Christian Armstrong carmstr8, dcho24
Daniel Cho Ivanka, T2

Project – II

Model, Constraints, Variables

We use the model suggested in class with decision variables as shifts and hours as opposed to a start and end times. Specifically, our internal representation is a list of days, where each day holds an inner list with tuples of shift and hours in the shift per employee.

Strategies

We experimented with various techniques but found the simplest to solve the most cases. In the following experiments, we ran the solver for 60s each as generally if the solver did not solve the constraint relatively quickly, it often would not solve the problem within 300s.

1. As a baseline, giving the solver the flattened list of variables solved 7 cases.
2. Selecting the min index and then max value for the variable solved 10 cases.
3. Selecting the min index and then smallest value solved 3 cases.
4. Selecting the max variable with either max or min solved 0 cases.
5. We then tried various phases.
   (a) Splitting into shifts and hours solved 7 cases.
   (b) Splitting into training, then the rest of the variables solved 7 cases. We assumed that solving for the most constrained section and then the rest would perform well, but this was not the case (possibly due to erroneous use of the cp solver?).
   (c) Splitting into training, then monthly solved 0 cases. This was most likely due to incorrect use of the solver.
6. We tried random restarts, increasing the fail limit and incrementing the seed, which did not improve results.

In summary, we hypothesized that splitting the problem into the training days and then monthly phases would greatly help the solver. However, this was not the case in our experiments, where breaking the problem into phases only hurt performance.

On the other hand, selecting the min index and max value performed the best. This may be because it does something similar where it first solves the training days, then sequentially solves for the days. Moreover, it appears that solving for the max shift/number of hours also helps. Maybe it shouldn't prioritize evening shifts (probably day shifts), but assigning the max number of hours performs better than starting the solver with the least.

Schedule Analysis

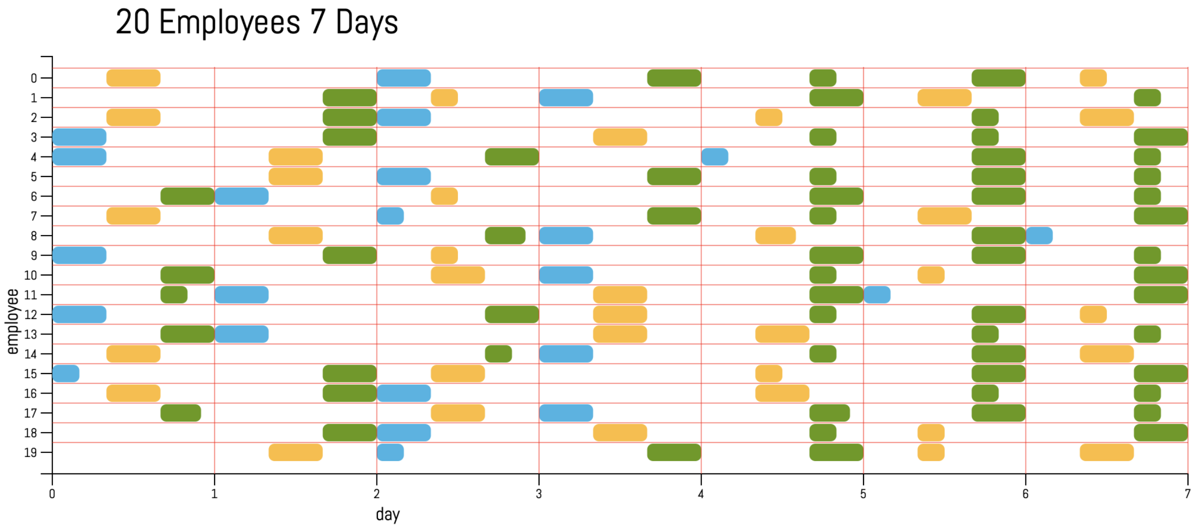

Figure 1: Visualized Schedule

Here we've sampled a schedule for a week and 20 employees created by our solver. This schedule has a few issues.

1. The first is that the solver prefers the evening shift. This is most likely due to our formulation, though. Because the evening shift has a value 2 and the solver initially chooses the max value, it generally tries to assign people the evening shift. We could fix this by switching the values of indication.

2. Another issue is that the shifts can vary quite a bit depending on the person. In a real schedule, we try keep shifts consistent. For example an employee designated a night shift employee should continually get the night shift and not a day shift.

3. Lastly, there is no constraint on consecutive shifts between days. For example, an employee could work the night shift and then the day shift, which could be a consecutive 16 hours of work in the perspective of the employee.

To solve these issues, we might either add extra constraints or encourage the solver to prefer some assignments. By preference we mean that the solver initially chooses some value, which would be preferred, but is not a hard constraint.

Time Spent: 5 hours