

Screen Name: KidButtowski

Name: Anderson Addo

CS Login: aaddo

Variables and Constraints Used

The four main decision variables (or rather, structures) were 2D arrays for employee start times (startTimes), end times (endTimes), shift codes (like off work shift has a code of 0) (shiftCodes) and hours worked (hoursWorked). Each of these 2D arrays were of size [numEmployees][numDays].

For the constraints:

- We had the values in the shift codes array imply the ranges for the values for entries in the start and end time arrays.
- If a shift code was not 0, we also applied a constraint that for any employee e and any day d , the value of the difference between $\text{endTimes}[e][d]$ and $\text{startTimes}[e][d]$ should be between the min and max working hours.
 - The values of these differences were used to fill the hours worked array.
- I used a count constraint to make sure that for any day and any shift code, the number of shifts with that shift code was greater than or equal to the value in $\text{minDemandDayShift}[\text{day}][\text{shiftcode}]$. This required populating an new array with the existing IloIntVars that corresponded to just the shift codes used in that day (since the 2D arrays are indexed by employee, not day)
- I used a similar approach (i.e. of making a sub array of existing vars) to ensure that, for any day, the sum of all hours worked for any particular day was greater than or equal to minDailyOperation .
- I used allDiff to ensure that, for any employee, $\text{shiftCodes}[\text{employee}][1 \text{ to } 4]$ were all different.
- For every employee (for every week), we made a new array of existing variables $\text{hoursWorked}[\text{employee}][\text{weekoffset} + (0 \text{ to } 6)]$ and ensured that the sum of these was within the range of 20 to 40.
- For making sure that there were no consecutive night shifts, it would have been nice to use a [sliding window counting constraint](#). I think constraint is a pretty large bottleneck as it stands; for every entry of the shiftCodes array, we add a constraint such that an entry $\text{shiftCodes}[\text{employee}][\text{day}] = 1$ implies $\text{shiftCodes}[\text{employee}][\text{day} + 1] \neq 1$
- Lastly, for every row in the shiftCodes array, we check that the count of entries with value 1 is less than or equal to $\text{maxTotalNightShift}$

Possible Improvements

One thing that I noticed is that any individual employee's schedule is pretty haphazard. They jump from evening shifts to say shifts to off days with no order; that would be very disorienting in a real world schedule.