Murun Enkhtaivan

Final Draft Solution Report for PEAK Project for MATH 335 – Happiness Solution for RAs

Dr. Nathan Axvig

**Introduction and Purpose**

Concordia Residence Life office is in charge of six traditional halls. In each hall, Director’s Assistant creates a schedule based on RAs weekly availability. Currently, RAs fill out online survey, indicating times they are available and unavailable. In this process, the problem is RAs are sometimes assigned to shifts that they do not like and have to work the same shift for the entire semester. Therefore, they give up their assigned shift to someone else, increasing their opportunity cost and causing some people to take many shifts. The purpose of the project is to evenly distribute total shifts with less conflict and give RAs their preferred shifts, while also keeping the number of desk workers from 5-7. Our solution to the problem is, on online survey, ask RAs available, unavailable, and preferred shifts. We have created a happiness level - objective function and wanted to increase RAs’ satisfaction level with the schedule using big M method.

**Description of the Problem**

Our main goal in this project is to use linear programming to minimize the conflicts of schedule provided by the workers (Resident Assistants) which will in result increase the ‘happiness’ or willingness to work for RAs while making sure that all available time-slots are filled. When creating a schedule, DAs has to consider which day each RA is on the duty. Because RA cannot be on the duty and work at the information desk at the same time, DA has to consider that time when s/he is scheduling. DA makes RAs duty schedule too. Therefore, to simplify the problem, we suggested DA to make the duty schedule first and let RAs know. Then, ask RAs to fill out online available survey, which will be availability that duty hours are incorporated to. To get to know more about the problem, we met DA of Erickson Hall, Michelle Ziperovich couple times. We figured out that number of people working in the desk should be kept between 5-7 people to avoid additional cost to train more people. There is a Res Life rule that a desk worker cannot work for 5 hours in a row. On Monday through Thursday, there are not total shifts worth 5 hours. Therefore, we had to incorporate 5 hours rule to make a schedule for Friday through Sunday shifts.

**Methodology**

We used different aspects of Operations Research principles to come up with a model to make our goal work. We used R studio to code our model that would come up with a feasible solution that fulfills all of our criteria. We also extensively made use of Excel for extracting and storing user data. Our main objective is to assign all shifts with appropriate availability of workers while considering the following constraints:

1. All shifts should be filled by exactly 1 person.

2. A worker is not allowed to work more than 4 consecutive hours.

3. Student workers’ availability as from the survey (example figure (i)).

4. Each student worker cannot work more than 5 hours in a week.

We have constraints that do not justify the work our model does. By implementing operations research technique known as the Big M method, our model penalizes the unavailability option with a higher value and renders it ineffective in the model. Moreover, ‘1’ and ‘0’ are assigned to preferred times and available (but not preferred) times respectively. After the juicy part of finding the best shift for an employee, the model’s objective function calculates the happiness level of the group by multiplying the resulted value from the model of either ‘0’ or ‘1’ for each employee with the preferred value (‘0’ or ‘1’) provided in the schedule for the shift and adding the same for all employees.

MATH

max.

*s.t.*

All shifts should be filled by exactly one person:

.  
.  
.  
.

Any worker is not allowed to work 5 or more consecutive hours:

.

.

.

.

.

.

.

.

.

Student worker’s availability:

Each student worker cannot work more than 5 hours in a week.

.

.

.

.

.

.

.

Where

i = RAs (1…….n);

j = Available Shifts (1……..m);

k = Days (1(for Monday)………(7 for Sunday))

Cij  = Preference/Availability/Unavailability from RAs (figure (i));

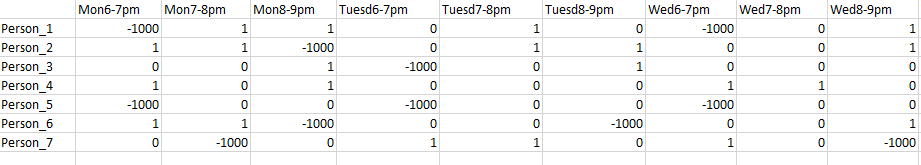
Xij = Solution for preferred schedule;

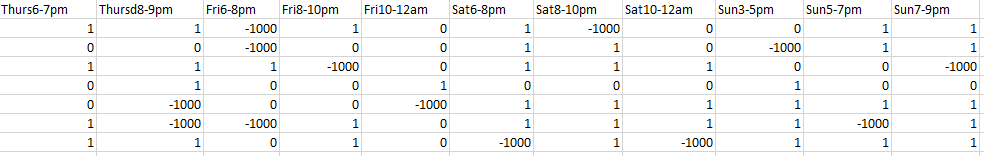
Hj = Hours in a shift;

Hkj = Hours of all shifts in a day;

Cij x Xij = Satisfaction/Happiness

As an example, we have demonstrated the use of our linear program in computing schedule for a hypothetical deskwork that is semantically correct with the original schedule that DAs usually work with. We also made an extra file for hours contained in each shifts so that we can have a more in-depth interface to work with.

Figure (i) shows the working schedule of preference of working times.



‘’-1000’ represents the time preferred, 1’ represents the time available and ‘0’ represents the time not available.

Fig (i)





Fig (ii)

After running the r-script that would extract the data from the excel files, we organize and create various data variables for organization while creating the linear program. Various programming, mathematical and logical skills were required in building the objective function, constraints and inequalities for the script.

The following is just a snippet of formulation of one of the five constraints which shows the overview of various techniques used in it.

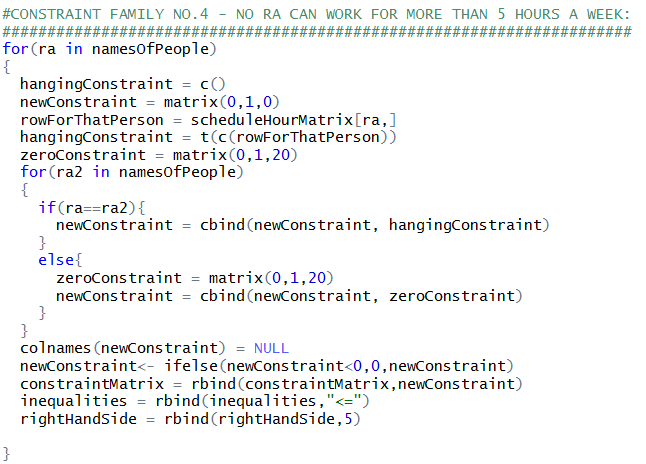


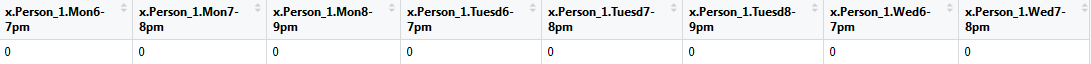
Fig (iii)

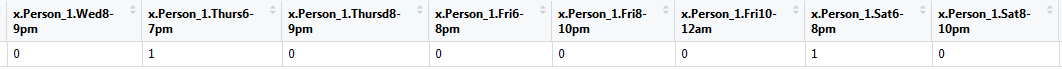
At the end of the script, a linear programming object linked with all the constraints that are also created and stored in the script will handle the solution. The Big M method that has been used is interweaved with the program and takes high consideration of not assigning shifts that are not preferred. The following is the objective value that is created by the program, which in our case will always be maximum as our program will achieve it for not having a complicated and lengthy schedule and having to maximize the happiness level of working staffs.

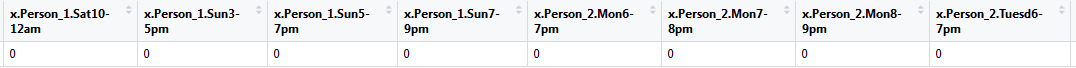


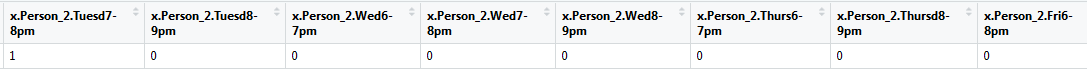
Fig (iv)

The most important part of our programming is when we find the solution for the schedule that is to be refined and structured with every worker getting the shifts they preferred while also remaining under various constraints set by residence life.









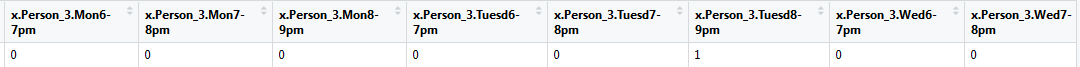
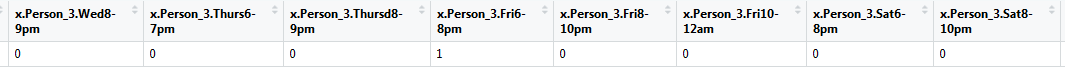
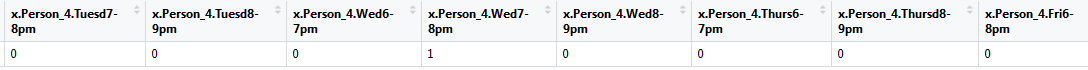
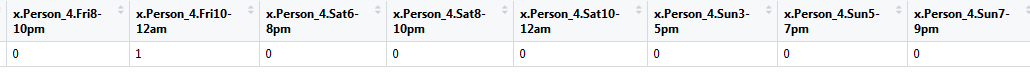
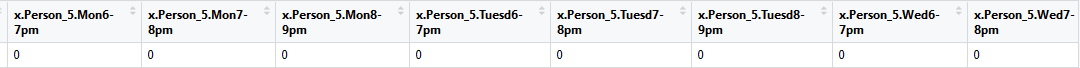
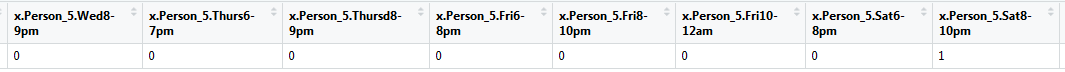
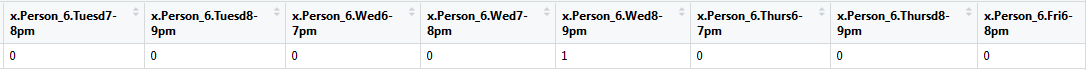
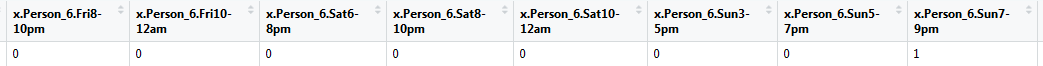
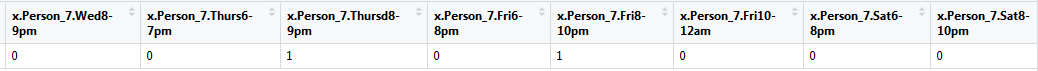
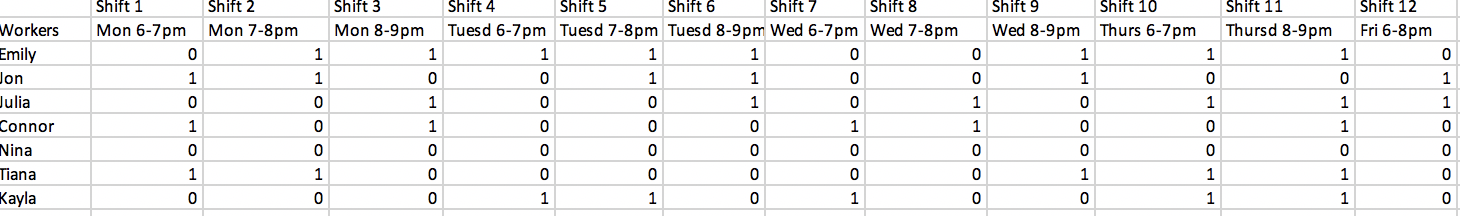
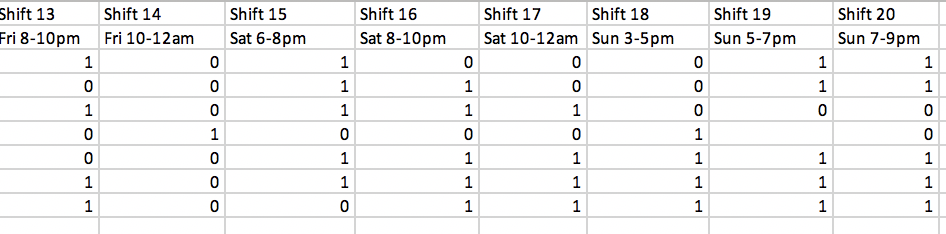
             

Fig (v)

**Results, Recommendations, and Instructions**

From our results, we would recommend the Residence life to use our model in the future to assign the workers at the desk. This model seamlessly works to erase all potential conflicts of the workers and assign an appropriate time slot to work.



‘1’ represents the time available and ‘0’ represents the time not available.

Fig (vi): Current Availability of RAs

Our model is beneficial and flexible as it allows DAs to make changes to the schedule whenever required. This eventually helps the DAs to come up with new schedules anytime when needed. For example, during finals week, choirs or other campus events which requires high student participation, the DAs can update the availability of RAs if needed and get a model to assign the required shifts. DAs can also take the benefit of the model if they bring in a new worker. However, our model still lacks the effectivity to take account that changes in RAs schedule directly. Instead DAs have the responsibility of assigning RAs to duty and only then they can be filled in for generating desk schedule. Therefore, a further extension of our project can be including the duty schedule and put RAs in schedule accordingly. However, it would require higher calculations, complications and constraints as all RAs are expected to be conducting duty, but only a handful of them can be at desk.

Our model will target to get maximum result except in the case of unexpected scenario like no RA preferring to work in a shift although available. Our model can only break if no one is available to work a shift thus giving indication for the DAs to assign new workers. Our model can get complicated if there are more complicated student or there are schedules that change days in the calendar. Our program is not functioned with the option for workers to work under five hours between two days because the current schedule restricts that event. However, the program is robust in the sense that it does what it needs to and in a effectively and satisfying manner.

**Conclusion**

Our final optimal solution is binary assigning values of 1 for which the worker should take the shift and 0 for which they should not and someone else should be preferred before them without any conflict. Although, it might be difficult for DAs to adjust to the program, there is a mutual benefit for the RAs and DAs in a long-term basis as the RAs get their preferred shifts and the DAs do not have to stress about doing it on their own.