USC Marshall
School of Business

DSO 570 Final Project Report
Marshall Class Scheduling

May 12, 2020

Session: 12:30PM

<u>Team Members</u>

Chutong Yan, Ji Jie, Astha Srivastava, Rory Wang, Jayant Maheshwari

Contents

1. Executive Summary 3
2. Opportunity for Improvement 4
2.1 Time Consuming Scheduling Methodology 4
2.2 Inappropriate Utilization of Classroom Space 4
2.3 Professor's Preference Not Satisfied 6
2.3.1 Avoid Teaching More Than Two Days a Week 6
2.3.2 Desire Back-to-back classes 7
3 Optimization Methodology 9
3.1 Input Data 9
3.2 Output 9
3.3 Decision Variables, Objective and Constraints 10
3.4 Envision of End-User Interaction 10
4 Optimization Results 12
4.1 Potential Gain 12
4.1.1 Efficiency in course scheduling 12
4.1.2 Efficient Space Utilization 12
4.1.3 Number of professors who are allocated at least one back-to-back class 13
4.1.4 Number of professors who has to work more than two days a week 14
5 Discussion 15
5.1 Appropriateness of Methodology 15
5.1.1 Rational Behind Choice of Input Data, Output Data, Objectives, and Constraints 15
5.1.2 Key Assumptions 15
5.1.3 Model Weaknesses 16
5.2 Final Recommendation 17
Appendix 18
A1. Mathematical Formulation 18
A2. Discussion of Technical Details 21

1. Executive Summary

The USC Marshall School of Business has 7 departments, 22 academic programs, and enrollment of around 5000 undergraduate students and 1000 graduate students. With multiple programs, priorities, and preferences, it is difficult for Institutional Research and Academic Administration to create a classroom schedule without scrambling and headaches. Hence, the team was hired by Marshall to investigate the possibility of applying optimization to improve the efficiency and transparency of the current system and to rigorously quantify the potential gains.

In the process of data exploration based on the historical schedule, the team identified three major pain points:

- **Time-consuming scheduling methodology**
- **Inappropriate utilization of classroom space**
- **Professors' preferences not satisfied**. Specifically, the team investigated two general trends regarding professors' preference: 1) Avoid working more than two days a week 2) Desire back-to-back classes

To address the opportunities for improvement, the team proposed an optimization tool prototype that intakes the readily available time slots, classroom and class information, and outputs the schedule for all classes and a summary of the schedule performance. To estimate the potential gain from the optimization, the team applied the tool using sample data based on classes in the 2019 Spring term, with 533 classes, 38 classrooms, and 241 timeslots. As a result, the prototype generated a new free-from-conflict schedule within 40 minutes, which pushes the number of professors who have to work more than two days a week to 0, increases the average utilization rate to 89.58%, and increases the proportion of professors who are allocated at least one back-to-back class by 17.5%.

Based on the results, the team offered Shannon and her team with a series of final recommendations:

- **Pilot the optimization tool with a small section of classes to gain confidence**. For example, utilize the model to schedule only graduate classes or MBA classes.
- **Select the weight of scheduling features by sensitive analysis**. Scheduling features include average classroom utilization rate, number of professors who are allocated at least one back-to-back class, and the number of professors who have to work more than two days a week. By default, the weights for the features above are 1, 1, and 0.2.
- **Adopt the output of the optimization tool as a scheduling start point**, **then address special cases manually**. For example, very few graduate classes with a length of more than 3 hours are not considered by the model and need adjustment by hand.
- **Partition the classes and run parallelly to save runtime**. Currently, allocating regular spring term classes takes the model 25 - 40 minutes. To save the runtime, it is suggested to split the input courses, classrooms and time slots into 2 - 4 non-overlapping parts, run each part parallelly, and finally concatenate the outputs to create the full schedule.

2. Opportunity for Improvement

Based on the exploration of 2015 - 2019 schedules and faculty survey results, the team identified and focused on three major opportunities for improvement, which are explained in detail in 2.1, 2.2, and 2.3.

2.1 Time Consuming Scheduling Methodology

As addressed by Shannon and Hal from the Marshall Office of Finance and Administration, the scheduling work of allocating courses and classrooms at USC Marshall must begin almost one year before the semester begins. For spring 2019, the initial allocation is made at the beginning of spring 2018.

The first phase of the allocation work begins with Marshall Office giving out assigned time slots to each department in Marshall business school, then each department coordinator is responsible to populate the slots with courses as well as assign teaching duties to individual professors. During this phase, although the allocation is based hugely on historical schedules, the task still consists of making changes to new courses or new sections of existing courses and considering professor preferences, and all work mentioned above is done on an ad-hoc basis, which is inefficient, labor costly and time-consuming. This situation has resulted in a lot of last-minute scrambling and headaches for certain departments in the past. In rare cases, the classroom may still be unassigned at the time students choose classes.

Since improving the efficiency and complexity of the scheduling process could not only decrease the workload of the administration office, but also beneficial to the student as there would be fewer occurrences in which students have to make a course selection with unassigned slot, it's important to build a reusable system which would assist in the scheduling process with efficient and correctness, and take the burden, of course, assignment task off the office staff's chest.

Based on the methodology, the team is devoted to building an efficient schedule recommending model, which would generate a recommended schedule on time and assist in increasing the efficiency of the administration office.

2.2 Inappropriate Utilization of Classroom Space

The capacity of Marshall classrooms and seats offered for each class ranges widely from 15 to more than 200. To ensure efficient usage of classroom space, it's critical to match the size of the class and the capacity of the room.

To evaluate the extent of space utilization, the team created three metrics:

1. **Minimum utilization rate in each term,** which quantifies the extreme cases of classroom usage
2. **Average utilization rate in each term**, which quantifies the overall classroom usage. Inevitably, the normal average is somewhat biased as classrooms of different sizes are given the same weight. However, for simplicity, the team decided to adopt ordinary average statistics as the evaluation metric.
3. **Percentage/number of classes utilized less than ¾ of classroom space**. Based on research, it is said that utilization rate between 70% to 80% is commonly considered as good (Classroom and Class Laboratory Utilization Analysis, Dec 2014), thus the team decided to take 75% as the threshold for *efficient* utilization.

Looking across all terms regarding the first metric, the team discovered that sometimes, the utilization rate can be as low as 3.7 % (Table 2.1), which is undoubtedly a huge waste of space. More specifically, in term 20191, ACCT - 574, a class only offering 4 seats, was assigned to room JKP202, a room that can accommodate 54 people. Since there were available smaller rooms, such as ACC 312 with a capacity of 20, this extreme case indicates the inappropriateness in matching class and room size.

Term	Course	Seats Offered	Classroom Capacity	Minimum Utilization Rate
20153	ACCT-559	10	77	12.99%
20161	ACCT-528	2	54	3.70%
20162	GSBA-523T	16	54	29.63%
20163	ACCT-559	15	77	19.48%
20171	ACCT-582	5	78	6.41%
20172	GSBA-523T	25	56	44.64%
20173	ECON-352	73	269	27.14%
20181	ACCT-574	3	54	5.56%
20182	BUCO-552	30	77	38.96%
20183	ACCT-528	8	54	14.81%
20191	ACCT-574	4	54	7.41%

Table 2.1 : Class with minimum utilization rate in each term

The same problem can also be found in the other two metrics. For example, as shown in Figure 2.1, the overall utilization rate fluctuates about 10% from term to term. Because the seats offered for classes and the capacity of rooms are in general similar across Spring/Fall/Summer semesters, the utilization rate gap between 2017 Summer (84.4%) and 2019 Summer (77.2%) indicates that there's room for improvement in terms of classroom usage.

It is also noticeable that the low points of the line graph are all in the summer semesters (2016_2, 2017_2, 2018_2, 2019_2). This pattern is counter-intuitive, as summer semesters usually have fewer courses, which potentially leads to flexible assignment of the classrooms. After investigation, the team discovered that for utility saving, Marshall only opens one building, JFF, for summer sessions. As the choice of capacity in JFF is limited, the space of JFF rooms is in general large, and the class size of summer sessions is commonly small, the average utilization rate was pushed down during summer sessions.

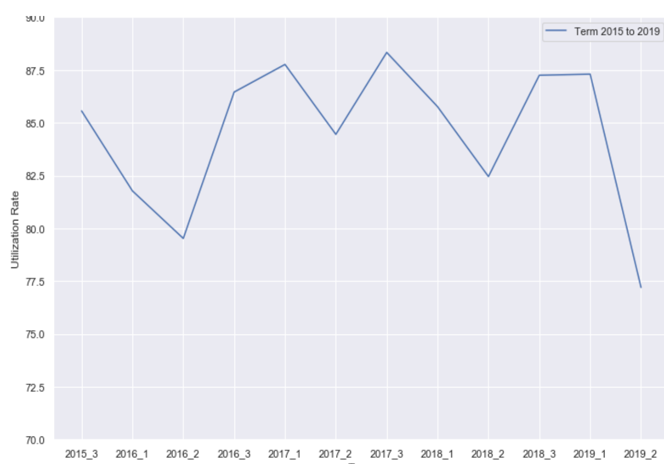

Figure 2.1 : Overall Average Utilization Rate

Finally, looking at the percentage of classes that *inefficiently* utilized the classroom assigned (Table 2.2), the team discovered that in general, more than 20% of classes in each term has at least a quarter of space left wasted, which, in another way, confirms the team's assertion that the classroom utilization is one of the critical opportunities for improvement.

Term	Number of Classes that utilized < 75% of classroom space	% of Classes that utilized < 75% of classroom space
20153	87	23.32%
20161	108	29.43%
20162	14	35.00%
20163	99	18.71%
20171	92	17.07%
20172	7	20.59%
20173	96	17.42%
20181	132	23.00%
20182	9	25.00%
20183	115	21.26%
20191	109	20.45%

Table 2.2 : Number of classes with utilization rate < 75% in each term

2.3 Professor's Preference Not Satisfied

Faculties are the backbone of any educational system. Given that faculties may be a visiting faculty or might be involved in research activities, it is important to consider their preferences while scheduling class, otherwise it might lead to a hampered teaching experience for professors and even restricted learning experience for the students.

Based on the faculty survey data that was provided, the team narrowed down a few cases to understand the general pattern of faculty preference. Although it cannot be concluded that the survey result is representative of all the faculties, it still acts as directional guidance to understand the overall trend. The patterns found are discussed in 2.3.1 and 2.3.2.

2.3.1 Avoid Teaching More Than Two Days a Week

Field	Extremely Important	Important	Somewhat Important	Almost Indifferent
Schedule all teaching during either M/W only or T/H only	46.43%	39.29%	8.33%	5.95%

Table 2.3 : Faculty Survey Results

The above survey (Figure 2.3) is on a Likert scale of 1-4 (1 being most important). It is observed that 94% of respondents believe that *scheduling all teaching during either M/W only or T/H only* is at least somewhat important (rated between 1 and 3) and 46.43% of faculty who took the survey believe it is the most important. Also, on the comment page, several respondents left messages stating that *maximum of 2 days a week is strongly preferred*. In general, it is clear that professors want to avoid teaching more than 2 days a week. This pattern makes sense from a real-life perspective, as it will allow faculty to greatly reduce the commute time, make a more effective routine, and remain productive for the entire week.

To evaluate how past schedules perform in satisfying this preference, the team used *the number of professors who have to teach more than two days a week* as a measurement.

Term	Professor working for more than two days a week (#)	Professor working for more than two days a week (%)
20153	58	29.15%
20161	66	26.83%
20162	5	17.24%
20163	52	23.74%
20171	54	23.28%
20172	6	21.43%
20173	60	27.15%
20181	57	23.46%
20182	7	25.00%
20183	48	21.33%
20191	44	19.73%

Table 2.4 : Number of professors who have to work more than two days a week

As shown in the table above (Table 2.4), the team can conclude that in general 20% - 30% of professors have to work more than 2 days a week in the past semesters. This number is way above the percentage of professors who are almost indifferent to teaching more than 2 days a week (5.95%). In other words, many professors' preference regarding the number of working days per week are not satisfied by past schedules.

2.3.2 Desire Back-to-back classes

Field	Extremely Important	Important	Somewhat Important	Almost Indifferent
Whenever teaching multiple sessions on the same day, have the sessions be back to back	11.90%	26.19%	34.52%	27.38%

Table 2.5 : Faculty Survey Results

The above survey (Table 2.5) shows that although *teaching back-to-back class* is not exactly the first choice for most professors, 73% of professors believe it is at least somewhat important (rated between 1 and 3). The team also conducted focus group interviews with two professors and, based on the results, it is concluded that many professors prefer back-to-back classes. This pattern is understandable in a real-life scenario, as it minimizes the time of professors to stay on campus, helps them teach better if the consecutive classes are different sections of the same course, and makes office hour arrangement more flexible. Although there are cases where some professors prefer long breaks between classes to clear students' doubts or have a rest, as only a general trend is discussed here, these particular cases are not considered for now.

However, as shown by the blue lines in Figure 2.2 (See next page), the percentage of professors who had back-to-back classes (defined as a class with interval less than 30 minutes) are generally around 50% in 2015-2019 schedules. As mentioned previously, around 73% (the red dashed line) of professors believe that having back to back classes is at least in a sense important. The gap between the professor's willingness and the actual assignment indicates a potential opportunity for improvement.

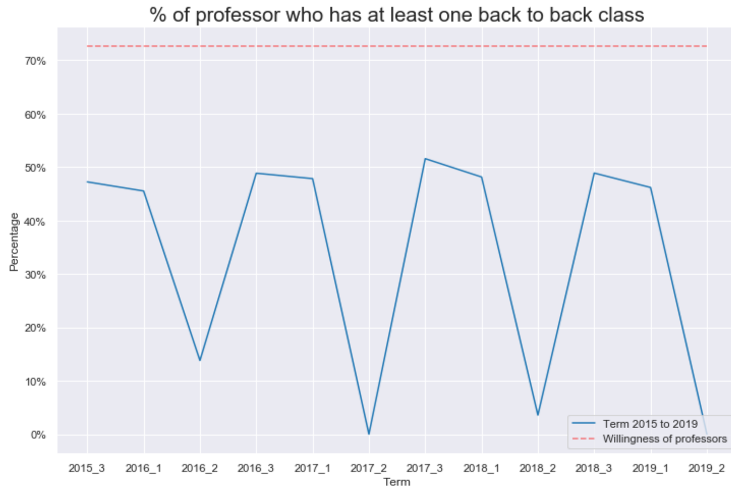

Figure 2.2 : % of professors who have at least one back-to-back class

Additionally, even though many professors do prefer back to back sessions, they still don't want these sessions to take place in different classrooms. It is really difficult for professors to wrap up, take a rest and go to a different classroom, or in some extreme cases to a different building in a short 10-minute break. As expected, several survey respondents left messages stating *having back-to-back classes in the same classroom is strongly preferred*, as it allows professors to take a well-deserved break between their sessions.

However, as depicted in Figure 2.3, the past scheduling of the classroom fails to fully justify this preference. In each term, around 10% - 20% of professors face the problem of rushing to different rooms during the short break between consecutive class time. The huge drop in the graph is due to fewer classes in the summer term.

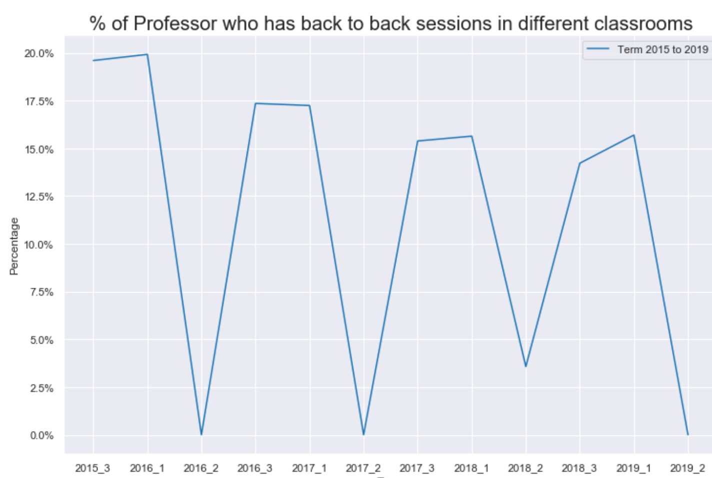

Figure 2.3 : % of Professors who have back-to-back classes in different classrooms

3 Optimization Methodology

3.1 Input Data

The input data fed into the optimization tool is composed of six parts:

1. A set of available **time slots** to allocate *undergraduate* classes.
2. A set of available **time slots** to allocate *graduate* classes.
3. A set of available **classrooms** to allocate *undergraduate* classes and the corresponding capacity.
4. A set of available **classrooms** to allocate *graduate* classes and the corresponding capacity.
5. A set of **classes** (including *undergraduate and graduate*) and the corresponding *level* (undergraduate or graduate), *section index, class index, units*, *number of seats offered*, *session* (full semester, first half or second half) and *instructors* (names of the first and second instructors) information.
6. Three **weights** that end users intend to assign on three performance metrics, including average classroom utilization rate, the number of professors who have to work more than two days a week and the number of professors who are allocated at least one back-to-back class. For easy reference, the three corresponding weights assigned are named as weight1, weight2 and weight3. The weight is between 0 and 1, while 1 means the most important and 0 represents totally indifferent. Only the second metric is set as a punishment to the overall performance, thus the higher the weight assigned on this particular metric, the more the model discourage such cases to occur. By default, the three weights are 1, 1, 0.2 respectively.

The sets of time slots for undergraduate and graduate classes can be overlapping, but they are independent of each other, usually with different start time, end time, and duration. The set of classrooms for undergraduate and graduate classes are completely different, that is, there are no classrooms that both undergraduate and graduate classes can be allocated to.

The within-level time slots, i.e. timeslots at either undergraduate or graduate level, can also conflict with each other. They take the format of *Day of Week + Start Time + End Time + Session,* eg. MW 11:00 - 12:30 PM (Full Semester). Thus, timeslots such as MW 11:00 - 12:30 PM (Full Semester) and M 8:00 - 11:00 AM (First Half) overlap and conflict with each other.

3.2 Output

The output of the optimization tool is composed of 2 parts:

1. A **summary** of the performance of the output schedule. The evaluation metrics include:
   - The average classroom utilization rate (defined as U for easy reference in the report)
   - The number of professors who are allocated at least one back-to-back class (R)
   - The number of professors who have to work more than two days a week (Q)
   - The scheduling score calculated by (weight1*U – weight2*Q + weight3*R).
2. A **detailed schedule** for every undergraduate and graduate class, including the *timeslot* and *classroom* each class is allocated to, together with basic information of the class (*section index*, *class index*, *session, units, seats offered, instructors' name*), and the *classroom utilization rate*.

3.3 Decision Variables, Objective and Constraints

The **decision variable** for the optimization tool is binary, indicating whether a specific class is allocated to a certain classroom at a certain time slot.

The **objective** is the scheduling score mentioned in 3.2, which is calculated by the average classroom utilization rate times weight1, minus the number of professors who have to work more than two days a week times weight2, plus weight3 times the number of professors who are allocated at least one back-to-back class.

Six **constraints** are included in the model:

1. <u>Hard</u>: Each class eventually only occupies one classroom and one timeslot.
    - More specifically, if it is a double-unit class (i.e. 3 units for graduate and 4 units for undergraduate classes), it takes one of the timeslots categorized as a *full semester for 1.5/2 hours twice a week* or *full semester for 3/4 hours once a week*.
    - If the class is a single-unit class (i.e. 1.5 units for graduate and 2 units for undergraduate classes) taken in a full semester, then it takes one of the timeslots categorized as a *full semester for 1.5/2 hours once a week*.
    - If the class is a single-unit class taken in the first half of the semester, then it takes one of the timeslots categorized as the *first half for 1.5/2 hours twice a week* or the *first half for 3/4 hours once a week*.
    - If the class is a single-unit class taken in the second half of the semester, then it takes one of the timeslots categorized as *second half for 1.5/2 hours twice a week* or *second half for 3/4 hours once a week*.
2. <u>Hard</u>: Each classroom cannot be occupied by more than one class at the same time.
3. <u>Hard</u>: Each class cannot be allocated to any classrooms that fail to accommodate the number of seats offered by the class.
4. <u>Hard</u>: Each professor cannot be allocated to more than one classroom or class at the same time.
5. <u>Soft</u>: Make the number of professors who are allocated at least one back-to-back class as large as possible.
6. <u>Soft</u>: Make the number of professors who have to work more than 2 days a week as small as possible.

The precise formulation using a mathematical notation can be found in Appendix A1.

3.4 Envision of End-User Interaction

Shannan Faris, Assistant Dean of Institutional Research and Academic Administration at the USC Marshall School of Business, and her team are the end-users of the proposed optimization tool.

To use the optimization software, Shannan and her team should take the following two steps:

1. Open a Command window (in Windows) or a Terminal (in mac). Navigate in the command line to the directory containing *optimize.py* using *cd directoryName*.
2. Type *python optimize.py input.xslx output.xlsx weight1 weight2 weight3* in the command line. Press *Enter* on the keyboard to run the command. If successfully solved, the output file can be found under the same directory as *optimize.py*.
    - *input* represents the name of the input file and *output* is the name of the output file.

- The last three values are the weight assigned to the three evaluation metrics: average classroom utilization rate, the number of professors who are allocated at least one back-to-back class, and the number of professors who have to work more than two days a week. The weights can be any decimals between 0 and 1, with 1 indicating the corresponding feature is very important and 0 as not important at all. If the end-user didn't input any weights, then the software will by default interpret weight1 = 1, weight2 = 1 and weight3 = 0.2.

The software prints messages to help the end-user either track the optimization status or identify the error. Below are three examples:

1. If any of the input/output files do not exist, the software will notify the end user by printing *File "{FileName}" is not found* in the command line.
2. If any of the key input information is missing, such as input file or output file, then the software will notify the end user by printing *Correct syntax: python optimize.py inputFile outputFile weight1(optional) weight2(optional) weight3(optional)*
3. If the optimization successfully started, the software will update the end-user with current progress and the time consumption of each part. For instance, after setting up the objective of the model, the tool will notify the end user by printing *Set Objective --> XX minutes elapsed*. When the optimization work is finished, the software will inform the end-user by printing *Results in "{outputFile}"*. (As shown in Figure 3.1)

```
Preprocess Data Input
Data Preprocessing Finished --> 0.2 minutes elapsed
You input 533 classes, 107 undergraduate timeslots,134 graudate timeslots, 30
undergraduate classrooms and 8 graduate classrooms
Optimization Starts
Set Variables --> 0.1 minutes elapsed
Set Objective --> 3.2 minutes elapsed
Set Constraint 1 --> 0.1 minutes elapsed
Set Constraint 2 --> 0.2 minutes elapsed
Set Constraint 3 --> 0.1 minutes elapsed
Set Constraint 4 --> 1.4 minutes elapsed
Set Constraint 5 --> 4.6 minutes elapsed
Set Constraint 6 --> 0.2 minutes elapsed
Optimize --> 28.0 minutes elapsed
Write Solution--> 0.2 minutes elapsed
Successfully Finished Optimization in 40.5 minutes
Results in "Sample.xlsx"
```

Figure 3.1 : Sample of end user interaction

4 Optimization Results

To test run the model and estimate the potential gain from the tool, the team ran the optimization model using sample data of classes in the 2019 Spring term, with 533 classes, 38 classrooms, and 241 timeslots. A new suggested schedule was produced, and evaluation analysis had been performed on this particular schedule. Four performance metrics are used to help evaluate the optimization model and allocation strategy as encompassed in the optimization algorithm. The metrics will be visualized using multiple graphs in 4.1, with a blue line indicating the original value in past schedule, an extra yellow line indicating the new value from model output, and an arrow indicating the difference. The final values of the four metrics are as following:

1. Scheduling score (weight1*U - weight2*Q + weight3*R, with all weights set as default) = 118.17
2. The *Average Utilization Rate* (U) = 89.57%
3. The number of professors who are allocated at least one *back-to-back class* (R) = 143 (64.13% of total professors)
4. The number of professors who have to work more than *two days a week* (Q) = 0

The potential gain from applying optimization methodology is explained metric by metric in 4.1

4.1 Potential Gain

4.1.1 Efficiency in course scheduling

An essential benefit of using the model is the huge increase in the efficiency in course scheduling work. Originally the scheduling work of allocating courses and classrooms has to begin almost one year before the semester begins, done by administration staff doing it manually. Now with the existence of the schedule recommending model, the work will be done on time. For as little as 40 minutes the model can produce a free-from-conflict schedule for all 534 classes in 2019 Spring Semester, while pushing the classroom utilization rate to the highest, increasing the number of professors who have *back-to-back* class and decreasing the number of professors having to work for more than two days in a week as much as possible. In a word, the tool would, to a large extent, increase the scheduling efficiency of the administration office.

4.1.2 Efficient Space Utilization

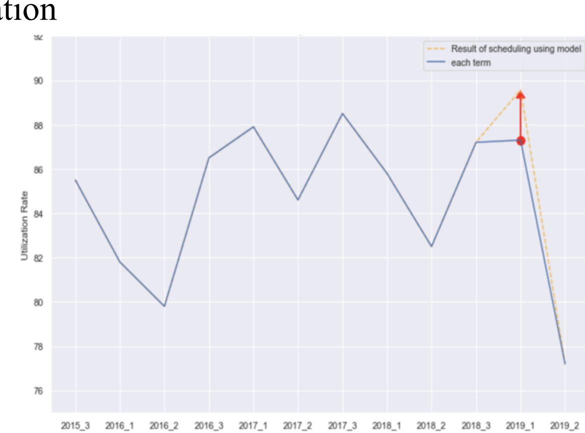

Figure 4.1 : Overall average utilization rate

Term	Course	Seats Offered	Classroom Capacity	Minimum Utilization Rate
20191 (Original Schedule)	ACCT-574	4	54	7.41%
20191 (New Schedule)	BUCO-635	10	78	12.82%

Table 4.1 : Class with the minimum utilization rate

Term	Number of Classes that utilized < 75% of classroom space	% of Classes that utilized < 75% of classroom space
20191 (Original Schedule)	109	20.45%
20191 (New Schedule)	79	14.82%

Table 4.2 : Number of classes with utilization rate < 75% in each term

As for the first objective, which is to maximize the *utilization rate* of each classroom, the model performance is shown in figure 4.1. The blue line indicates the original *average utilization rate* before using the model, while the yellow line indicates the new *average utilization rate* from the model output. There's an obvious increase in the *average utilization rate* using the model, as depicted by the red arrow. The original schedule has an *average utilization rate* of 87.3% in the 2019 Spring Semester, and after using the model to conduct class schedule, the *average utilization rate* increased to 89.57%.

Besides, as shown in table 4.1 and 4.2, the original schedule has the minimum *utilization rate* of 7.41% and is having 20.45% of the classes with *utilization rate* < 75% in the 2019 Spring Semester, and after the model was applied, the minimum *utilization rate* increased to 12.82% and the number of classes with *utilization rate* < 75% has dropped to 14.82%. These metrics indicate that the model is indeed helpful in terms of improving space utilization. The higher the *average utilization rate* is, the more efficient are the spaces in Marshall classrooms being used in general.

4.1.3 Number of professors who are allocated at least one back-to-back class

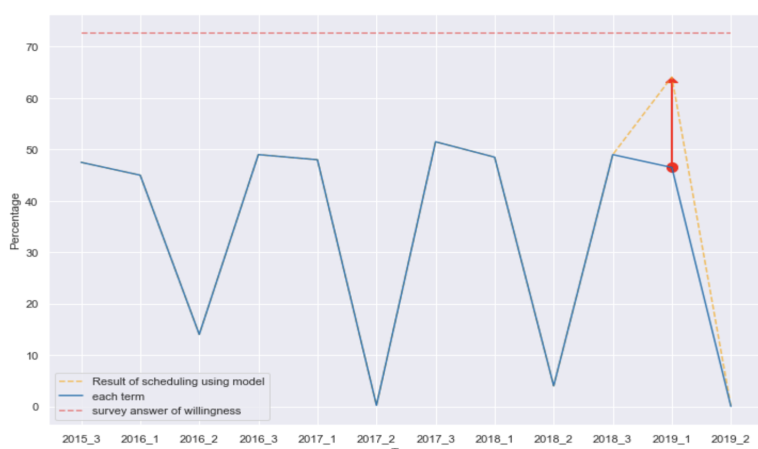

Figure 4.2 : % of Professors who are allocated at least one back-to-back class

As for the second objective, which is to maximize the number of professors who are allocated at least one *back-to-back* class, the model performance is shown in figure 4.2. The blue line indicates the original number of professors who have at least one *back-to-back* courses, while the yellow line shows the new number outputted from the model. It's clear that there's an obvious increase: Only 46.5% of the professors were assigned at least one *back-to-back* course in the original 2019 Spring schedule, while the number increased to 64% after applying the optimization tool. It confirms that the model is effective in terms of successfully increasing the occurrences of *back-to-back* courses in the schedule.

4.1.4 Number of professors who has to work more than two days a week

Number of Days	Count	Percentage
1	453	80.60%
2	109	19.40%
Total	562	100%

Table 4.3: % of Professor who has to work more than *two days a week*

As for the third objective, which is to minimize the number of professors who have to work more than *two days a week*, the model performance is shown in table 4.1. Compared to the original 2019 Spring schedule where 44 professors need to work more than *two days a week*, now in the new schedule, 80.6% of the professors work for only *one day a week*, and the remaining 19.4% professors work for *two days a week*. More importantly, the output schedule makes it possible for all professors to work at most *two days a week*, a feature strongly desired by most faculty. The analysis above indicates that the model did a reasonable job in satisfying this particular need.

5 Discussion

5.1 Appropriateness of Methodology

The main methodology used in the model is to base the model on realistic situations, making the model useful in terms of time and labor saving and easy to be incorporated into the current workflow without huge changes. Besides, the potential improvements from the model should be readily quantifiable.

5.1.1 Rational Behind Choice of Input Data, Output Data, Objectives, and Constraints

The input data contain the basic information of courses, classroom, and related professor. These data are essential in the process of course scheduling to correctly assign each course taught by a certain professor to a certain classroom and time slot. To create as fewer decision variables as possible and save the computation power, the input data mentioned above should be separated by Undergrad/Graduate level courses and provided in two different worksheets in the same MS Excel file, under specific columns and rows as shown in the sample input Excel file attached with this report. All of these are already in the school scheduling office's possession, so they are easy to collect without having to spend additional fetching costs or laboring effort.

The output of the model is shown in clear tables in separate worksheets for Undergrad/Graduate schedules, with all the key information needed for the administration office for scheduling: the *section index*, *course name*, *units, classroom* assigned, *time* assigned, and *instructor* information. Additional information such as *seats offered, classroom capacity,* and *utilization rate* are also offered so that the end-user can easily double-check capacity conflicts and get a rough idea of classroom utilization. The separated worksheets are easy to interpret, ready to use, and convenient to extract key information such as session index. The optimization metrics in a separate worksheet also indicate the key performance of the model, making it easy for users to understand the overall utilization rate as well as other desired features of the scheduling in a quantitative way.

The objectives of the model are to maximize the average utilization rate, minimize the number of professors who have to work more than two days a week and to maximize the number of professors who have at least one back-to-back class at the same time. Additionally, the model allows flexible weighting of the three metrics, as it intakes the corresponding weights as input. That being said, the end-users can easily change the weight assigned on any metric by simply typing the weight in the command line, which is free from changing the model and can be accepted by non-technical users with little knowledge of Python. The reason behind the choice of objectives is because the main goal of the model is to offer a schedule focusing on optimizing the overall efficiency of classroom utilization while taking two specific professor preferences, the desire for working at most two a week and for back-to-back classes into account.

The constraints of the model are listed in Section 3.3 in detail. The hard constraints are basic restrictions needed to create a conflict-free schedule, while soft constraints are developed to help satisfy the professors' preference as much as possible.

5.1.2 Key Assumptions

While conducting data preparation for model building, the sample data was built on below assumptions:

1. Before class scheduling starts, information including the instructor, the seats offered as well as in which session each class will take place is already decided.
2. All the graduate level courses either take 1.5 or 3 hours, while all the undergraduate level courses are either take 2 or 4 hours
3. Each classroom either only accommodates graduate classes or undergraduate classes.
4. Courses in 1.5 units will not be allocated in the evening time
5. Marshall classrooms are closed after 6:30 PM on Friday and no courses can be assigned.
6. All courses under Marshall school of business will only be assigned to classrooms in Marshall business school-owned buildings

All the assumptions listed above were set with regards to simplify the scheduling process, improve student/professor satisfaction, or to comply with the current Marshall convention.

5.1.3 Model Weaknesses

While several constraints are taken into account when building the model, there are still several remaining weaknesses yet to be solved in the future:

1. **Non-personalized preference**: Each professor's preference and availability are not taken into consideration. The model simply follows the generic preference patterns obtained from the anonymous survey while ignoring the availability restriction or personalized preference of each professor. Also, the differences in requirements for full-time and visiting faculties are not considered in this analysis. Being assigned to inappropriate/unavailable time slots for professors may not go well with their schedule or teaching style and can hamper the learning experience.
   In addition, the model only takes into consideration two features regarding professor preferences, while others important trends, such as desire for having back-to-back classes in the same classroom mentioned in 2.3.2, are neglected due to time constraint.
2. **Neglecting course popularity and Bundled Courses**: A few popular classes and mandatory courses must be taken together should not be assigned to the same day and time. However, this has not been taken into consideration by the model. It may result in students dropping out of a popular or must-take course due to scheduling conflicts, which may create a negative sentiment and prevent students from having the best education needed.
3. **Pre-decided classrooms**: All classrooms are pre-decided to either only take undergraduate or graduate-level classes to improve the computational efficiency, but at the cost, the results might not be able to capture the best combination by considering all the combinations.
4. **Input timeslots need manual adjustment**: All time slots are listed in the input file rather than being auto generated by codes. Also, they are developed based on the standard duration of courses. That being said, if the time slots are to be modified (for example, the standard length of Undergraduate/Graduate level courses is changed in the future), then the software requires the user to adjust the time slots manually in the input file, which can be in a sense troublesome.
5. **Exceptions need manual adjustments**: As mentioned in point 4, the model only considers the standard length of classes. However, there are very rare cases where undergraduate classes are 6 hours or 3 hours. Exceptions like this cannot be automatically justified by the model and require the end-user to manually adjust based on the output file.

5.2 Final Recommendation

The Marshall approach of class scheduling consists of utilizing historical course allocations and manually dealing with changes as well as individual department preferences. The proposed model could not only release the administration office from the headache of the complicated course scheduling but also suggest the schedule with the best utilization and satisfy the professors' preference.

While it's suggested for Shannon and her team to utilize the model, there's a series of final recommendations that are to be given:

- **Start using the optimization tool with piloting a small section of classes**. Even though the detailed description on how to use the model and the illustration of the sample output and result metrics generated by Spring 2019 data are listed above, it's still hard for the Marshall administration office to trust the model and utilize the tool to the fullest immediately. It is recommended to start using the model with a smaller size of input data and familiarize the team with the tool and the way to interpret the output data. For instance, set the input data for the model with only MBA courses, times and classrooms when first trying out the model, and check to confirm if the schedule is proper, free-of-conflict, and achieves the preference-related goals. In this case, the model could be gradually used to the fullest.
- **Set the weight based on scheduling needs or school policy**. Such as space utilization, back-to-back classes, and number of days to work can be considered with different importance by easily inputting the corresponding weight in the command line. The logics and restrictions related to these features might change from term to term based on school policy or department needs, so it would be helpful for Shannon to tune the weights by conducting sensitivity analysis. That is, Shannon and her team can gradually change the weights allocated to each feature and see how the performance metrics and class schedules change accordingly, and finally, select the set of weights that gives the most desired schedule performance.
- **Adopt the suggested schedule and manually address additional requirements**. There will always be course changes and additional scheduling requirements in every term, such as professor availability from individual departments, and faculty preferences on time slots or classrooms, which should be taken into account when conducting course scheduling but not captured by the model. Thus, in addition to the output schedule generated by the optimization model, special requirements still have to be handled manually by the administration office.
- **Perform data cleaning to increase model efficiency**. It took the proposed model 25 - 40 minutes to run on all the data of the 2019 Spring term, with 533 classes from 7 departments and 22 academic programs, 38 classrooms, and 241 timeslots. Since the bigger the data size is, the longer it takes to finish running the model and produce the desired output, it's suggested to perform necessary data cleaning to decrease the data size. For instance, splitting the input undergrad/graduate courses, classrooms, and time slots into 2 - 4 non-overlapping parts will massively decrease the size of each dataset, and in that case, it would be possible for the model to run on separated parts parallelly to increase model efficiency. Finally, Shannan and her team can simply concatenate the optimization output of all parts to create the full schedule.

Appendix

A1. Mathematical Formulation

Input data:

- I: the set of classes.
  - a: the set of double-unit class (i.e. 3 units for graduate and 4 units for undergraduate classes)
  - b: the set of single-unit class (i.e. 1.5 units for graduate and 2 units for undergraduate classes) taken in full semester
  - c: the set of single-unit class taken in the first half of semester
  - d: the set of single-unit class taken in the second half of semester
- J: the set of classrooms
- T: the set of timeslots. Below are 7 kinds of timeslots:
  - A: full semester for 1.5/2 hours twice a week
  - B: full semester for 3/4 hours once a week
  - C: full semester for 1.5/2 hours once a week
  - D: first half for 1.5/2 hours twice a week
  - E: second half for 1.5/2 hours twice a week
  - F: first half for 3/4 hours once a week
  - G: second half for 3/4 hours once a week
- S: the set of days of week, including Monday, Tuesday, Wednesday, Thursday, Friday
- V_s: the set of timeslots on day of week s. eg. V_M indicates the timeslots on Monday (includ. M and MW).
- K: the set of professors
- L_k: the set of classes that professor k teaches
- M: the set of back-to-back/consecutive timeslots. Timeslots are defined as back-to-back if the interval between two timeslots is less than 30 minutes
- N: the number of total classes to be allocated
- U_{ij}: the space utilization rate for allocating class i to classroom j, calculated by seats offered over capacity
- z_{ij}: whether the seats offered for class i is less than or equal to the capcaity of classroom j
- O: the set of conficting timeslot pairs
- W_U, W_Q, W_R: the weights assigned on U,Q,R respectively (by default, they are 1,1,0.2 respectively)

Decision variables:
Let X_{ijt} denote whether class i is allocated to classroom j at timeslot t (binary) ($i \in I, j \in J, t \in T$)

Auxillary Variable:

- y_{jt}: the total number of classes allocated to classroom j at time t (integer)
- w_{kt}: the total number of classes allocated to professor k at time t (integer)
- U: the average classroom utilization rate (in %) of the schedule (contineous)
- R: the number of professors who are allocated at least one back-to-back class (integer)
- Q: the number of professors who have to work more than two days a week (integer)
- $H_{kt_1t_2}$: whether professor k teaches back to back classes at consecutive timeslots $(t_1, t_2) \in M$ (binary)
- r_k: whether professor k is allocated at least one back-to-back class (binary)
- Z_{ks}: whether professor k is allocated any class on day of week s (binary)
- q_k: whether professor k has to work more than two days a week (binary)

Unless otherwise noted, every summation of i is over I, of j is over J and of t is over T.

Objective Maximize the schduling score

$$\text{Maximize} \qquad W_U * U - W_Q * Q + W_R * R$$

$$U = \frac{1}{N} \sum_{i,j,t} U_{ij} X_{ijt} \qquad \text{average utilization rate}$$

- U_{ij}: the space utilization rate for allocating class i to classroom j, calculated by seats offered over capacity
- N: the number of total classes to be allocated
- R and Q will be defined in constraint 5 and 6

Constraint 1

English description: Each class eventually only occupies one classroom and one timeslot. More specifically, if it's a double-unit class, then it takes one of the timeslot in A/B. If it's a single-unit class taught throughtout the full semester, then it takes one of the timeslots in C. If it's a single-unit taught in the first half, then it takes one of the timeslots in D/F. If it's a single-unit class taught in the second half, then it takes one of the timeslots in E/G

$$\begin{aligned}
\sum_{t,j} X_{ijt} &= 1 && \text{For every class } i \in I \\
\sum_{t \in A \cup B} \sum_{j} X_{ijt} &= 1 && \text{For every double-unit class } i \in a. \\
\sum_{t \in C} \sum_{j} X_{ijt} &= 1 && \text{For every single-unit class } i \in b \text{ taught throughtout the full semester.} \\
\sum_{t \in D \cup F} \sum_{j} X_{ijt} &= 1 && \text{For every single-unit class } i \in c \text{ taught in the first half.} \\
\sum_{t \in E \cup G} \sum_{j} X_{ijt} &= 1 && \text{For every single-unit class } i \in d \text{ taught in the second half .}
\end{aligned}$$

Constraint 2

English description: Each classroom can not be occupied by more than one class at the same time.

$$\begin{aligned}
\sum_{i} X_{ijt} &\leq 1 && \text{for each room } j, \text{ timeslot } t. \\
y_{jt_1} + y_{jt_2} &\leq 1 && \text{For each classroom } j \text{ and conflicting time slots } (t_1, t_2) \in O. \\
y_{jt} &= \sum_{i} X_{ijt} && \text{For each classroom } j \text{ and time slot } t.
\end{aligned}$$

- y_{jt}: the total number of classes allocated to classroom j at time t (integer)

Constraint 3

English description: Each class can not be allocated to any classrooms that fail to accommodate the number of seats offered by the class.

$$\sum_{t} X_{ijt} \leq z_{ij} \qquad \text{for each class } i, \text{ classroom } j$$

- z_{ij}: whether the seats offered for class i is less than or equal to the capcaity of classroom j

Constraint 4

English description: Each professor can not be allocated to more than one classroom or class at the same time.

Breaking down the above:

- Each professor can't be assigned to a certain timeslot in more than one classroom or class

$$\sum_{i \in L_k} \sum_{j \in J} X_{ijt} \leq 1 \qquad \text{for each professor } k, \text{ timeslot } t$$

- As the timeslots are designed in a way that thay can overlap. Another constraint is added to make sure the classes taught by a certain professor won't have time conflicts. For example, M/W 8-9:30am first half semester and M 8-11am full semester overlaps. In this case, we say that slot t_1 and t_2 conflicts with each other. The set of conflicting timeslots is O.

$$\begin{aligned}
w_{kt_1} + w_{kt_2} &\leq 1 && \text{For each professor } k \text{ and conflicting time slots } (t_1, t_2) \in O. \\
w_{kt} &= \sum_{i \in L_k} \sum_{j} X_{ijt} && \text{For each professor } k \text{ and time slot } t.
\end{aligned}$$

- w_{kt}: the total number of classes allocated to professor k at time t (integer)
- L_k: the set of classes that professor k teaches

Constraint 5

English description: Make the number of professors who have to work more than 2 days a week as small as possible.

Break it down:

Timeslots like MW 8:00 - 9:30 AM and M 9:30 - 11:00 AM are consecutive, let's say the set of consecutive timeslots (t_1, t_2) is M.

1. For each professor, we check whether there are consecutive slots occupied by the same professor's class. The middle part of formula is the total number of classes professor k has to teach at two consecutive timeslots t_1 and t_2 plus 1.

For easy reference, let's call this number a_1. a_1 takes 3 different values: 1,2,3.

- If $a_1 \in [1, 2]$, i.e. the professor doesn't teach back-to-back classes at consecutive timeslots t_1 and t_2, then $H_{kt_1t_2}$ must be 0 to satisfy the LHS.
- If $a_1 = 3$, i.e. the professor do teach consecutive classes at t_1 and t_2, then $H_{kt_1t_2}$ must be 1 to satisfy the RHS.

That is, $H_{kt_1t_2}$ is a binary auxillary variable indicating whether the professor teaches back-to-back classes at consecutive timeslots t_1 and t_2. N is the total number of classes to be allocated by the model, which is a sufficiently large upper bound.

$$3H_{kt_1t_2} \leq \sum_{i \in L_k} \sum_j X_{ijt_1} + \sum_{i \in L_k} \sum_j X_{ijt_2} + 1 \leq H_{kt_1t_2} N + 2 \quad \text{for each prof } k \text{ and consecutive time slots } (t_1, t_2) \in M$$

- M: the set of back-to-back/consecutive timeslots. Timeslots are defined as back-to-back if the interval between two timeslots is less than 30 minutes
- N: the number of total classes to be allocated
- $H_{kt_1t_2}$: whether professor k teaches back to back classes at consecutive timeslots $(t_1, t_2) \in M$ (binary)

2. Then, for each professor k, sum over all consecutive timeslots $(t_1, t_2) \in M$ to check whether he/she is allocated at least one back-to-back class. Let's say it is represented by a binary auxillary variable r_k

$$r_k \leq \sum_{(t_1,t_2) \in M} H_{kt_1t_2} \leq N r_k \quad \text{for each prof } k \in K$$

- r_k: whether professor k is allocated at least one back-to-back class (binary)

3. Sum over all professors to get the total number of professor who are allocated at least one back-to-back class

$$R = \sum_{k \in K} r_k$$

- R: the number of professors who are allocated at least one back-to-back class (integer)

Constraint 6

English description: Make the number of professors who have to work more than 2 days a week as small as possible.

Break it down:

1. For each professor k, check whether any class is allocated for him/her on day of week s. Let's say it is represetned by a binary auxillary variable Z_{ks}.

The middle part of the fomula is the number of courses professor k teaches on day of week s. For easy reference, let's call this number a_2, if $a_2 = 0$, then Z_{ks} must be 0 to satisfy the LHS. If $a_2 > 0$, then Z_{ks} must be 1 to satisfy the RHS. N is the total number of classes to be allocated by the model, which is larger than any possible number of classes a professor would teach on a single day.

$$Z_{ks} \leq \sum_{i \in L_k} \sum_{j} \sum_{t \in V_s} X_{ijt} \leq Z_{ks} N \quad \text{for each prof } k \in K \text{ and each day of week } s \in S$$

- S: the set of days of week, including Monday, Tuesday, Wednesday, Thursday, Friday
- V_s: the set of timeslots on day of week s. eg. V_M indicates the timeslots on Monday (includ. M and MW).
- Z_{ks}: whether professor k is allocated any class on day of week s (binary)
- N: the number of total classes to be allocated

2. Then, for each professor k, check whether he/she has to work more than two days a week. Let's say it is represented by a binary auxillary variable q_k.

The middle part sums Z_{ks} over all days of week $s \in S$ to get how many days professor k has to work per week. For easy reference, let's call this number a_3. If $a_3 \in [0, 1, 2]$, then q_k must be 0 to satisfy the LHS. If $a_3 \geq 3$, then q_k must be 1 to satisfy the RHS.

$$3q_k \leq \sum_{s \in S} Z_{ks} \leq N q_k + 2 \quad \text{for each prof } k \in K$$

- q_k: whether professor k has to work more than two days a week (binary)

3. Finally, sum over all professors to get the number of professors who have to work more than 2 days a week, Q

$$Q = \sum_{k \in K} q_k$$

- Q: the number of professors who have to work more than two days a week (integer)

A2. Discussion of Technical Details

Reasoning Behind Assumptions:

- The team assumes that all the courses that we feed into the model have an assigned instructor, and the seats offered as well as in which session (full semester/first half/second half) each class will take place is already decided. This assumption follows Marshall's current convention and makes sense in a realistic term. Take a professor as an example, if these courses are scheduled in advance, later when the department assigns an instructor to these courses later, there is a good chance that it can violate some hard constraints. For example, when a session without a pre-assigned instructor is allocated to Monday 8 - 9.30 am, and later the department assigns a professor to teach this class. However, there is a chance that this professor has a conflict in schedule and consequently and Marshall needs to re-allocate the schedule.
- The team assumes that only double-unit (4 units for undergrad and 3 units for grad) classes will be 4/3 hours a week for full semester, and all other unit classes are either 2/1.5 hours a week for a full semester or 4/3 hours a week for a half semester. This assumption complies with the pattern with current Marshall classes and there are very few exceptions. Shannon and Hal can schedule these extreme cases manually based on the model output.
- The team assumes all the undergraduate courses and graduate courses take place in completely different classrooms. Even though it is possible and better to schedule the undergraduate and graduate courses

together in all classrooms, it would save us more than 90% of the time if we schedule these courses separately. Also, this assumption is in line with the way Marshall scheduled classes in the past few years, which prevents the model from resulting in drastic changes that could not be easily accepted by the team.

- The team assumes that no 1.5 units of graduate courses will occupy an evening slot, which is, again, complied with the rules and convention of Marshall.
- The team assumes no courses will take place on Friday evening, as in reality, all Marshall classrooms are closed on Friday evenings.
- The team assumes all courses will take place in and only in Marshall classrooms. This assumption is based on Marshall's preference, as Shannon and Hal mentioned that the team would love to use its classrooms as much as possible instead of using USC classrooms to minimize labor and costs.

Discussion of weakness and future work:

- **Non-personalized preference**: So far in our model, the team considered professor's preferences on an overall aspect, instead of taking every professor's own special preference and availability into account. The high overall performance doesn't necessarily mean everyone is satisfied and some professors may be assigned to a timeslot that does not work for them. To solve this problem, what the team needs to do is to collect professors' individual preferences over certain features and their availability over time slots, and then set their availability as a hard constraint and personalized preference score as a soft constraint.
- **Neglecting Course Popularity**: Another problem the team is now facing is that we didn't consider the correlation between different courses. Some courses are popular among students or required by school policy and many students are planning to take these courses in the same semester. However, we didn't add constraints in our model to make sure these courses will not conflict with each other. In future work, we will prepare a dataset that indicates all the courses that should not happen on the same day and the same time, including required courses and popular elective courses and convert these into hard or soft constraints (depending on how strict we want the constraints to be).
- **Pre-decided classrooms**: We scheduled undergraduate courses and graduate courses separately, which increased our efficiency tremendously at the cost of a maybe lower, but acceptable performance. We can increase the performance by scheduling undergraduate courses and graduate courses together, but that will also result in a huge increase in the model running time. Therefore, whether we update our model or not is totally according to the user's willingness. If our users think finding the optimal solution outweighs the running time, or they are equipped with advanced computers that can speed up the optimization time, then the team can update the model to make sure all classrooms are considered for each class.
- **Input timeslots need manual adjustment:** In our current model, the time slots for both undergraduate courses and graduate courses are pre-assigned and fixed. However, if in the future, Marshall decides to change the time slots for courses, the team can update the input file for them.
- **Neglecting other preference features**: Our current model has done a reasonable job in improving utilization rate, increasing the percentage of professors who have back to back classes and decreasing the number of professors who have classes for more than two days. However, there are some other features that we neglect at the moment but can be fully addressed in the future, such as, unpopular class time for students. The team can address this particular example by setting the number of professors who have early morning classes or late evening classes as a soft constraint and minimize this number in the objective function. As a result, the scheduling score will have to balance the trade-off between the number of days for each professor to work and assigning classes to unpopular times.