CSCI2951O
Project 2: Employee Scheduling Report
Names: Colin Savage and Shreyas Mishra
CSlogins: csavage4 and smishr22 Screenname: avj & thinker

Hours worked: 20 hours combined

Model variables and constraints

For our model, we just used the constraints described in class. We originally had another approach that involved 4 or 5 matrices to store decision variables (see our old submission from 3/3) but we went with this new approach and achieved much faster runtimes.

Search Guidance

We started by considering the shift matrix (which maps each employee to their corresponding shifts) and the length matrix (which maps each employee to the hours worked in the shift). For each of these two matrices, we set up two phases where we were choosing the minimum value first then the maximum for assignment. However, this approach turned out to be too slow as the domain size was too large here and each subsequent assignment wasn't making a huge difference. Since we were effectively just choosing random values in a large matrix (potentially >1000 values) there was no structure to this approach, and mistakes at the beginning could potentially be found at the end, leaving a long backtrack to go.

If we were to do this task by hand, the first thing that we would aim for would be the variable with the minimum domain (usually this would be the variable with the maximum number of constraints), as this aligns with the fail first strategy discussed in class. For this reason, we began with the shift matrix. We tried multiple approaches, including splitting this by week, by day, doing a week in the shift matrix, then a week in the length matrix, (same for day), before we settled on a happy medium. This ended up being a set of 2n phases (n=number of days), where we dedicated 14 phases to each week. We would do the shift matrix entries for each day, followed by the length matrix entries for this day. This way, we added additional structure to our search by handling the daily constraints immediately, while also handling the weekly constraints in an equally reasonable amount of time as our other approaches. Since a satisfying assignment of the lengths once we have shifts (only constraint being # daily operation and min and max hours per employee) is essentially given (barring too few shifts for an employee) it was clear that we should just handle this matrix after.

Despite this, we were still lacking in a bunch of test cases, with some not concluding in time. What we did to change this was to change the value chooser for the shifts from

minValue to maxValue, which made everything work almost instantly. Before, we were assigning the fewest possible employees to work, which meant that there was potential for employees to be assigned 0 shifts a week and not meet minWeeklyHours. Similarly, minDailyOperation could not be met. Since both of these concern the lengthMatrix, we'd have to backtrack a ton back to the shift assignments. This all changed with the max value chooser. Though this created a significant bias towards employees being assigned the evening shift (value=3), and working every day, we were always able to find a satisfying assignment for the lengths, since even if every employee is working every day, we could set each shift to be 4 hours, and we would be all set (though this could change with other values of minWeeklyHours and maxWeeklyHours not 20 and 40). Similarly, minDailyOperation could almost trivially be assigned.

One additional thing to consider in a model such as this, is that an individual employee may have specific constraints. These could be specific times they can or cannot work, preferences for different shifts, or a certain hours quota or cap (some employees may just be part-time vs full-time).

A possibility that the Professor discussed in class was that the search guidance that we come up with might be tailored specifically for our input cases, and we believe that might be the case here. We got much faster runtimes when using the maxValue vs. the minValue for the input cases we have, though this could be very different if we had more stringent caps on hours or number of shifts per employee per week, which we don't have. A pitfall of the search guidance mentioned above is the type of value selector - using the minimum value chooser makes the model biased towards giving employees more morning shifts, and using the maximum value chooser does the same by giving more evening shifts. A consequence of this biased allocation with the minimum value selector is that the model will force most employees to work the least permissible number of hours (4 hrs in this case) during the morning shift but then force other employees to work a much longer shift, which might not even be possible due to the maximum workhours limit or the daily shift constraint (only 1 shift per day).