by any user with ease. As the tool aims to help simplify the whole scheduling process by eliminating initial department allocation by historical schedules, we envision the tool to be used by the scheduling team or whatever personnel that is responsible for the initial allocation of time slots. Below are the detailed steps to run the tool on any of the three major operating systems (Windows/Mac/Linux), assuming that Python 3.6+ with base packages and Gurobi are installed or are in an operational state:

1. Open Anaconda Prompt/PowerShell Prompt (Windows) or Terminal (Mac/Linux)
2. Navigate to your local repository that contains the respective files
3. Type and run the following command *'ipython optimization.py sample_input.xlsx sample_output.xlsx'*. For a general form, replace the previous command with *'ipython optimization.py inputFile outputFile'*, where *inputFile* and *outputFile* indicate input and output filenames, respectively.
4. Thereafter, the command will be executed for quite some time. After the execution is completed and if everything was input correctly and in accordance with the previous steps, the message *'Optimized successfully'* will pop up – indicating that the optimization has been done successfully and the output file has been created.

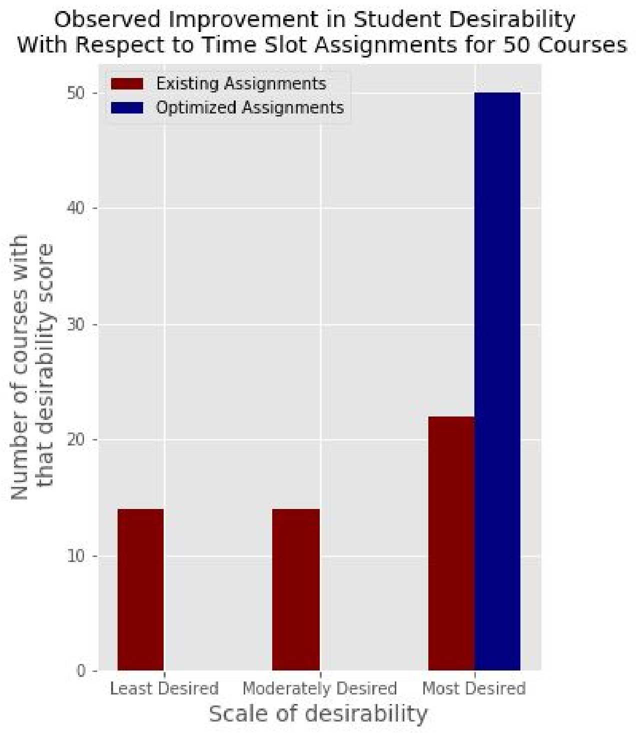
4. Optimization Results

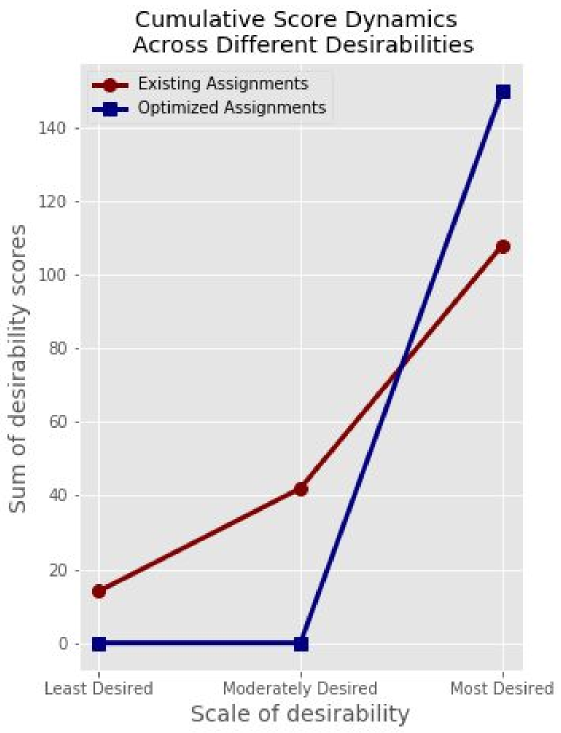
To evaluate our optimization efforts and quantify our gains, we used a simulation of course preference scores to compare the total desirability scores for all courses with respect to their time slot assignments before and after the implementation of our system. We labeled the courses on our sample input file with the different time slots that we coded (from 0 to 8) and matched it with our simulation of preference scores (from 1 to 3, with 3 being most favorable). Two new columns, "score" and "newscores", are added to a data frame representation of the *'course'* sheet for the purpose of evaluating our algorithm. The resulting DataFrame is as follows:



section	first_instructor	first_instructor_uid	first_days	first_begin_time	first_end_time	classes_begin	...	2	3	4	5	6	7	8	timeslot	scores	newscores
14437	Kickul, Jill	5183128649	W	18:30:00	21:30:00	2019-01-07	...	2	2	2	2	1	1	1	8	1	3
14446	Harmeling, Susan, S	7943439119	MW	15:30:00	16:50:00	2019-01-07	...	1	1	1	1	2	2	2	5	1	3
15169	Patton, Gregory, Hall	7019055214	T	18:30:00	21:30:00	2019-01-07	...	1	1	1	1	2	2	2	8	2	3
15181	Snyder, Kirk, Dylan	5556211303	TH	12:30:00	13:50:00	2019-01-07	...	3	3	3	3	2	2	2	3	3	3
15182	Snyder, Kirk, Dylan	5556211303	TH	12:30:00	13:50:00	2019-03-04	...	3	3	3	3	2	2	2	3	3	3

Summing up all the values in column "scores" shows that the existing allocation of time slots produced a total course preference score of 108. The sum of all the values in "newscore," which illustrates the scheduling results of our algorithm, is 150. As a consequence, we found and concluded that all new individual course scores are equal to the maximum preference, 3. As a result, while we're not confident that our algorithm can always match a course to its most attractive time slot, we can quantify that our algorithm does seem to provide an increase in the overall satisfaction of stakeholders that constitute the scores, like students and faculty. This is enforced by the objective of our algorithm, which aims to maximize the total preference scores for each class or section. The following graphs illustrate the change in preference scores visually.





The graphs show that the existing assignment of classes only favored most desired time slots slightly over moderately and least desired time slots, whereas our team's proposed algorithm can satisfy all of the stakeholders' most desired time slots. In fact, evaluating the number of courses in each preference level shows that we were able to increase the overall preference score for 28 out of the 50 courses, and the improved schedule can potentially lead to higher faculty retention rate and satisfaction rate.

In addition to a quantifiable improvement in faculty satisfaction, our algorithm also eliminates the need for initial department allocation completely. As a result, we are able to simplify the scheduling process altogether to increase the overall efficiency of the academic scheduling. Rather than requiring department coordinators to fill in the initial time slots allocated to their department with courses, an initial schedule generated by our algorithm will directly allocate courses for all departments. Department coordinators will only need to review the schedules and any ad hoc requests can be handled with the administrators.

5. Discussion

5.1 Appropriateness of Methodology

A key detail we observed is that faculty members have all the stakes in class scheduling yet they are not directly involved or included in the current class scheduling process until late in phase II. School administrators and department coordinators participate in most of the scheduling work though they are not the ultimate end users or followers of the final schedules. This is why we chose to focus on creating our initial allocation schedule mainly according to preference scores, which, to a large extent, include faculty's reflections too. We assume that most ad hoc requests and late adjustments that occur in the current scheduling process are due to dissatisfaction of the allocated time slots from faculty members. If faculty preferences are taken into account from the beginning of the scheduling process, only small changes will need to be made later on to adjust the initial allocation.

To take into account such preferences, we've designed a survey that allows different stakeholders, including faculty, to rank their preferences by time of day (morning, afternoon and evening). The survey results are then aggregated, and become a key input for our algorithm, which aims to maximize the total sum of preference scores in accordance with the survey information. If we assume that these scores are well representative of faculty preferences, then achieving a higher satisfaction score will indicate a higher overall satisfaction from faculty members. We believe this in turn will create less ad hoc scheduling changes and thus not only shorten but simplify the entire scheduling process. We used simulations of preferences to help us build and test our algorithm, along with other key input data such as course length, beginning and end dates and course frequency.

Understanding the actual scheduling process performed by administrators is complex, we aimed to create an algorithm that is as flexible as possible to avoid potential errors. However, our data and constraints do rely on a few assumptions. First, we assume that prior to using the tool, we already have all the basic information of the courses offered. This information consists of all the data variables we need as an input to our algorithm, which includes the course frequency, length, start date, end date and faculty assignment. Second, we also assumed that all graduate student classes can only be 1.5 or 3 units. All courses that are less than 1.5 units (such as special cases of courses that are 0 or 1 unit) are changed to 1.5 units. Third, much related to the previous assumption discussed, we also assumed that the 1.5 and 3 unit courses correspond to class lengths of 90 and 180 minutes respectively. Fourth, we assumed that there are only 8 classrooms available for all graduate programs. Finally from a logistical perspective, we are also assuming that faculty preference scores are obtainable and easily generalizable in real life.

Once we defined the scope of our data with the assumptions, we created constraints to guide our allocation of courses so that there are no overlaps in resources. The overlaps we tried to avoid include overlaps in classroom usage for any particular time slot and day, overlaps in time slots assigned to the same professor, and overlaps in first-half, second-half and full semester courses. We also set up restrictions so that the class assignments follow the same logic as current schedules. For example, we made sure for classes that meet twice a week that the meeting times are scheduled on either Mondays and Wednesdays or Tuesdays and Thursdays in the same time slots.

Although we have tried our best to optimize the schedule by systemizing the scheduling process, there are two major weaknesses we must address before our system can provide truly usable output schedules. The first is that our current system does not include a strict enforcement that classroom sizes offered must be larger than or equal to the class size. While we understand this is a key constraint, we struggled with the formulation mathematically to enforce the rule on classroom sizes. Additionally, we also need to work on ensuring 180-minute classes are assigned to two back-to-back time slots on the same day. Our current algorithm does assign consecutive time slots for most 180-minute classes but no strict constraint is in place to ensure full accuracy. This is another constraint that we had difficulty implementing that can affect the efficiency and accuracy of our algorithm.

5.2 Final Recommendation

We believe our algorithm is useful as a prototype to automating the entire scheduling process at USC Marshall School of Business. Unlike the existing scheduling system, the algorithm we proposed places a strong emphasis on preference scores and thus considers it as a key input data even in the initial allocation of time slots. We hope that when implemented, the overall satisfaction of involved stakeholders with the initial schedule allocation will increase and therefore decrease the amount of ad hoc adjustment requests and shorten the entire scheduling process.

However, the scope of our investigation is limited and so our algorithm is not ready to be implemented school-wide. We have designed the algorithm specifically for the scheduling of graduate classes and have made assumptions that may not apply to undergraduate classes. With that said, the structure and logic of our class scheduling algorithm can serve as a good basis for a further exploration in completely automating the scheduling process. The benefit of creating our algorithm with Gurobi is that more data variables and constraints can always be added to see if there is an improvement in our objective output.

As a final word, we do not recommend adopting our algorithm immediately as is. Rather, we recommend considering the results we have achieved as evidence that incorporating faculty preferences can greatly increase the efficiency of the current scheduling process. Our algorithm also shows it is possible to create a fully automated system that can immensely reduce the workload of both school administrators and department coordinators. Having illustrated said possibility, we recommend investing more resources in developing an automated scheduling system that can be implemented school-wide.

Technical Appendix

A1. Mathematical Formulation

Data Variables:

- I: set of course sections
- J: set of days $\{M, T, W, H, F\}$
- K: set of classrooms
- L: set of time slots $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$
- A: set of 90min course
- B: set of 180min course
- C: set of courses only meet once per week
- D: set of courses meet twice per week
- R: set of professors
- r_{il}: section i's ranking for time slot l, from 1 to 3, with 3 being the most wanted time slots and 1 being the least wanted time slots
- F: set of first-half semseter courses
- S: set of second-half semseter courses
- U: set of full semester courses

Decision Variables:

- Let x_{ijkl} be a binary variable denoting if a section i is assigned to day j,classroom k, and time slot l
- Let a_i be a binary variable denoting if a section i is scheduled on Monday
- Let b_i be a binary variable denoting if a section i is scheduled on Tuesday
- Let c_i be a binary variable denoting if a section i is scheduled on Wednesday
- Let d_i be a binary variable denoting if a section i is scheduled on Thursday
- Let e_i be a binary variable denoting if a section i is scheduled on Friday
- Let u_{ik} be a binary variable denoting if a section i is assigned to classroom k at all
- Let w_{il} be a binary variable denoting if a section i is assigned to time slot l at all
- Let z_i be a binary variable denoting if a section i is assigned to MW, $i \in D$

Objectives and constraints:

$$
\begin{array}{rcl}
\text{Maximize} & \displaystyle\sum_{i\in I, j\in J, k\in K, l\in L} x_{ijkl} r_{il} & \\
\text{s.t.} & & \\
\text{(1 class per classroom, time slot, day)} & \displaystyle\sum_{i} x_{ijkl} \le 1 & \text{for } j\in J, k\in K, l\in L \\
\text{(If class } i \text{ is scheduled on Mon)} & \displaystyle 0.0001 \sum_{j=M,k\in K,l\in L} x_{ijkl} \le a_i \le \sum_{j=M,k\in K,l\in L} x_{ijkl} & \text{for } i\in I \\
\text{(If class } i \text{ is scheduled on Tue)} & \displaystyle 0.0001 \sum_{j=T,k\in K,l\in L} x_{ijkl} \le b_i \le \sum_{j=T,k\in K,l\in L} x_{ijkl} & \text{for } i\in I \\
\text{(If class } i \text{ is scheduled on Wed)} & \displaystyle 0.0001 \sum_{j=W,k\in K,l\in L} x_{ijkl} \le c_i \le \sum_{j=W,k\in K,l\in L} x_{ijkl} & \text{for } i\in I \\
\text{(If class } i \text{ is scheduled on Thu)} & \displaystyle 0.0001 \sum_{j=H,k\in K,l\in L} x_{ijkl} \le d_i \le \sum_{j=H,k\in K,l\in L} x_{ijkl} & \text{for } i\in I \\
\text{(If class } i \text{ is scheduled on Fri)} & \displaystyle 0.0001 \sum_{j=F,k\in K,l\in L} x_{ijkl} \le e_i \le \sum_{j=F,k\in K,l\in L} x_{ijkl} & \text{for } i\in I \\
\text{(1 if class meets once a week)} & a_i + b_i + c_i + d_i + e_i = 1 & \text{for } i\in C \\
\text{(2 if class meets twice a week)} & a_i + b_i + c_i + d_i + e_i = 2 & \text{for } i\in D \\
\text{(MW or TH if class meets twice a week)} & a_i + c_i = 2z_i & \text{for } i\in D \\
\text{(MW or TH if class meets twice a week)} & b_i + d_i = 2(1 - z_i) & \text{for } i\in D \\
\text{(Overlapping time slots for professors)} & \displaystyle\sum_{i\in p_r, k\in K} x_{ijkl} \le 1 & \text{for } l\in L, j\in J, r\in R \\
\text{(If section } i \text{ is assigned to classroom } k) & \displaystyle 0.0001 \sum_{j\in J,l\in L} x_{ijkl} \le u_{ik} \le \sum_{j\in J,l\in L} x_{ijkl} & \text{for } i\in I, k\in K \\
\text{(1 clasroom max for each section } i) & \displaystyle\sum_{k\in K} u_{ik} \le 1 & \text{for } i\in I \\
\text{(If section } i \text{ is assigned to time slot } l) & \displaystyle 0.0001 \sum_{j\in J,k\in K} x_{ijkl} \le w_{il} \le \sum_{j\in J,k\in K} x_{ijkl} & \text{for } i\in I, l\in L \\
\text{(1 time slot per week for 90min sections)} & \displaystyle\sum_{l\in L} w_{il} = 1 & \text{for } i\in A \\
\text{(2 time slots per week for 180min sections)} & \displaystyle\sum_{l\in L} w_{il} = 2 & \text{for } i\in B \\
\text{(Course overlap in 1st-half of sem)} & x_{mjkl} + x_{njkl} \le 1 & \text{for } m\in F, n\in U, j\in J, k\in K, l\in L \\
\text{(Course overlap in 2nd-half of sem)} & x_{mjkl} + x_{njkl} \le 1 & \text{for } m\in S, n\in U, j\in J, k\in K, l\in L
\end{array}
$$

A2. Discussion of Technical Details

As described in section 5.1 of the report, the largest flaw with our proposed algorithm is the lack of two important constraints that are crucial to the accuracy and usability of our outputs. The two constraints are:

i. classroom sizes offered must be larger than or equal to the class size;
ii. 180-minute classes must be assigned to two back-to-back time slots on the same day.

Aside from the missing constraints, we believe our mathematical formulation and Gurobi code is accurate and thorough enough to ensure the integrity of our algorithm. However, there are some additional details regarding our assumptions that can potentially weaken the efficiency and usability of our system, which we could not account for with the restricted scope we chose to work with in our limited time frame. Specifically, we made a series of assumptions on the possible data types in our input data to limit variability. These assumptions were made so that we can formulate our constraints with more ease and clarity. However, we expect that in reality, there are many more distinct values for the input data.

An example is the length of class. We currently assume that all classes are either 90 minutes or 180 in length, though we know that there are exceptions to this rule. Similarly, we also assume that all classes are either 1.5 or 3 units. If we were to take into account more variability within these input data values, we will essentially have to improve the flexibility of our algorithm. This will include potentially defining new variables that are relevant and adding more decision variables to create additional constraints.