Classroom Scheduling Optimization

DSO570 Final Project

Prepared by:

(alphabetically ordered by last name)

Bo Kyung (BK) Cho

Qian (Heidi) Huang

Imani Mufti

Nanchun (Aslan) Shi

Chao Wang

**Table of Contents**

[Executive Summary](#_85w3bsdo1gq) **2**

[Opportunity for Improvement](#_4t3na1u95s1c) **3**

[Optimization Methodology](#_x03rih8f8owb) **5**

[Optimization Results](#_c3viajw7kkc1) **10**

[Discussion](#_i31mmlych31s) **13**

[Technical Appendix](#_v6cpg2yn7rsk) **15**

### Executive Summary

In this project, we attempted to optimize two parts of the current scheduling system. We attempted optimization only for classes which are taught on both Mondays/Wednesdays at JKP and JFF in Spring 2019.

Our first objective was to take into instructors’ teaching time preference by scheduling classes at the instructors’ most preferred time.We simulated hypothetical preference scores of teaching time slots and designed a rule-based system to assign teaching time. However, due to limitations of this program, we were not able to achieve a better teaching time preference score after optimization based on the simulated data. A total of 69 instructors were teaching classes on those days and based on our results only 2 instructors were better off than before. The main flaw here is that the program assigns instructors to time-slots when they are unavailable to teach.

Our second objective was to maximize room efficiency by minimizing the number of empty seats in a scheduled class. The initial room efficiency score was 88.7% and the optimized score was the same as the original. The reason is that while our program does reassign rooms, the seats offered for the course and the classroom size remain the same, due to which we did not see different results.

In summary, we were able to create a software that saves time compared with manual assignment of time and room, but due to limited time and resources, our software was not able to deliver results that are 100% feasible.

### Opportunity for Improvement

**Qualitative Situation Analysis**

In order to identify potential opportunities of improvement, we have conducted a situation analysis for the current class scheduling system at USC Marshall. The following is a breakdown of the strengths and weaknesses of the status quo:

1. Strengths

* Phase I of the current time-slot allocation is based on historical schedules. Therefore, it is safe to assume that the resulting changes are not drastic apart from an instructor’s change in schedule or the addition of a new course/section. This limits the workload for the scheduling team relative to creating time slots from scratch every year.

2. Weaknesses

* The current scheduling process takes many months to complete. The process involves communication between Shannon’s team, department coordinators, department chairs and instructors. This makes it time-consuming and tedious to implement.
* Instructor preferences are not collected in a systematic manner. The instructors may reach out to the department coordinator to discuss their preferred time slots. Different instructors may reach out at different times of the scheduling process. If they reach out after Phase I, it would be challenging to incorporate their preferences without disrupting the schedules of other instructors. For instructors who do not reach out at all, they may be assigned an undesirable timeslot which would adversely influence their overall happiness.
* Even after Phase I, a significant fraction of classes often remains unscheduled. This places undue stress on departments as they have to race against time to allocate classes and rooms before student registration.
* It does not take into consideration the rising demand for Marshall courses. Using historical records will not be a sustainable solution if the size of the student cohort continually increases over the years. There must be a mechanism which optimizes for classroom efficiency.

**Quantitative Simulation Analysis**

In order to quantify these weaknesses, we simulated teaching time preference data based on a faculty survey conducted in 2018 (refer to ‘Simulation Method’ later in the report for details and underlying assumptions). Our final program attempts to account for the following:

1. Scheduling Time

It is difficult to provide an exact figure for the current scheduling time as we only know that the current process takes months. Timely allocation is the biggest hurdle facing Shannon’s team which our program successfully addresses for Monday/Wednesday classes.

2. Instructor Happiness

Using our simulated teaching time preference, we can estimate the average happiness scores for instructors based on the initial allocation of results. We find that the average happiness score for the 69 instructors who are scheduled to teach Monday/Wednesday classes is 2.17 and the sum of the happiness scores is 150.05. Documentation of the simulation can be found in “Documentation of tables.ipynb”, and documentation of the plots related to the analysis can be found in “Documentation of plots.ipynb”

With the results of our final program, we can average happiness scores across different assigned classes and sections for the instructor to compute the happiness score for an instructor teaching Monday/Wednesday classes. We will compare this statistic to the before and after average happiness score of the instructor after optimization. The details of this are discussed in the ‘Optimization Results’ section of the report.

3. Classroom Efficiency

We define classroom efficiency as:

The classroom efficiency in Spring 2019 stood at 88.7% for classes scheduled on Monday/Wednesday. Our program is going to attempt to maximize this statistic. Plots related to the analysis can be found in “Documentation of plots.ipynb”

### Optimization Methodology

**Input Data**

1. Courses to be Scheduled

The information of the courses to be scheduled is one of the datasets input by the end user. The data should include the following fields:

* course: Name of the course
* section: Section number of the course
* first\_days: The days assigned for the course in the previous semester
* first\_begin\_time, first\_end\_time: the time assigned for the course in the previous semester
* first\_room: The room assigned for the course in the previous semester
* first\_instructor\_uid: The university ID of the instructor of the course
* seats\_offered: The number of seats available for students to register

Note that to limit the scope of this project, we limited the sample input to Monday/Wednesday courses that were assigned to JFF and JKP buildings in Spring 2019 only, and excluded the situation where a course has a second instructor assigned (which only applied to one course in Spring 2019).

2. Available Rooms

The information of the rooms available for scheduling courses is another dataset input by the end user. The data should include the following fields:

* room: The name of the room formatted as building code + room number. For example, “JFF328”
* capacity: The number of people that the room can accommodate

3. Time Slot Table

The numbering of time slots used to segment time. In this project, we break down hours between 8am to 10pm in a day into 28 half-hour time slots so that courses of all kinds of lengths can be represented by an integer number of time slots.

4. Instructor preference score in half-hour time blocks

To incorporate instructors’ preferences for teaching time, the program takes in their preference scores for time slots broken down into half-hour time slots. This preference data could be collected by distributing a survey to instructors who have teaching responsibilities in the upcoming semester. The survey could be sent to department coordinators by Shannon and Hal. Using a survey to collect teaching time preference will provide a centralized view of all instructors regarding teaching time and save department coordinators the time to interact with each instructor one-on-one and keep track of time requests. In addition, instructors will have the incentive to fill out the survey because it directly impacts their personal schedule.

Since there is no existing teaching time preference data, we simulated the time preference data for demonstration purposes based on the information provided by a faculty survey conducted in 2018. The report of this survey (“faculty\_survey\_report.pdf”) showed that these aspects matter the most to instructors:

* Some instructors cannot teach in early morning because of carpool duty and morning traffic
* Some instructors can only teach at night because they have other day time jobs to cover, but some do not want to teach night classes
* Some instructors prefer to not have classes on Friday, especially undergraduate courses, since students are less likely to focus in class by the end of the week
* Some instructors prefer to schedule all teaching during either Monday/Wednesday, or Tuesday/Thursday
* Some instructors want to have all classes scheduled back to back if they have multiple sessions on the same day

Simulation Method

We incorporated these preferences into the simulation by conducting the following steps:

* Decide the following encoding of preferences: 3 - highly preferred time slots, 2 - preferred time slots, 1 - available time slots, 0 - unavailable time slots
* First mark each instructor’s actual teaching slots in Spring 2019 as 1
* Add the following preferred time patterns to randomly selected instructors:
  + Having one or two days from Monday to Thursday available/preferred to teach
  + Having one or two mornings or afternoons from Monday to Thursday available/preferred to teach
  + Having Monday/Wednesday and/or Tuesday/Thursday highly preferred
  + In addition to the actual teaching time slots in Spring 2019, only having one another morning or afternoon open to teach
  + Only having the actual teaching time slots in Spring 2019 available
  + Having time slots before 10am or after 6pm marked as unavailable
* Make sure very few instructors have preference score higher than 1 on Friday
* Mark all unused time slots as 0

In this way, we simulated different possible preference patterns as one of the input data to the program.

**Program Output**

The output of the program will be the original course data plus two additional columns based on optimization result: assigned time block and assigned room.

**Decision Variables, Objectives, Constraints**

We took a two-step approach to the optimization process. First, we assign teaching times to courses based on instructors’ teaching time preferences. Second, we assign rooms to courses based on most efficient use of room capacity as well as teaching convenience.

1. Time Assignment

To assign time to courses, we wrote a rule-based program instead of using Gurobi. Therefore, there were not explicit decision variables nor objectives encoded into the program. Overall, the objective is to assign courses to the most preferred time slots for as many instructors as possible to maximize instructor happiness, and at the same time try to assign two sections of the same course taught by the same instructor back to back to maximize instructors’ teaching efficiency.

The rules we built into this program are as follows:

* To limit the scope of this project, we used the time blocks used for Monday/Wednesday classes in Spring 2019, which are grouped below as (‘day of the week’, ‘starting time’, ‘ending time’):

1 ('MW', '08:00:00', '09:20:00'), 2 ('MW', '08:00:00', '09:50:00'),

3 ('MW', '09:30:00', '10:50:00'), 4 ('MW', '10:00:00', '11:50:00'),

5 ('MW', '11:00:00', '12:20:00'), 6 ('MW', '12:00:00', '13:50:00'),

7 ('MW', '12:30:00', '13:50:00'), 8 ('MW', '14:00:00', '15:20:00'),

9 ('MW', '14:00:00', '15:50:00'), 10 ('MW', '15:30:00', '16:50:00'),

11 ('MW', '16:00:00', '17:50:00'), 12 ('MW', '17:00:00', '18:20:00'),

13 ('MW', '18:00:00', '19:50:00'), 14 ('MW', '18:30:00', '19:50:00')

This means that when we assign a Monday/Wednesday class that is 1 hour 50 minutes long, only time blocks 2, 4, 6, 9, 11, 13, 14 will be candidate slots, and the other time blocks will be candidate slots for classes that are 1 hour 20 minutes long.

* We can see from the Spring 2019 data that the actual teaching time of all Monday/Wednesday classes are either 1 hour 20 minutes or 1 hour 50 minutes, and that every instructor only taught one course, but he or she could teach multiple sections of the same course.
* The program goes through the courses one by one. If a course only has one section, or the program is assigning time for the first section listed under the course and the instructor, the program assigns the time block of the same time duration and with the highest added preference score (for example, the preference score of a time block of 1 hour and 50 minutes is the sum of the preference score of 4 time slots). If several time blocks are tied as having the highest preference score, the program will randomly assign the section to one of them.
* If the program is assigning time for a section 2 of a course that has another section 1 taught by the same instructor, and section 1 has already been assigned to time block A, the program will assign the time block right after time block A to this section. In this way we can have two sections taught by the same instructor occur back to back. If time block A happens to be the last time block available, then the program randomly assigns a non-overlapping time block to this section.
* To avoid assigning more courses than the number of rooms available to a time block, we limit the number of classes possibly assigned to each time block to the number of rooms available. When assigning a class, if all time blocks of the instructor’s highest preference have been assigned to other courses, we randomly assign a time block that is of secondary preference but still marked as available for the instructor.

In this way, we assign a time block for all Monday/Wednesday courses and maximize the happiness of as many instructors as possible, though as we pointed out before the program did not work as well as we wished. Anyway, with time assigned, we move on to the next step: room assignment.

2. Room Assignment

For room assignment, we made use of Gurobi and wrote the program with the decision variables, auxiliary decision variables, objective, and constraints as follows. Note that we optimized room assignment for each time block specified in Time Assignment separately.

* Decision variable: in the specified time block, whether to assign a particular course to a particular classroom
* Auxiliary decision variable: in the specified time block, the number of extra seats allowed for a course scheduled in a classroom
* Objective: Maximize room use efficiency in the specified time block, which means maximizing the use of available seats in classrooms for each course scheduled in the rooms, and at the same time minimizing the total number of extra seats needed in the classrooms for different classes
* Constraints:
  + In the specified time block, every class gets assigned exactly one room
  + In the specified time block, for every room, there could be maximum one class assigned
  + In the specified time block, the course assigned to a room needs to offer seats no more than the room capacity plus a small number of extra seats allowed
  + In the specified time block, the extra seats allowed in each room should be no bigger than a number specified in the program. After several iterations, we decided to allow 10 maximum extra seats. Too few extra seats would render the problem unsolvable in this case.

In addition to the above constraints, we also placed a few additional rules to ensure feasibility of room assignments.

* Courses taught by the same instructor will be assigned the same room as much as possible
* If two sections are taught by the same instructor but were not scheduled in two consecutive time blocks, and the later section cannot use the same room as the earlier section because that room has been assigned to another course that has time overlap with the later section, then we randomly assign a room still available in this time block to the later section

In this way, we were able to assign a classroom for all Monday/Wednesday courses and maximize the room capacity efficiency. From here we output the optimization result.

**Usage of the Program**

The Phase I of classroom scheduling is reimagined as the process described below to increase efficiency as well as collecting additional data that empowers better optimization results.

* Step 1: Shannon and Hal distribute instructor teaching time preference survey and class sign-up survey to department coordinators to kick start the classroom scheduling process. The two surveys will mainly be multiple choice questions and prefilled with information from last year. For example, when the department coordinator of the DSO department selects its program name at the beginning, all the classes offered by the DSO department in last year will appear with the option to change information such as instructor for the courses. There will also be options to add new courses or delete existing courses. The instructor teaching time preference survey will be distributed by department coordinators to all instructors who have teaching responsibilities in the next semester. Similar to the class survey, when an instructor selects his or her university id, his or her preference from last year will appear as the default. The survey will prompt instructors to select days and times highly preferred, preferred, and available but not preferred to teach, or simply mark days and times that will not be available for teaching.
* Step 2: By the end of Phase I, Shannon and Hal export the class sign-up survey to be input data for classes to be scheduled, and export the teaching time preference survey to be input data for instructor time preference. In addition, Shannon and Hal will prepare the available room data as the input data for available rooms and use the time slot data corresponding to time slot breakdown in teaching time preference survey as the time slot numbering data.
* Step 3: With the data specified above, Shanon and Hal will run the prepared optimization program AssignmentProgram.py in Terminal on a Mac, or Anaconda Prompt of a PC, or running the program in a Jupyter Notebook environment. Detailed steps can be found in the PDF file “Documentation of optimization Tool”.
* Step 4: Check the feasibility of the time and room assignments in the output. From here Shannon and Hal can move on to the second phase of the classroom scheduling process.

### Optimization Results

1. Scheduling Time-Savings

Shannon’s team can use the input files and execute the AssignmentProgram.py program to immediately optimize for the classes scheduled in JFF and JKP on Monday/Wednesday.

2. Instructor Happiness

One of the objectives of our optimization was to maximize the number of classes which are being taught at a time block for which the instructor has specified his preference and to ensure that no class is scheduled where time-slot preference score is 0 which means the instructor is unavailable.

In the simulation, there were 135 classes scheduled at JKP and JFF taught by 69 different instructors. Our optimization program incorporated the instructors preference scores to output an optimized schedule. After the optimization, 76 of the 135 classes were reassigned to a different time slot on Monday/Wednesday.

To illustrate an individual example of how the program works:

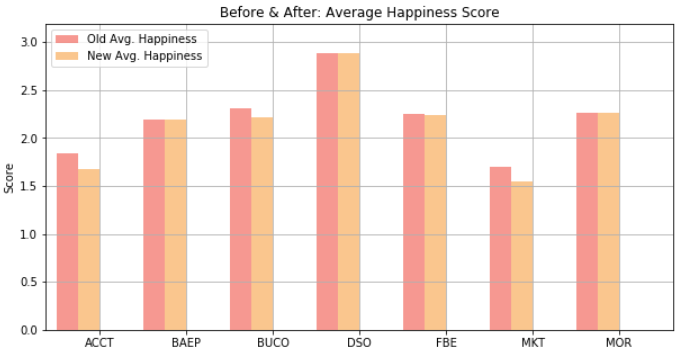
Instructor Gregory Kling’s class timing for ACCT-530 Section 14206 was scheduled for Monday/Wednesday (9:30 - 10:50 AM). The class comprises three half hour time slots, each of which corresponding to the instructor’s preference score of (2,2,2). This allocation had a total preference score of 6. Our optimized version reallocates the class to a time block where the instructor has a higher preference score and reschedules it to Monday/Wednesday (12:30 - 13:50 PM). In this allocation, the three half hour time blocks correspond to the instructor’s preference score of (3,3,2). The optimized allocation has a preference score of 8 which is greater than before. Our program intends to optimize for the highest preference score in this way.

However, the above is an individual desired outcome from our analysis. From 69 instructors teaching on Monday/Wednesday, only 2 instructors are actually better off than before. The following table illustrates the overall before and after average happiness scores for the instructors:

|  |  |
| --- | --- |
| Original Avg. Happiness | 2.17 |
| Optimized Avg. Happiness | 2.12 |
| Difference | -0.05 |

Based on these results, our final optimization does not do as well as our expectations. Our program reduces the average happiness score of the simulated results by 0.05. There are three possible reasons for this. First, our happiness scores are simulated based on some assumptions, so they may not truly reflect instructors’ preferences. Second, our constraints that assign different sections of the same course into a consecutive sequence of time slots might limit the power of happiness optimization. In the future if given more time, we would like to modify the constraint and make it more flexible. Last, since our program only takes into consideration the Monday/Wednesday classes, we are not able to measure global improvement. Therefore, the results above will serve only as an illustration of the functionality of our program.

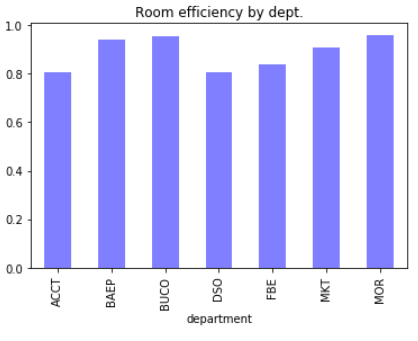
For reference we also check the average happiness score by department:



From our results, we see that there is a slight loss in average happiness for ACCT, BUCO and MKT departments. If we had more time, we would attempt another version of our optimization which not only incorporates a better average score formulation but also accounts for fairness across departments.

3. Classroom Efficiency

Our optimized program has an estimated classroom efficiency statistic of 88.7% which is the same as the original. There was some reassignment of rooms but the room size and class registration size does not change due to which the overall ratio remains the same. The following is a breakdown by department:



### **Discussion**

**Appropriateness of Methodology**

1. Instructor Happiness

We believe that the current methodology would have been the best fit for the situation because it would allow us to capture the preferences of faculty. It was important that we divide the time duration of classes on Monday/Wednesday into 30 minute slots due to classes being of multiple durations such as 1 hour 20 minutes (3 slots) and 1 hour 50 minutes (4 slots) classes. In order to capture the preference for each slot, we believed this would have been a feasible approach. The program is only as good as the input data. Therefore, the results are strongly dependent on the results of the time slot preferences survey which will be shared with faculty.

2. Classroom Efficiency

We wanted to ensure that our optimization program takes into consideration the rising demand for USC Marshall classes. Our objective is to maximize the use of available seats in the classroom for each course in each room. First we want to make sure that every class actually gets assigned an available room. In order to account for the overlap, we needed to ensure that every class has exactly one room assignment. In terms of classroom capacity, we allow for a little wiggle room for the number of extra seats allowed in addition to the room capacity. These extra seats allowed should not be greater than the number defined by us in the program after several trials.

3. Limitations

* The most fundamental limitation of our program is that it did not fully achieve our objectives. Based on our results, average instructor happiness is lower than before and out of the 69 instructors teaching in the semester, and only 2 had a greater average happiness score than before. Furthermore, some instructors have been assigned time where their preference to teach is 0, meaning they are unavailable to teach. Our room efficiency statistic did not change. Although the program reassigns rooms, the room size might remain the same.
* We can think of three possible reasons why our results did not line up with the big picture vision we had set out to achieve:
  + Our preference table is generated with limited insights on faculty teaching time preference and has plenty of randomness introduced to the process, so it may not be truly representative.
  + Because we have not used Gurobi for time assignment, it is possible that there is no time slot available for a class because the program assigns rooms on a first come first serve basis based on at what position in the input file the class is listed. Also, to simplify things, we limited ourselves to using the time blocks defined in Spring 2019 instead of allowing the program to add additional feasible time blocks. If there are no time blocks available for that class then our program would “sacrifice” that class and assign an unfeasible time for it.
* Our results are likely to look different once we incorporate the other days of the week and all the buildings in our analysis. Given the limitations of our logic, it was not feasible to try to optimize for the entire schedule. Please refer to the ‘Discussion of Technical Details’ section for the technical limitations of our program.

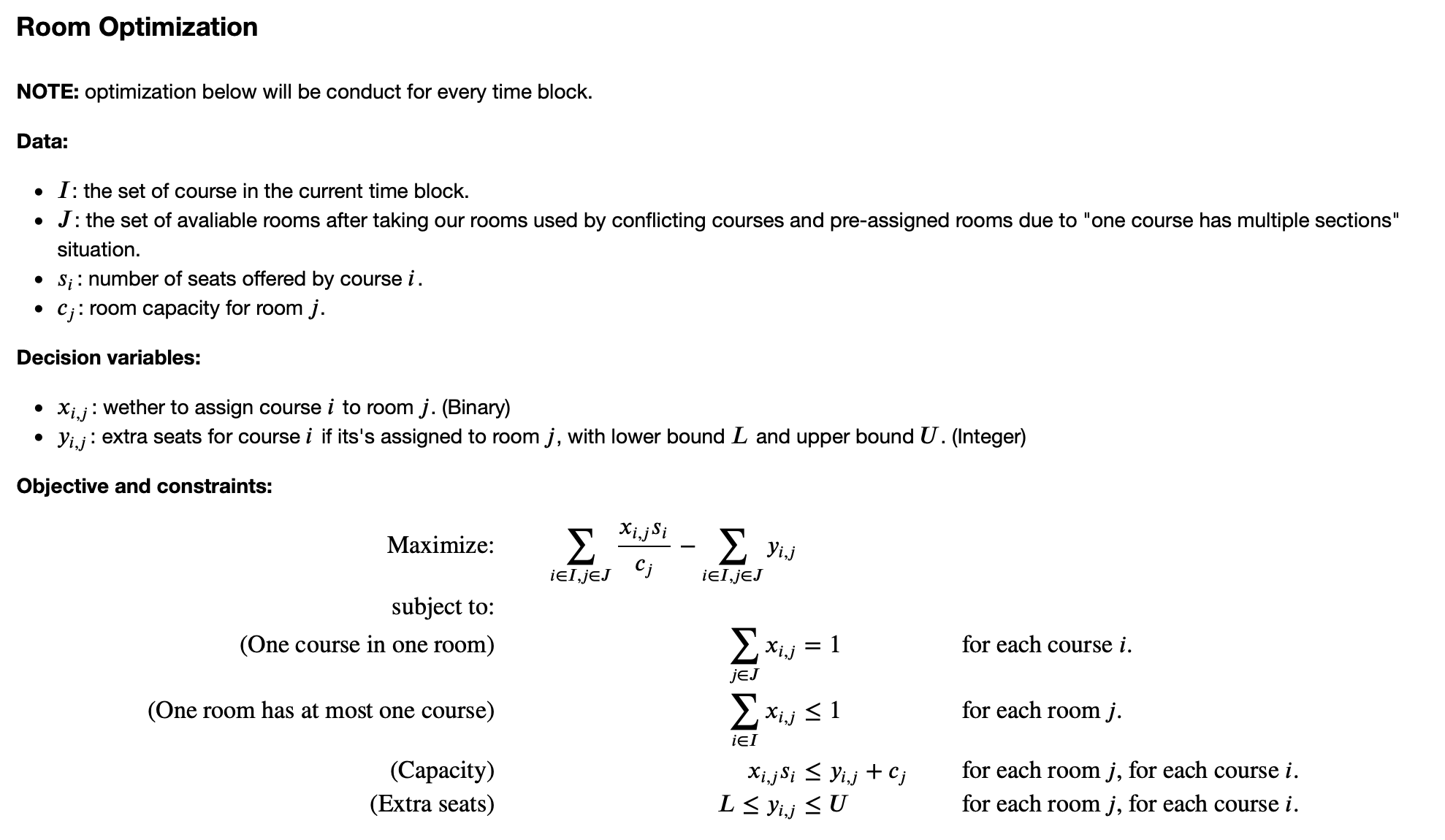
4. Final Recommendation

We recommend to Shannon’s team a redesign of the Phase I process in the current scheduling system is necessary. Phase I should involve more data collection in a more efficient manner to reduce manual labor. Regardless of the results of our optimization, a redesign of Phase I is necessary if any program will be used to optimize the schedule.

We also recommend that our tool be used with caution as it is not a 100% feasible solution.

### **Technical Appendix**

**Mathematical Formulation of Room Optimization**



**Discussion of Technical Details**

There are several limitations of the program that we would like to address if given more time and resources.

* Expand the types of courses that the program can take beyond two-session courses and full-semester courses

Currently the program only takes in courses with two sessions that happened on Monday and Wednesday. However, there might be room for further optimization if we can optimize courses that occur only once or three times in a week. For example, we could potentially move classes previously on Monday and Wednesday to Tuesday and Thursday if the instructors’ preferred teaching time changed, or assign classes previously on Wednesday to Friday if the previous time no longer works for the instructor in the next semester. Overall, expanding the scope of the types of courses will allow us extra room to find the best teaching time to cater to instructors’ needs.

In addition, currently we treated all classes as full semester classes. We would like to take the semester duration into consideration if granted more time.

* Improve the flexibility of the program to optimize time allocation beyond time block constraints

We designed the program to only use time blocks used by Monday/Wednesday classes. However, to optimize schedules for classes shorter or longer than 3 or 4 half-hour time blocks we need to break the barrier of these time blocks and be more flexible of our approach to time.

* Use optimization tools to find the best time allocation with maximized happiness score

Currently, we iterate through the list of courses one by one to assign the best possible time for each course. This is conducted in a greedy fashion to achieve a kind of “local optimum”, and when there are several candidate time blocks for a course available we use random assignment. However, this is done only once in a linear fashion, and we might miss the most optimal schedule for all classes without the help of optimization tools such as Gurobi. If we were given more time, we would like to use Gurobi to tackle the different time constraints, such as having two sections of the same course taught by the same instructor scheduled back to back.