Operations Research, Spring 2020 (108-2)
Case Assignment 2

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

Facing the CSR scheduling challenge and relevant data described in Case Assignment 1, Mikasa now wants to find a systematic way to solve this problem. What she wants is not just an *answer*, which is a schedule for a specific month. What she wants is a *solution*, i.e., a model or an algorithm that can produce a schedule given the rules and data for any month. As the head of the Operations Research team, you now want to apply the recently learned *Linear Programming* to construct a decision support tool.

You understand that Linear Programming has its limit. First, while Linear Programming can determine the number of CSRs to be assigned to each shift in each day, it cannot restrict the number to be integer. You decide to ignore this issue for a while. After all, getting a suggestion with fractional numbers is better than getting nothing.

The second issue is more critical. While Linear Programming can determine the number of CSRs to be assigned to each shift on each day, it cannot specifically determine *which* CSRs should be assigned to that shift on that day. For example, it may tells you that three CSRs should be assigned to shift 1 on March 1, but it cannot tell you which three CSRs should be done so. This is indeed an issue because then you have no way to ensure each CSR to have enough days off, not too many night and afternoon shifts, etc. You also will ignore this issue for a while.

Instead of ignoring those realistic constraints directly, you will find some *proxies* for them. Below we will describe how this should be done. Your task is then to (1) *formulate* this CSR scheduling problem into a Linear Program by relaxing some of the realistic constraints, and (2) *solve* the Linear Program you formulate. The optimal solution you obtain by solving your Linear Program, though cannot be executed directly, will still help Mikasa do decision making.

Please always remember the words by George Box: "All models are wrong, but some are useful." There is no way to have a model that is 100% realistic. All we can do, and all we want to do, is to construct a model that can help us make better decisions.

Okay, let's start!

1 Formulation with proxies 代理

1.1 Decisions and their proxies find decision variable

To formulate a Linear Program, the first question is always to ask "what are the decisions we want to make?" For the CSR scheduling problem, it is obvious:

> For each shift, for each day, we want to determine *which CSR* should be assigned to that shift on that day.

As we mentioned above, this cannot be found with a Linear Program. Therefore, we will deal with a proxy of this decision:

> For each shift, for each day, we want to determine *how many CSRs* should be assigned to that shift on that day.

You may see that the proxy decision, which Linear Programming is capable to deal with, helps us get closer to the real decision we want to make. Please start defining your decision variables based on this proxy decision. When you find you need to define more decision variables, please do so.

1.2 Objective

Recall that our objective function is to minimize the total lack amount across all periods in the coming month. With our proxy decision, this may be perfectly calculated. As long as you know the number of CSRs assigned to each shift on a day, you know how many CSRs will be on duty in each period. You then know the lack amount in each period. Therefore, we do not need a proxy for the objective.

1.3 Constraints and their proxies

There are many constraints mentioned in Case Assignment 1. Let's deal with them one by one.

The first constraint is *each CSR should have eight days off.* As we cannot have a specific CSR-shift relationship, there is no way to formulate this constraint in our Linear

Program. What we may (should) do is to ensure that the total number of off days across all CSRs should follow this constraint. As there are forty CSRs, in total there should be $8 \times 40 = 320$ times that CSRs are assigned to shift 0 in this month. Therefore, our proxy for this constraint is 1 *in total there should be 320 CSRs assigned to shift 0 in this month.*

The next two constraints are about 2 *shift requests* and 3 *leave requests* (please take a look at the two MS Excel sheets to refresh your memory). As there is no way to assign a specific CSR to a shift on a specific day, our proxy for this constraint is that *an enough number of CSRs should be assigned to that shift on that day.* In particular, at least one CSR should be assigned to shift 5 on March 10, at least two CSRs should be assigned to shift 0 on March 20, etc.

The next two constraints are about 4 *senior limit* and 5 *manager limit* (please take a look at the two MS Excel sheets to refresh your memory). As most of the CSRs have at least two years of experience, and the manager limit seems not a big deal, we will simply ignore them.

The next constraint is 6 *a CSR can have at most one night shift in seven consecutive days.* This means a CSR can have roughly $\frac{31}{7} \approx 4.43$ night shifts in a month, and $4.43 \times 40 = 177.2$. To leave some buffer, our proxy for this constraint is *there can be at most 177 CSRs assigned to night shifts in this month.* Similarly, for the afternoon shifts, our proxy is *there can be at most 354 CSRs assigned to afternoon shifts in this month.*

Finally, 7 *a CSR must have at least one day off in seven consecutive days.* We follow the same idea as above to construct our proxy as *there should be at least forty CSRs assigned to shift 0 for every seven consecutive days.* More precisely, from March 1 to March 7 at least forty CSRs should be assigned to shift 0, from March 2 to March 8 also at least forty CSRs should be assigned to shift 0, etc.

We again ignore Richard's comments.[1]

2 Your tasks

1. (40 points) Following the above construction of proxies, please formulate a Linear Program that may find an optimal allocation of CSRs. You may have noticed that as your Linear Program can only have fractional variables, which seems to be weird for CSR allocation and the calculation of lack amounts. In this Cast Assignment,

[1] Do not be late for a meeting!

let's ignore this issue. When one needs to produce a real schedule, she/he will do some rounding and manual adjustment. In short, please formulate a "Linear" Program and do not have non-fractional variables.

For this problem, please write down a compact mathematical formulation. Do not submit a computer program.

2. (30 points) Submit a (set of) computer program that may solve the Linear Program you formulated in Problem 1. You may submit an AMPL model file and an AMPL data file. Alternatively, you may submit a (set of) Python or C++ programs that invoke gurobi to solve this problem.

3. (10 points) Please summarize the optimal solution you obtain with your computer programs submitted in Problem 2. You should summarize the allocation of CSRs and the total lack amount returned by your computer programs.

4. (20 points) Recall that the company is considering to adjust the number of CSRs. Is 40 the right number? How about 38, 35, or 50? Our programs may now help us tackle this issue. Please rerun your programs several times, each time with a different number of CSRs, to see how the number of CSRs affect the total lack amount. In particular, please generate a plot depicting the total lack amounts when the number of CSRs is 35, 36, ..., and 45. Then make some comments on whether the company should increase or decrease the number of CSRs. Please note that when you change the number of CSRs, some numbers in the proxy constraints should also be changed accordingly.

This case assignment counts for 7.5% in calculating the semester grades.

3 Submission rules

- **Teams**. Students should form teams to work on this case study. Each team should have three to five students. The team formation have been done previously. Please keep working with the team members you previously made commitment to.

- **Things to submit**. Please submit a set of computer programs (for Problem 2 above) and a PDF file (for the remaining problems). Include the student IDs and names of all team members in the PDF file.

- **How to submit**. Please include all the files (regardless of the number of files) you want to submit into a single ZIP file. Please submit the ZIP file to NTU COOL. Each team should make only one submission, i.e., only one student should make a submission.
- **Deadline**. The deadline of this assignment is ***1:00 AM, April 16***. Works submitted between 1:00 and 2:00 will get 10 points deducted as a penalty. Submissions later than 2:00 will not be accepted.

4 Final note

While this case assignment look like an assignment, from the instructor's perspective it is more like a demonstration. It demonstrate how a real problem may be modeled, how some real-world things should be ignored or replaced by proxies, and how a model may help people do decision making. Please note that modeling is a technique but also an *art*, i.e., there can be many different ways to model a real problem. Maybe you may find a better way to model this problem, but in this Case Assignment, please follow the way we specify. In your final project, you will have the full right to model a real problem by the way you like!