Improved custom solution for hospital assignment problem during COVID-19

Grecu Cristian

May 2023

Introduction

Studying the solution proposed in the paper *Challenges and solutions for the integrated recovery room planning and scheduling problem during COVID-19 pandemic*, published by Marouane Chaieb, Dhekra Ben Sassi, Jaber Jemai and Khaled Mellouli, we decided to implement a custom solution for this assignment problem. This solution broadly follows the original solution proposed by the paper, with some major differences in terms of problem and constraints definitions.

Defining the improved problem

In this subchapter, we will provide a detailed description of how we defined the problem and the constraints. While in the original paper they used 2 possible types of patients (critical and severe), which are correlated with 2 possible types of nurses (one type of nurse for each type of patient), we decided to define the problem for a generic nurse and a generic patient. After all, we can solve the original problem by splitting it into 2 branches: one with critical patients and the specialized nurses and one with the severe cases, treated by other nurses.

Because we can not use the data used for the paper because it is sensible and provided by the government, we decided to generate data to simulate a real life scenario. The variables that we used in the problem are:

- **num_days** - number of days for which data will be generated
- **num_nurses** - number of nurses
- **num_patients** - number of patients
- **num_shifts** - number of shifts (usually 3 shifts of 8 hours each)
- **max_treatment_duration** - maximum duration for a treatment (in chunks of 10 minutes)
- **max_working_time** - maximum working time for a nurse (usually 8 hours)
- **max_shifts_worked** - maximum number of shifts worked per nurse
- **min_shifts_worked** - minimum number of shifts worked per nurse
- **min_nurses_per_shift** - minimum number of nurses per shift

Data is generated randomly for patients because this best reflects the real life scenario. The assignment of nurses is solved using a secondary constrained assignment problem, with the observation that nurse data is generated randomly, like patient data, in case of an infeasible solution. An entry for a patient has the form {(**day, patient): [shift, treatment_duration]**} while an entry for a nurse has the form {(**day, nurse): shift**}. The patients are splitted equally to shifts to make the problem balanced. Every day, a patient or a nurse can be assigned to different shifts.

In the following, we will provide the constraints for the 2 assignment problems. After the solution for the nurses assignment problem is returned, it is used to calculate the solution for the hospital assignment problem.

The constraints that should be respected for nurses assignment problem are:

- A nurse should work at most 1 shift per day.
- There should be a minimum number of nurses per each shift, each day.
- Each nurse should work between a minimum and a maximum number of shifts, during the experiment; we can consider working a shift as working a day.

The objective of this problem is to maximize the number of shifts worked. More shifts worked influences the volume of patients which can be treated.

The constraints that should be respected for hospital assignment problem are:

- A patient should be treated by at most one nurse in a day.
- A nurse can not work more than maximum working time per day(8 hours); this time is calculated as the sum of treatment durations per patient in a day.
- A patient with a scheduled treatment in a shift should be treated only by a nurse who is working in the same shift (patient_shift = nurse_shift).

The objective of the problem is to maximize the number of treated patients during the number of days defined.

Proposed solution and one example of assignment

The solution is implemented in Python and it uses the OR Tools framework and Cp Solver.

For the nurse assignment problem, we defined a dictionary called nurses assignment of the form {(**day, nurse, shift**): **BoolVar**} where True value corresponds to the following predicate: "“On day **day**, nurse **nurse** will work shift **shift**.”. The constraints are defined using this dictionary.

For the hospital assignment problem, ee defined a dictionary called hospital assignments of the form {(**day, nurse, patient**): **BoolVar**} where True value corresponds to the following predicate: “On day **day**, nurse **nurse** will treat patient **patient**”. The constraints are defined using this dictionary.

Another important and improved aspect of our solution is that instead of printing in the terminal, we provide 3 **html files** corresponding to nurses assignment, patients assignment and hospital assignment. We consider that this is a better way to visualize the solution and check the extent to which constraints have been respected.

In the following, we present a possible scenario and the resulting assignment. For the ease of analyzing the assignment tables, we initialize variables with small numbers.

Possible scenario

num_nurses = 5
num_shifts = 3
num_days = 7
num_patients = 20
max_treatment_duration = 10 *# in 10 minutes chunks*
max_working_time = 8 * 60 *# in minutes*
max_shifts_worked = 5
min_shifts_worked = 3
min_nurses_per_shift = 1 *# less than num_nurses // num_shifts*

Results

On day 1, number of treated patients: 20 / 20
On day 2, number of treated patients: 18 / 20
On day 3, number of treated patients: 19 / 20
On day 4, number of treated patients: 19 / 20
On day 5, number of treated patients: 20 / 20
On day 6, number of treated patients: 20 / 20
On day 7, number of treated patients: 19 / 20

Nurses assignment table

	day_1	**day_2**	**day_3**	**day_4**	**day_5**	**day_6**	**day_7**
nurse_1	**2**	**3**	**3**	**-**	**-**	**1**	**1**
nurse_2	**3**	**1**	**3**	**1**	**-**	**2**	**-**
nurse_3	**1**	**-**	**2**	**2**	**2**	**-**	**2**
nurse_4	**2**	**2**	**3**	**-**	**1**	**3**	**-**
nurse_5	**-**	**-**	**1**	**3**	**3**	**2**	**3**

Patients assignment table

	day_1	day_2	day_3	day_4	day_5	day_6	day_7
patient_1	2	2	3	3	1	2	1
patient_2	3	3	2	3	3	3	2
patient_3	1	2	1	2	1	2	1
patient_4	3	3	1	3	2	1	1
patient_5	2	1	3	3	1	1	1
patient_6	2	2	3	1	2	2	3
patient_7	2	3	2	3	2	2	1
patient_8	1	2	1	1	2	2	1
patient_9	3	1	1	1	2	3	2
patient_10	3	2	2	3	2	1	2
patient_11	3	2	1	2	3	3	2
patient_12	2	2	2	1	1	2	1
patient_13	3	1	2	1	3	3	1
patient_14	2	3	3	3	3	2	2
patient_15	3	2	2	1	1	2	1
patient_16	1	2	1	3	2	1	1
patient_17	1	1	1	3	2	2	2
patient_18	2	1	2	2	3	3	1
patient_19	3	2	2	1	1	2	1
patient_20	2	2	2	3	1	2	1

Hospital assignment table

	day_1	day_2	day_3	day_4	day_5	day_6	day_7
patient_1	nurse_1	-	nurse_1	nurse_5	nurse_4	nurse_2	nurse_1
patient_2	nurse_2	nurse_1	nurse_3	nurse_5	nurse_5	nurse_4	nurse_3
patient_3	nurse_3	nurse_4	nurse_5	nurse_3	nurse_4	nurse_2	-
patient_4	nurse_2	nurse_1	nurse_5	nurse_5	nurse_3	nurse_1	nurse_1
patient_5	nurse_1	nurse_2	nurse_1	nurse_5	nurse_4	nurse_1	nurse_1
patient_6	nurse_1	nurse_4	nurse_1	nurse_2	nurse_3	nurse_2	nurse_5
patient_7	nurse_1	nurse_1	nurse_3	nurse_5	nurse_3	nurse_2	nurse_1
patient_8	nurse_3	nurse_4	nurse_5	nurse_2	nurse_3	nurse_5	nurse_1
patient_9	nurse_2	nurse_2	nurse_5	nurse_2	nurse_3	nurse_4	nurse_3
patient_10	nurse_2	nurse_4	nurse_3	-	nurse_3	nurse_1	nurse_3
patient_11	nurse_2	nurse_4	nurse_5	nurse_3	nurse_5	nurse_4	nurse_3
patient_12	nurse_1	nurse_4	nurse_3	nurse_2	nurse_4	nurse_5	nurse_1
patient_13	nurse_2	nurse_2	-	nurse_2	nurse_5	nurse_4	nurse_1
patient_14	nurse_1	nurse_1	nurse_1	nurse_5	nurse_5	nurse_5	nurse_3
patient_15	nurse_2	-	nurse_3	nurse_2	nurse_4	nurse_5	nurse_1
patient_16	nurse_3	nurse_4	nurse_5	nurse_5	nurse_3	nurse_1	nurse_1
patient_17	nurse_3	nurse_2	nurse_5	nurse_5	nurse_3	nurse_5	nurse_3
patient_18	nurse_1	nurse_2	nurse_3	nurse_3	nurse_5	nurse_4	nurse_1
patient_19	nurse_2	nurse_4	nurse_3	nurse_2	nurse_4	nurse_5	nurse_1
patient_20	nurse_1	nurse_4	nurse_3	nurse_5	nurse_4	nurse_5	nurse_1

Evaluation of problem solving time and average percentage of treated patients in different scenarios

We analyze the consistency of our solution by repeating an experiment with the same variables values for **n** number of times and construct 95% confidence interval for time and average number of patients treated per experiment. If n is less than 30, we are using the **student's t-distribution** and if n is greater or equal to 30, we are using **normal distribution**. **APTP_95** stands for average percentage of treated patients 95% confidence interval and **time_95** stands for 95% confidence interval for time. The following table shows the confidence intervals for 3 different experiments. Each experiment is repeated 10, 20 or 30 times.

num_nurses	num_patients	num_days	repetitions	APTP_95	time_95(s)
20	100	7	10	[58.7, 60.3]	[0.565, 0.606]
20	100	7	20	[57.8, 59.6]	[0.551, 0.596]
20	100	7	30	[58.5, 59.5]	[0.55, 0.579]
50	250	14	10	[99.4, 100]	[5.38, 31.118]
50	250	14	20	[99.5, 99.9]	[10.68, 18.85]
50	250	14	30	[99.8, 100]	[13.2, 16.1]
100	800	5	10	[93.5, 97.8]	[39.04, 70.45]
100	800	5	20	[93.45, 97.1]	[47.4, 64.3]
100	800	5	30	[95.1, 96.2]	[52.3, 58.4]