1 Variables

1.1 Demand Matrix

The demand matrix will have 3 dimensions:

- Time (24 rows per day, or 168 rows in total)
- Subject ($\sim$ 5-6 columns: English, Math, Accounting, CAD, Study Skills, etc.)
- Campus (6 slices: one for online and one for each of the campuses)

The entry $D_{j,k,l}$ represents the demand for time slot j, in subject k, at campus l, where each index ranges from 1 to $n_j (= 168), n_k, n_l (= 6)$ respectively.

1.2 Availability Matrix

The availability matrix will have 4 dimensions:

- Tutor
- Time
- Subject
- Campus

The entry $A_{i,j,k,l}$ represents the availability of tutor i at time slot j, in subject k, at campus l, where each index ranges from 1 to n_i, n_j, n_k, n_l respectively. Each entry is either 0 or 1.

1.3 Decision Variables

The variable $x_{i,j,k,l} \in \{0, 1\}$ indicates whether tutor i is assigned to time slot j for subject k at campus l.

2 Constraints

2.1 Availability

Tutors can only be booked for the time slots and subjects for which they are marked as available on the availability matrix.

For each i, j, k, l, if $A_{i,j,k,l} = 0$, then we add the constraint that $x_{i,j,k,l} = 0$.

2.2 Avoid Multiple Bookings at the Same Time

Since the variables for each subject and campus are now independent of each other, we must add the constraint that the same tutor cannot be booked for multiple subjects or at multiple campuses at the same time.

For each tutor i and time slot j, we add the constraint

$$\sum_{k=1}^{n_k} \sum_{l=1}^{n_l} x_{i,j,k,l} \leq 1$$

2.3 Avoid Consecutive Time Slots at Different Campuses

If a tutor is working at multiple campuses, each contiguous shift should take place at the same campus, so there should not be consecutive time slots for the same tutor at different campuses.

For all tutors i, time slots j, subjects k_1, k_2, and campuses l_1, l_2 such that $l_1 \neq l_2$, we add the constraint

$$x_{i,j,k_1,l_1} + x_{i,j+1,k_2,l_2} \leq 1.$$

(This disallows consecutive appointments at different campuses, but allows consecutive appointments for different subjects. If we wanted to disallow consecutive appointments for different subjects as well, we would require that $k_1 \neq k_2$ <u>or</u> $l_1 \neq l_2$.)

2.4 Online vs. In-person

The online schedule goes from 9 AM to 9 PM each day and is open on weekends, while the in-person schedule goes only from 10 AM to 5 PM each day and is only open on weekdays. Thus there should be a constraint for each in-person schedule that prevents tutors from working in non-operating hours at the in-person locations.

The time slots starting at 9:00 and 9:30 AM are the 1st and 2nd time slots of the day, and the time slots after 5:00 PM are the 17th, 18th, ..., 24th time slots of the day. Since our indexing for time slots starts at $j = 1$, the non-operating hours at in-person campuses correspond to when $((j-1) \mod 24) + 1 \in \{1, 2, 17, 18, \ldots, 24\}$, and the weekend hours correspond to when $\frac{j-1}{24} \geq 5$.

Assuming that $l = 1$ represents the online schedule, and $j = 1, 2$ represent the time slots starting at 9:00 and 9:30 AM while $j = 17, \ldots, 24$ represent the time slots starting at 5:00, ..., 8:30 PM, we add the constraints

$$x_{i,j,k,l} = 0$$

for each tutor i, time slot j such that $((j-1) \mod 24)+1 \in \{1, 2, 17, 18 \ldots, 24\}$ and $\frac{j-1}{24} \geq 5$, subject k, and campus $l \geq 2$.

2.5 Work Hours Constraints

We require that no tutor works for more than 24 hours per week, more than 7 hours a day, or more than 5 consecutive hours in the same day (summed over all campuses and subjects).

For each tutor i, we add the following constraints:

$$\sum_{j=1}^{n_j} \sum_{k=1}^{n_k} \sum_{l=1}^{n_l} x_{i,j,k,l} \leq 24$$

$$\sum_{k=1}^{n_k} \sum_{l=1}^{n_l} \sum_{t=1}^{24} x_{i,24d+t,k,l} \leq 7 \qquad \text{for each } d = 0, \ldots, 6$$

$$\sum_{k=1}^{n_k} \sum_{l=1}^{n_l} \sum_{r=0}^{5} x_{i,24d+t+r,k,l} \leq 5 \qquad \text{for each } d = 0, \ldots, 6,\ t = 1, \ldots, 24$$

2.6 Bounds

For each i, j, k, l, we require that $0 \leq x_{i,j,k,l} \leq 1$ and $x_{i,j,k,l}$ is an integer.

3 Objective function

We want to minimize the squared difference between the number of tutors working at each time slot, subject, and campus, and the demand for that time slot, subject, and campus.

To condense this into one objective function, we can take the squared difference between the number of tutors working at each tuple (j, k, l), and sum over all j, k, and l:

$$\text{minimize} \sum_{j=1}^{n_j} \sum_{k=1}^{n_k} \sum_{l=1}^{n_l} \left(D_{j,k,l} - \sum_{i=1}^{n_i} x_{i,j,k,l} \right)^2$$