Week 12: Reusable Software

Session 24: Creating Reusable Software for Shift Scheduling

Problem Description

Imani is in charge of scheduling shifts for all nurses at the emergency department at Trojan hospital. **In order to retain quality staff, the hospital would like to create schedules that satisfy their preferences as much as possible, while respecting operational constraints and treating nurses fairly.** In the past, she created schedules by hand, which is very time consuming and she was never sure whether she has found the best schedule. Recently, after learning about linear optimization modeling, she would like to apply her knowledge to create an optimization tool that can be used to automatically generate the best schedule.

Based on her experience, she has created a template input file to store all of the relevant input data. See the `12-scheduling-input-1.xlsx` file for a smaller file for development purposes. The file contains 9 nurses and 1 week. See the `12-scheduling-input-2.xlsx` for a larger file containing a realistic instance, with 50 nurses and 9 weeks.

Ideally, a schedule should maximize the total preference scores of nurses. The preference scores are specified in the input data in the sheet called "Preferences". For each shift, a nurse may indicate a preference score of 0, 1, or 2. A score of 0 indicates that the nurse is unavailable to work that shift. A score of 1 indicates that the time is not preferred, but the nurse is willing to work if needed. A score of 2 indicates a preferred time slot.

Operational constraints:

- There are three nurse shifts in a day: morning, evening, and nights. Each shift should have exactly a certain number of nurses, which is specified in the input file in the sheet called "Requirements". You may assume that Shift 0 corresponds to a morning shift, shift 1 to an evening shift, shift 2 to a night shift, shift 3 to a morning shift, etc. Moreover you may assume that the total number of shifts is a multiple of 3.
- If a nurse works in a night shift, he/she must take the two prior shifts off as well as the two next shifts off. For example, if a nurse works on Shift 2, then she must not work in shifts 0, 1, 3, and 4. If a nurse works on Shift 5, then she must not work in shifts 3, 4, 6, and 7. This ensures that nurses get proper rest.
- No nurse may be scheduled to consecutive shifts. However, it is possible for example to work both shifts 1 and 3, as they are not consecutive and neither is a night shift.
- If a nurse indicates that he/she is unavailable to work in a given shift (i.e. preference score of 0 for a shift), then the nurse cannot be assigned to that shift.

Fairness constraint: The total number of shifts assigned to a nurse must be similar across nurses. Precisely speaking, there is a certain inequality k, and the total number of shifts worked by any given nurse must be within k of each other. For example, if $k = 2$, then the constraint would be violated if Alice is assigned 10 shifts and Bob is assigned 7 shifts. But it would not be violated if every nurse is assigned between 8 and 10 shifts (inclusive).

Write a function called `computeSchedule` with three input parameters:

- **inputFile**: an Excel file of the same format as the `12-scheduling-input-1.xlsx` and `12-scheduling-input-2.xlsx` files.
- **k**: the inequality parameter in the fairness constraint.
- **outputFile**: an output file generated by the function which contains an optimal schedule in one sheet. In the other sheet, the file contains the optimal objective value, the value of k, as well as the optimal objective value divided by the total number of shifts assigned, which can be interpreted as the average preference score for an assigned shift. For sample outputs, see the `12-scheduling-output-1.xlsx` and `12-scheduling-output-2.xlsx`

files attached on Blackboard. Note that there may be multiple optimal schedules, so your code may output a different schedule for the same value of k, but the optimal objective value must be the same.

In-Class Exercise: English Description and Concrete Formulation

On a piece of paper, jot down a succinct summary of the English description as well as fragments of the concrete formulation. You don't have to hand in anything. The purpose is to help you develop a clear idea of how to formulate the abstract formulation later. Think of this as doing scrap work on an exam.

ENGLISH

Decision: determine which nurse must be assigned to which shift.

Obj & Constraints:

Maximize the total preference score

- Nurses must be assigned according to preferences.
- A nurse in night shift must have preceding & following 2 shifts off.
- Each shift should have required number of nurses.
- No consecutive shifts.

CONCRETE:

Decision Variable: X_{ij} : whether nurse i must be assigned to shift j.

P_{ij}: preference of nurse i for shift j.

Obj & Constraints: minimize: $P_{10}X_{10} + P_{11}X_{11} + \dots P_{nm}X_{nm}$

(No consecutive) $X_{11} + X_{12} \le 1$ $\Rightarrow$ for Alyssa

$X_{10} + X_{11} \le 1$

(Min Nurse Req) $X_{1,1} + X_{2,1} + \dots + X_{n,1} = r_j$ (r in R for shift j)

(Night Shift Rest) $X_{Al,0} + X_{Al,2} \le 1$, $X_{Al,4} + X_{Al,2} \le 1$ .....

(Preference) $X_{ij} \le P_{ij}$

$y_{ale} = X_{ale,0} + X_{ale,1} + \dots$

$L \le y_{ale}, y_{aly}, y_{ant} \le L + k$

Exercise 12.3 Abstract Formulation for Shift Scheduling

Download the Jupyter notebook attached to the Blackboard link for this exercise and submit it there after completing it. The notebook asks you to write an abstract formulation for the above problem. To help you get started, below are some data variables that may be helpful. You don't have to use these and you can add other ones as you see fit.

Data:

- I: the set of nurses.
- n: the number of days being scheduled.
- J: set of shifts, $J = \{0, 1, 2, \cdots, 3n-1\}$.
- J_{night}: set of night shifts, $J_{night} = \{2, 5, \cdots, 3n-1\}$.
- p_{ij}: the preference score of person i for shift j.
- q_j: the number of nurses needed for shift $j \in J$.
- k: the inequality parameter in the fairness constraint.

Decision Variables:

x_{ij} : whether nurse i should be in shift j.

L : for fairness constraint.

Objective:

Maximize $\sum x_{ij} p_{ij}$

Constraints:

Exercise 12.4 Reusable Software for Shift Scheduling

Create a new Jupyter notebook and write in it your function `computeSchedule`, following the instructions in the problem description. Make sure you put all of your final code in one cell with the comment `# Final Code` such that if the Kernel is restarted and only that cell is run, the code will work.

As seen in the test code below, it would be a good idea if the function prints something to let you know it has finished. You can do something similar to what is suggested in the sample outputs, but it is not mandatory.

[1]:
```
# Final Code
```

[4]:
```
# Test code
computeSchedule('12-scheduling-input-1.xlsx',3,'12-scheduling-myOutput-1.xlsx')
```

```
Finished Optimizing! Objective value: 83.0
```

[3]:
```
computeSchedule('12-scheduling-input-2.xlsx',1,'12-scheduling-myOutput-2.xlsx')
```

```
Finished Optimizing! Objective value: 4200.0
```