Final Project Report
Mat 226 - Operations Research
Professor Turner

Ngoc Tran, Tu Nguyen, Taner Kiral, Azlan Munir, Omar Chavez, John Lang

April 29th, 2017

1 Problem overview

Every year, the senior class at Wabash College has to take oral comprehensive exams over their major(s). Each student will answer a series of questions from a board of three professors pertaining to their course of study. The current method for assigning oral comprehensive exams is usually very time-consuming, so we are making adjustments to the strategy. Using techniques learned from operations research, we have developed a method to quickly and efficiently schedule oral comprehensive examinations. Furthermore, we are also trying to even out the distribution of number oral comprehensive exams scheduled among all the professors.

In the process of assigning oral comprehensive exams, we need to consider faculty members and student unavailability. Furthermore, there are 10 rules that must be satisfied for a perfect system:

1. The 1st faculty chair needs to be from the department of the student's 1st major.
2. The 2nd faculty chair needs to be from the department of student's 2nd major, otherwise the student's 1st minor.
3. 3rd faculty chair needs to be from the department of the student's 3rd major, otherwise assign an at-large.
4. 3rd faculty chair needs to be from different division than 1st or 2nd oral comprehension chair.
5. If a student has 3 majors, he will need a 4th faculty chair, which would be labeled an at-large, and the student's oral comprehension time has been manually scheduled in the past.
6. Try to avoid scheduling back to back for faculty members if at all possible.
7. Don't use new or part-time faculty for 1st faculty member chair.
8. If the 1st faculty chair is a 2nd-year faculty member, then the 2nd and 3rd faculty shouldn't be a new faculty member.
9. Each faculty member should have no more than 12 comp times, but in past years members have averaged around 8 comprehensive exams per faculty member.
10. Each faculty member should have no more than 4 comps in one day.

2 Formulation of the model

2.1 Variables

First, we will denote,

- TEACHER as the set of all the professors,
- STUDENT as the set of all the students,
- DAY as the set of all the days,
- SESSION as the set of all the sessions.

Then, we created 6 integer variable i, j, k, l, V, and Z. In particular,

- Variable i represents the day of the exam, which can take the values 1, 2 or 3,
- Variable j represents the session number in the day , which can range from 1 to 7,
- Variable k represents the professor and, with our current number of professors, which can take integer value from 0 to 79 taking only integer values,
- The variable l represents the student, which can take integer value from 0 to 189.
- $V[l, div]$ Stands for the number of professors in the comps board of student l in division div,
- $Z[k]$ represents the difference between the average scheduled sessions per professor and the actual number of scheduled sessions of professor k. In particular, by calculation we have that on average each teacher have 7 session. Thus, we will have $Z[k] = |\sum_{l \in STUDENT} C[k, l] - 7|$.

All of the variables above are given for the Wabash Class of 2017's Oral Comprehensive examination schedule. For different colleges or years, the possible range of variables will likely have to be changed depending on the number of students, the number of professors, the number of possible days and the number of possible sessions per day.

Furthermore, we also created binary variables Y, C, P, U, and X. In particular,

- $Y[i, j, l]$ whether student l has an oral comprehensive appointment schedule at day i and session j. For every student there $Y_{i,j,l}$ can be 1 for only one pair of i and j,
- $C[k, l]$ whether teacher k is the chair of student l,
- $P[k, l, e]$ whether teacher k is the e^{th} chair of student l. (e can take the value of 1, 2, 3, or 4),
- $U[l, div]$ whether student l has a faculty at division div as his at-large comps board,
- $X[i, j, k, l]$ whether teacher j has an oral comps appointment with student l at day i and session j,

Lastly, based on the data given by the excel sheet, we will have constants UCAP, SNR, BUSY, BUSZ, TRIPLE. In particular,

- UCAP$[k]$ the preferred maximum of oral comps for teacher k. The default value of UCAP$[k]$ for each k in TEACHER is 12,
- SNR$[k, yr]$ whether teacher k is at his/her year yr at Wabash (0: more than 3 years, 1: first year, 2: second year),
- BUSY$[i, j, k]$ whether teacher k is busy at day i, session j (0: not busy, 1: busy),

- BUSZ$[i,j,l]$ whether student l is busy at day i, session j (0: not busy, 1: busy),
- TRIPLE$[l]$ whether student l is a triple major (0: not a triple major, 1: a triple major). This constant is used in the constrains to enforce the **5th rule on section 1**.

2.2 Constraints

Now, we will first develop the set of constraints that will guarantee the desired rules specified in section **1. Problem Overview**.

Using the definition of $Z[k]$ in **k,Section 2.1**, we have the first two set of constraints.

The first set of constraints is *Zdefn1*. For every teacher k in the TEACHER set, we will have that

$$\sum_{l \in \text{STUDENT}} C[k,l] - 7 \leq Z[k].$$

The second set of constraints is *Zdefn2*. For every teacher k in the TEACHER set, we will have that

$$7 - \sum_{l \in \text{STUDENT}} C[k,l] \leq Z[k].$$

The third set of constraints is *Student_Timeslot_binary*. For every student l in the STUDENT set, we will have

$$\sum_{i \in \text{DAY},\ j \in \text{SESSION}} Y[i,j,l] = 1.$$

This constraint is used to guarantee that every student l is only scheduled for 1 and only 1 oral comps session.

The fourth set of constraints is *Student_Timeslot_Count*. For every i in DAY, j in SESSION, and l in STUDENT, we will have

$$\sum_{k \in \text{TEACHER}} X[i,j,k,l] = (3 + \text{TRIPLE}[l]) * Y[i,j,l].$$

This guarantees the number of teacher scheduled for student l at day i session j match with the number of teacher required for that student's board. The TRIPLE$[l]$ constaint for student l is used for the implementing of **the 5th rules on section 1**.

The fifth set of constraints is *Teacher_Clone_Jutsu*. For every i in DAY, j in session and k in teacher.

$$\sum_{l \in \text{STUDENT}} X[i,j,k,l] \leq 1.$$

This is to guarantee that for teacher can only be a chair a student at a particular day and session,

The sixth set of constraints is *Prof_No_Consecutive_Sesh*. For every i in DAY, j in SESSION (exclude the last session), and k in TEACHER. we have that

$$\sum_{l \in \text{STUDENT}} X[i,j,k,l] + X[i,j+1,k,l] \leq 1.$$

This is to guarantee that every teacher does not have two consecutive oral comps session. **This is the 6th rule from section 1.**

The seventh set of constraints is *Prof_Max_Per_Day*. For every i in DAY and k in TEACHER, we have that

$$\sum_{j \in \text{SESSION},\ l \in \text{STUDENT}} X[i,j,k,l] <= 4.$$

This is to guarantee that for every day no teacher has more than 4 oral comps session. **This is the 10th rule from section 1.**

The eighth set of constraints is *Prof_Max_All*. For every k in TEACHER, we have that

$$\sum_{i \in \text{DAY},\ j \in \text{SESSION},\ l \in \text{STUDENT}} X[i,j,k,l] <= \text{UCAP}[k].$$

This is to guarantee that the total number of oral comps session scheduled for professor k does not exceed that professor's limit. By default, we have that for all teacher k in TEACHER UCAP$[k] = 12$. **This is the 9th rule from section 1.**

The ninth set of constraints is *Prof_Is_Busy*. For every i in DAY, j in session, and k in TEACHER, we have that

$$BUSY[i,j,k] * \sum_{l \in \text{STUDENT}} X[i,j,k,l] = 0.$$

This is to force the value of $X[i,j,k,l]$ to be 0 if the value of $BUSY[i,k,k]$ is 1. In other words, nothing will be scheduled at that particular day and session if the professor is busy.

The Tenth set of constraints is *Stud_Is_Busy*. For every i in DAY, j in SESSION, and l in STUDENT, we have that

$$BUSZ[i,j,k] * \sum_{k \in \text{TEACHER}} X[i,j,k,l] = 0.$$

Similar to the previous condition, this guarantees that nothing will be schedule at a particular day and session if the student is busy.

The eleventh set of constraints is *Is_Prof_Student_Pair*. For every k in teacher and l in student, we have that

$$\sum_{i \in \text{DAY},\ j \in \text{SESSION}} X[i,j,k,l] = C[k,l].$$

This is to guarantee that the pair of student and teacher is paired up in the correct day i and session j.

The twelfth set of constraints is *Integrity_P_Nontriple*. For every l in STUDENT and i in {1, 2, 3}, we have that

$$\sum_{k \in \text{TEACHER}} P[k,l,i] = 1.$$

This is to guarantee that each teacher can only be one chair for each student.

The thirteenth set of constraints is *Integrity_P_Triple* . For every l in STUDENT, we have that

$$\sum_{k \in \text{TEACHER}} P[k,l,4] = TRIPLE[l].$$

This guarantee that if a student l is a triple major then there must by one and only one teacher for the fourth chair. On the other hand, if a student l is not a triple major then there must not be any teacher in the fourth chair. This set of constraints is for **the 5th rules on section 1**.

The Fourteenth set of constraints is *Prof_Student_Timeslot*. For every teacher k in TEACHER and student l in STUDENT. we have that

$$\sum_{i \in \text{DAY},\ j \in \text{SESSION}} X[i,j,k,l] = \sum_{i \in \{1,2,3,4\}} P[k,l,i].$$

This makes sure that for each teacher k and for each student l, the sum of the number of scheduled session bee tween k and l equal to the number of time that teacher k is the chair of student l.

The fifteenth set of constraints is set up for **the 1st rule from section 1.** The first rule says that each student will only have 1 professor (i.e. k) in each of the student's major field. Major is represented by e which goes from 1 to mj_c where mj_c is major count. D_t is a major dept and there is 1 equation for each dept.

$$\sum_{k \in D_\mathrm{t}, e \in \{1....mj\_c\}} P(k,l,e) = 1$$

The sixteenth set of constraints is set up for **the 2nd rule from section 1.** The second rule says that the 2nd chair must be from one of the student's minors. Here e is 2 representing the 2nd chair. k belongs to the union of the student's minor departments. This will apply to students with 1 major.

$$\sum_{k \in \cup(D_\text{minor})} P(k,l,2) = 1$$

The seventeenth set of constraints is set up for **the 3rd rule from section 1**. The third rule says that a student's 3rd chair must be from his 3rd major or at large. So we use two variables U and V. In particular, U is binary variable which represents whether the at-large chair is in a certain division. V is an integer which counts the number of all other chairs in each division. For each l in STUDENT, we have that

$$U(l,f)_{f \in \{1,2,3\}} = \big\{0{,}1\big\}$$

$$V(l,f)_{f \in \{1,2,3\}} = \big\{0,1,2,3\big\}$$

$$U(l,f)_{f \in \{1,2,3\}} * (2 + \text{TRIPLE}) + V(l,f)_{f \in \{1,2,3\}} \leq (2 + \text{TRIPLE})$$

The U and V are only for at-large chair which is 3rd chair for double major and 4th chair for triple major. The TRIPLE[l] constaint for student l is used for the implementing of **the 5th rules on section 1**.

The fourth rule says that the at-large chair must be from different division than 1st or 2nd chair. This is implemented just like the 3rd rule.

The eighteenth set of constraints is *No_New_1st_Major_Students*. Let TEACHER1 $= \{k \in$ TEACHER $\mid SNR[k,1]\}$ which is the set of all the first year teacher. Then, we have that

$$\sum_{k \in \text{TEACHER1},\ l \in \text{STUDENT}} \text{SNR}(k,1)\ P_{k,l,1} = 0$$

This summation represents that the first faculty chair has to be a professor k employed at Wabash full time and longer than a year to be student l first chair in the student's comprehension examination. **This is the 7th rule from section 1**

The nineteenth set of constraints is *Maj_Prof_2ndYr_Then_No_New*, which is set up for **the 8th rule in section 1**. The 8th rule says that if our 1st faculty chair, which means they are the department of 1st major, then the 2nd and 3rd faculty chair should not be a new faculty member.

$$\sum \text{SNR}(K,2)\ P_{k,l,1} * (2 + \text{TRIPLE}(k)) + \sum \text{SNR}(k,l)\ Pk,l,2/3/4 \leq\ 2 + \text{TRIPLE}(k)$$

The first summation check if the 1st faculty chair is a second year (We do not need to worry about if the 1st faculty is new or part time because we worried about that in an earlier constraint). Also we multiply the first constraint to make our quadratic equation into a linear equation. This also helps to check if we need a fourth faculty member if we have a student that is a triple major. The second summation count all the first years and part time professor that is not part of the 1st faculty member and check if it violate the 8 rule. Thus we can seen an example of this in table 1.

Table 1 Rule 8 table

1	2	3	4	
0	0	0		Acceptable
0	0	0		Acceptable
1	0	0		Acceptable
1	0	1		Not Acceptable because it violates Rule #8
1	1	1		Not Acceptable because it violates Rule #8

2.3 Objective Function

The objective function for the problem is

$$\text{Minimize} \sum_{k \in \text{TEACHER}} Z[k],$$

where Z is the absolute value of the difference between the average number of oral comprehensive exams per professor and the number of oral comprehensive exams for a particular professor. We can write this as

$$Z[k] = \Big| \sum_{l \in STUDENT} C[k,l] - 7 \Big|.$$

The 7 from equation above is the floor of the average number of oral comps session for each teacher. We found this with the following calculation:

$$\Big\lfloor \frac{\text{Number of students} * 3}{\text{Number of Teachers}} \Big\rfloor$$
$$= \Big\lfloor \frac{189 * 3}{79} \Big\rfloor \qquad = 7.$$

$Z[k]$ is the norm-1 difference. Thus, our object is to minimize the sum of all the norm-1 difference of all the teachers.

3 Implementation

3.1 Information processing and Solving

The dataset that we test our model is the 2016 comprehensive exams. The original exam has 5 different sheets within an excel file. We used a python script to simplify and create two separate csv files, which are `student.csv` and `faculty.csv`. In particular, `init.py` converts `2016.xlsx` to

- `student.csv`, which contains index ID, Wabash Student ID, major id, and minor id,
- `faculty.csv`, which contains index ID, Wabash Teacher ID, Department ID, 1Y (if the teacher is a first year teacher), 2Y (if the teacher is a second year teacher), and UB (upper bound/UCAP).

In order to simplify the csv file, we store the index number of each student, faculty, major, minor and the department, instead of storing the actual names.

Then, we will use another python script to parse the data from `student.csv` and `faculty.csv` to a AMPL code. We feed the two csv files to `ampl_gen.py`, which generates `mock.mod` and `mock.dat`.

AMPL will process `mock.mod` and `mock.dat` and return the result. The AMPL solving procedure writes out to files the values of P and Y that are not 0 - denoting information about when and who would be Oral Comps chairs, which is then parsed. We use `parse_ouput.py` to create the `Schedule.csv` which is in human friendly format.

3.2 Result

The minimum value of sum of norm-1 differences is 105. This would be a reliable method of automated optimization of resource distribution, proving that an ideal solution exists by giving one, or showing that there is none at all. The method would eliminate the need for human efforts in manually assigning the chairs while re-sorting the lists frequently, all of which takes approximately at most 6 minutes. Moreover, this solves the problem all at once, rather than sequentially like the manual method, which would eliminate any kind of bias possible, making the final schedule more random, and even-out. A possible downside is such randomness may not be too welcomed: some professor might be more suitable in one of his departments than another; but in theory, all professors should be adequate to be oral chairs in all of their departments.

There are a few limitations with our model. Currently, the 6th rule in section 1 is strictly enforced, rather than being a recommended condition. Thus, in some very extreme theoretical case, our model might possibly not find a solution. If no solution is found with the constraints described in the paper, the constraints which are implemented to satisfy rule 6 can be dropped so that more feasible solutions are allowed. Another issue is that if there is some kind of special circumstance (i.e. a student can not make it during all possible sections) the constraint equations in the solver need to be edited to account for this. Currently, there is no way to set time-slots for students or professors without changing the model. A third issue is that there is currently no way of partially implementing rule 6. The way the model is set up, either all professors or no professors will have schedules which abide by rule 6.