Physician Scheduling During a Pandemic

Tobias Geibinger[1], Lucas Kletzander[1] Matthias Krainz[2], Florian Mischek[1], Nysret Musliu[1], and Felix Winter[1]

[1] Christian Doppler Laboratory for Artificial Intelligence and Optimization for Planning and Scheduling, DBAI, TU Wien, Vienna, Austria
{tgeibing,lkletzan,fmischek,musliu,winter}@dbai.tuwien.ac.at
[2] St. Anna Children's Hospital, Medical University of Vienna, Vienna, Austria
matthias.krainz@stanna.at

A Appendix

A.1 History

The history of previous assignments is captured by three additional input parameters as seen in Table 1. In order to provide continuity across planning horizon boundaries, the input contains the number of consecutive working days at the end of the previous schedule e_i, the last (non-common) assigned station s_i, and the last assigned shift t_i.

Table 1: Additional input parameters for history

History on consecutive working days	$c_i \in \mathbb{N} \quad \forall i \in P$
History on last assigned station	$s_i \in S^+ \quad \forall i \in P$
History on last assigned shift	$t_i \in T^+ \quad \forall i \in P$

Several constraints now need to be augmented with a second version covering the edge case:

For the maximum number of consecutive shifts (Equation (2)) e_i specifies the relevant history. Formally, at least one out of the first $7 - e_i$ days has to be assigned off. If e_i is larger than 6, the instance is not infeasible, but assigns a day off for the first day.

$$\bigvee_{k=1}^{\max\{7-c_i,1\}} (x_{i,k} = 0) \quad \forall i \in P \tag{1}$$

Forbidden shift sequences (Equation (3)) use t_i is used for the relevant history.

$$\neg(x_{i,1} \in Q_{(t_i)}) \quad \forall i \in P \tag{2}$$

Equation (10) needs to be extended to incorporate s_i.

$$\text{nvalue}(\{v_{(s_i)}\} \cup \{v_{(z_{i,j})} | j \in L\}) \leq 2 \quad \forall i \in P \tag{3}$$

For counting the number of station changes the initial value $ls_{i,0}$ is set to s_i (unless common).

$$ls_{i,0} = \begin{cases} 0 & \text{if } s_i \in C \\ s_i & \text{otherwise} \end{cases} \quad \forall i \in P \tag{4}$$

A.2 Redundant Constraints

We propose two groups of redundant constraints:

1. A set of redundant constraints that checks for each shift and skill requirement if enough employees have been assigned, whose associated preference is smaller than 4:

$$\sum_{i \in P} [x_{i,k} = t \wedge z_{i,k} = s \wedge p_{i,s,k} < 4] \geq d_{s,t,j,k} \tag{5}$$
$$\forall s \in S, t \in T, j \in L, k \in K \text{ where } t > 2 \vee s \in C$$
$$\sum_{i \in P} [(x_{i,k} = 1 \vee x_{i,k} = 2) \wedge z_{i,k} = s \wedge p_{i,s,k} < 4] \geq d_{s,1,j,k} \tag{6}$$
$$\forall s \in S \setminus C, j \in L, k \in K$$

2. A set of redundant constraints which ensures that the total shift, skill, and station assignments per day are equal to the associated cover requirements using global cardinality constraints (In the following we use square brackets for list comprehension and $\oplus$ for list concatenation):

$$cT_j = \left[|P| - \sum_{s \in S} \sum_{t \in T} \sum_{k \in K} (d_{s,t,j,k}) \right] \oplus \left[\sum_{s \in S} \sum_{k \in K} (d_{s,1,j,k}) - \sum_{u \in U} (ds_{u,j}) \right] \oplus \tag{7}$$
$$\left[\sum_{s \in S} \sum_{k \in K} (d_{s,2,j,k}) + \sum_{u \in U} (ds_{u,j}) \right] \oplus \left[\sum_{s \in S} \sum_{k \in K} (d_{s,t,j,k}) \,\middle|\, t \in T \setminus \{1,2\} \right]$$
$$\text{global\_cardinality}([x_{p,j} | p \in P], T^+, cT_j), \forall j \in L \tag{8}$$

$$cS_j = \left[|P| - \sum_{s \in S} \sum_{t \in T} \sum_{k \in K} (d_{s,t,j,k}) \right] \oplus \left[\sum_{t \in T} \sum_{k \in K} (d_{s,t,j,k}) \,\middle|\, s \in S \right] \tag{9}$$
$$\text{global\_cardinality}([z_{p,j} | p \in P], S^+, cS_j), \forall j \in L \tag{10}$$

$$cK_j = \left[|P| - \sum_{s \in S} \sum_{t \in T} \sum_{k \in K} (d_{s,t,j,k}) \right] \oplus \left[\sum_{s \in S} \sum_{t \in T} (d_{s,t,j,k}) \,\middle|\, k \in K \right] \tag{11}$$
$$\text{global\_cardinality}([y_{p,j} | p \in P], K^+, cK_j), \forall j \in L \tag{12}$$

A.3 Search strategies

We define three programmed search strategies that we use in our experiments:

- **default**: The solvers' default variable and value selection strategy.
- **search1**: For this search strategy we define an order of the decision variables such that we select on each day first all shift assignment variables for every physician before selecting all station assignments and finally selecting all skill assignments:

$$\begin{aligned} search1 \coloneqq\ & x_{1,1}, \dots, x_{|P|,1}, z_{1,1}, \dots, z_{|P|,1}, y_{1,1}, \dots, y_{|P|,1}, \\ & x_{1,2}, \dots, x_{|P|,2}, z_{1,2}, \dots, z_{|P|,2}, y_{1,2}, \dots, y_{|P|,2}, \\ & \dots, \\ & x_{|P|,|L|}, \dots, x_{|P|,|L|}, z_{|P|,|L|}, \dots, z_{|P|,|L|}, y_{|P|,|L|}, \dots, y_{|P|,|L|} \end{aligned}$$

  In general we use a smallest domain first variable selection heuristic together with a minimum value first value selection in **search1**. However, ties are broken by the order on the variables and the search assign groups of decision variables for a certain day and type in a consecutive manner (i.e. $x_{1,1}, \dots, x_{|P|,1}$ is completely assigned before the search moves on to $z_{1,1}, \dots, z_{|P|,1}$ and so on).
- **search2**: Select the shift, station, and skill assignment variables together for each physician one day after another:

$$\begin{aligned} search2 \coloneqq\ & x_{1,1}, z_{1,1}, y_{1,1}, x_{2,1}, z_{2,1}, y_{2,1}, \dots, x_{|P|,1}, z_{|P|,1}, y_{|P|,1}, \\ & x_{1,2}, z_{1,2}, y_{1,2}, \dots, x_{|P|,|L|}, z_{|P|,|L|}, y_{|P|,|L|} \end{aligned}$$

  In **search2** we again use a smallest domain first variable selection heuristic together with a minimum value first value selection where ties are broken by the order on the variables.

After all the decision variables have been assigned in **search1** and **search2**, auxiliary variables are assigned using the solvers default search strategy.