

OPTIMAL GROUP CONSTRUCTION

PHILIP MENECHINI

ABSTRACT. This quick report shows how optimal groups for six of seven projects can be constructed optimally based on preferences, skills, and availability. This report gives a detailed look at the constraints and variables in the integer program that can solve this group construction problem.

INTRODUCTION

This project was very interesting in the sense it was a project assigning people to projects. My task was to take all the twenty four people in this class and put them in "optimal" groups which each had a project that they needed to complete. First I needed to collect data on project preferences, skills and time availabilities in order to have a quantifiable way of optimizing these groups. After that I constructed an integer program that created groups that were optimal from this data. I had the following constraints when making groups.

- (1) There must be between six and seven projects assigned
- (2) There must be between three or four people in each group
- (3) Each group must have one "expert" at a certain skill (ranked themselves above a six on the skill)
- (4) All members of the group must be able to meet for at least two hours a week

VALUES:

- p_{ij} is a matrix which has all the preferences for each person on each project (each project is ranked from one to ten). Person i ranks project j by whatever number is in p_{ij} .
- $skill_{ig}$ is a matrix with all the skill data. This data is converted to a binary data type where a person i is considered an expert at skill g if there is a one in that entry, a zero if not. We want each of the skills to be in each group.
 - coding
 - writing
 - Latex
 - presenting
 - leadership
 - organization
 - time management
- $time_{ik}$ is a binary matrix which has a one if person i can meet at hour k (these are times 1-168 each representing a distinct hour in the week).

VARIABLES:

- x_{ij} is a binary variable with a one representing whether person i is in project j .
- y_j is a binary list where a one represents if project j is happening and a zero if it is not happening.

- w_{jk} is another binary matrix which is the opposite of the other binary matrix in that a zero represents that all people in group j can meet at hour i and a zero means the opposite.
- c_j is an integer array which is used for our loose constraint to punish the objective function when one of our skills is not in a group. This variable was included because I was worried the program might be infeasible since I have so many skills I want in each group. This variable must always be positive.

OBJECTIVE FUNCTION:

Now that all of our variables and values are thoroughly described, we can move on to defining the objective function.

The objective function described here is a maximization of how many students got in groups they preferred. We then subtracted our loose constraint variable to penalize any constraints not met.

$$\text{maximize } \left(\sum_{\forall i} \sum_{\forall j} p_{ij} x_{ij} \right) - \sum_{\forall j} c_j$$

CONSTRAINTS:

In order to make sure these groups had all elements I desired, I needed to make a bunch of constraints.

My first constraint was making sure that each person was assigned to exactly one project. I did this using the x_{ij} variable.

$$\forall i \left(\sum_{\forall j} x_{ij} = 1 \right)$$

My second constraint made sure that a student was not assigned to a project which was not happening. This constraint also made sure that each group had between three and four people.

$$\begin{aligned} \forall j \left(\left(\sum_{\forall i} x_{ij} \right) - 4y_j \leq 0 \right) \\ \forall j \left(\left(\sum_{\forall i} x_{ij} \right) - 3y_j \geq 0 \right) \end{aligned}$$

The third constraint imposed the limitation that I want either six or seven projects to happen and have groups. So at most I want one project to not be included if lots of students do not want to do it.

$$\begin{aligned} \sum_{\forall j} y_j &\leq 7 \\ \sum_{\forall j} y_j &\geq 6 \end{aligned}$$

The fourth constraint is the loose constraint I have been talking about. This constraint tries to make sure there is an expert of each skill in each group. How many skills are left

out equals c_j for project j .

$$\forall j \left(\sum_{\forall j} \left(\sum_{\forall i} skill_{ij} x_{ij} \right) + c_j \geq y_j \right)$$

Our hardest constraint is our time constraint making sure that each group gets at least two hours a week to meet. In this constraint we use our w_{jk} variable.

$$w_{jk} \leq \sum_{\forall i} x_{ij} - \sum_{\forall i} time_{ik} x_{ij}$$

$$w_{jk} \geq \frac{\sum_{\forall i} x_{ij} - \sum_{\forall i} time_{ik} x_{ij}}{4}$$

Since this variable is binary and there can only be 4 people in a group this variable will always be 0 if meeting at time k works for all members of the group and 1 otherwise. Now for the constraint that makes sure that each group can meet at least two hours.

$$\forall j \sum_{\forall k} w_{jk} \leq 166 y_j$$

This is so because there are 168 hours in a week and we want at least two zero entries in each project j of our w_{jk} variable.

Our last constraint is just there to make sure that the program does not cheat and assign negative values to c_j in order to increase the objective function.

$$\forall j c_j \geq 0$$

COMPUTATIONAL RESULTS AND ANALYSIS

After implementing the program in code I achieved the following groups which can be used for this project assignment problem. Below I have each member in each group as well as the times on each day that all group members can meet.

Group 0: Brendan, Kalju, Philip, Sarah,

meeting times: day: sunday hour: 15 (military) day: sunday hour: 16 (military) day: monday hour: 16 (military) day: monday hour: 17 (military) day: monday hour: 18 (military) day: monday hour: 19 (military) day: tuesday hour: 11 (military) day: tuesday hour: 19 (military) day: tuesday hour: 20 (military) day: wednesday hour: 17 (military) day: wednesday hour: 18 (military) day: thursday hour: 18 (military) day: thursday hour: 19 (military) day: thursday hour: 20 (military) day: saturday hour: 14 (military) day: saturday hour: 17 (military) day: saturday hour: 18 (military) day: saturday hour: 19 (military) day: saturday hour: 20 (military)

Group 1: project not happening

Group 2: Emma, Maura, Maxwell, Sejal,

meeting times: day: sunday hour: 15 (military) day: sunday hour: 16 (military) day: monday hour: 16 (military) day: monday hour: 17 (military) day: monday hour: 18 (military) day: monday hour: 19 (military) day: tuesday hour: 11 (military) day: tuesday hour: 12 (military) day: tuesday hour: 16 (military) day: tuesday hour: 17 (military) day: tuesday hour: 18 (military) day: tuesday hour: 19 (military) day: tuesday hour: 20 (military) day: wednesday hour: 17 (military) day: wednesday hour: 18 (military) day: wednesday hour: 19 (military) day: wednesday hour: 20 (military) day: thursday hour: 13 (military) day: thursday hour: 14 (military) day: thursday hour: 15 (military) day: thursday hour: 16 (military) day: thursday hour: 17 (military) day: thursday hour: 18 (military) day: thursday hour: 19 (military) day: thursday hour: 20 (military) day: friday hour: 12 (military) day: friday hour: 13 (military) day: friday hour: 14 (military) day: friday hour: 15 (military) day: friday hour: 16 (military) day: friday hour: 17 (military) day: friday hour: 18 (military) day: friday hour: 19 (military) day: friday hour: 20 (military) day: saturday hour: 12 (military) day: saturday hour: 13 (military) day: saturday hour: 14 (military) day: saturday hour: 15 (military) day: saturday hour: 16 (military) day: saturday hour: 17 (military) day: saturday hour: 18 (military) day: saturday hour: 19 (military) day: saturday hour: 20 (military)

Group 3: Alex M., Arjun, Matthew, Yongzhi,

meeting times: day: monday hour: 18 (military) day: monday hour: 19 (military) day: tuesday hour: 18 (military) day: wednesday hour: 20 (military) day: thursday hour: 14 (military) day: friday hour: 14 (military) day: friday hour: 19 (military) day: saturday hour: 14 (military) day: saturday hour: 18 (military) day: saturday hour: 19 (military)

Group 4: Alex R., Parker, Samuel, Yuchen,

meeting times: day: sunday hour: 15 (military) day: sunday hour: 16 (military) day: monday hour: 17 (military) day: monday hour: 18 (military) day: tuesday hour: 11 (military) day: tuesday hour: 12 (military) day: thursday hour: 13 (military) day: thursday hour: 14 (military) day: thursday hour: 15 (military) day: thursday hour: 16 (military) day: thursday hour: 17 (military) day: saturday hour: 16 (military) day: saturday hour: 17 (military) day: saturday hour: 18 (military) day: saturday hour: 19 (military) day: saturday hour: 20 (military)

Group 5: Evan, Hanna, Khai, Nathan,

meeting times: day: sunday hour: 15 (military) day: sunday hour: 16 (military) day: monday hour: 16 (military) day: monday hour: 17 (military) day: monday hour: 18 (military) day: monday hour: 19 (military) day: tuesday hour: 11 (military) day: tuesday hour: 12 (military) day: tuesday hour: 16 (military) day: tuesday hour: 17 (military) day: tuesday hour: 18 (military) day: tuesday hour: 19 (military) day: tuesday hour: 20 (military) day: wednesday hour: 17 (military) day: wednesday hour: 18 (military) day: wednesday hour: 19 (military) day: wednesday hour: 20 (military) day: thursday hour: 13 (military) day: thursday hour: 14 (military) day: thursday hour: 15 (military) day: thursday hour: 16 (military) day: thursday hour: 17 (military) day: thursday hour: 18 (military) day: thursday hour: 19 (military) day: thursday hour: 20 (military) day: friday hour: 12 (military) day: friday hour: 13 (military) day: friday hour: 14 (military) day: friday hour: 15 (military) day: friday hour: 16 (military) day: friday hour: 17 (military) day: friday hour: 18 (military) day: friday hour: 19 (military) day: friday hour: 20 (military) day: saturday hour: 12 (military) day: saturday hour: 13 (military) day: saturday hour: 14 (military) day: saturday hour: 15 (military) day: saturday hour: 16 (military) day: saturday hour: 17 (military) day: saturday hour: 18 (military) day: saturday hour: 19 (military) day: saturday hour: 20 (military)

Group 6: Elisabeth, Erica, Lorene, Yongkai,

meeting times: day: sunday hour: 15 (military) day: sunday hour: 16 (military) day: monday hour: 16 (military) day: monday hour: 17 (military) day: monday hour: 18 (military) day: monday hour: 19 (military) day: tuesday hour: 11 (military) day: tuesday hour: 12 (military) day: tuesday hour: 16 (military) day: tuesday hour: 17 (military) day: tuesday hour: 18 (military) day: tuesday hour: 19 (military) day: tuesday hour: 20 (military) day: wednesday hour: 17 (military) day: wednesday hour: 18 (military) day: wednesday hour: 19 (military) day: wednesday hour: 20 (military) day: thursday hour: 13 (military) day: thursday hour: 14 (military) day: thursday hour: 15 (military) day: thursday hour: 16 (military) day: thursday hour: 17 (military) day: thursday hour: 18 (military) day: thursday hour: 19 (military) day: thursday hour: 20 (military) day: friday hour: 12 (military) day: friday hour: 13 (military) day: friday hour: 14 (military) day: friday hour: 15 (military) day: friday hour: 16 (military) day: friday hour: 17 (military) day: friday hour: 18 (military) day: friday hour: 19 (military) day: friday hour: 20 (military) day: saturday hour: 12 (military) day: saturday hour: 13 (military) day: saturday hour: 14 (military) day: saturday hour: 15 (military) day: saturday hour: 16 (military) day: saturday hour: 17 (military) day: saturday hour: 18 (military) day: saturday hour: 19 (military) day: saturday hour: 20 (military)

As one can see the results created six groups instead of seven because so many people disliked project two.

CONCLUSION

Overall this program works very well at maximizing student's preferences. In changing group or student conditions constraints can be tweaked to whatever is desired in a group and the values can change according to the students. Perhaps my next project will be making a more general version of this program instead of just for one class and specific circumstance as I did. The results look very promising and perhaps a system like this can be used to create groups in all classes.

DEPARTMENT OF MATHEMATICS AND STATISTICS, UNIVERSITY OF MASSACHUSETTS, AMHERST