Journey

In this project I have tried my best to implement a constrain programming solution for the employee scheduling problem. For the model I have used the constraints we spoke about in class. I started by initializing the two main variables for the problem that were in form of matricies - the shifts and duration ones namely. They are indexed by employee and day. I also specified their domains. Shifts could take 0,1,2,3 and durations could take 0,4,5,6,7,8. I then implemented the constraints we spoke about. I did them sequentially using nested for loops to implement them for each variable in both the matrices. After implementing the constraints, I ran the ibm solver on them and it solved all the instances at an astonishing speed. I then created the function to extract the schedules in the correct output format from these values that the solver returned. I then spent the next 4 hours trying to understand these schedules. I first needed a way to check the validity of the schedules because I felt that the first and most essential way to evalauate the quality of the schedules was to ensure they are sound. To do this I created 7 functions - each of which checked the validity of the constraints mentioned in the project handout. I first checked the constraint where is the shift is off then the duration must be zero and vice versa. I immediately found my first bug. I had mistakenly used the number 1 to represent the off shits and the number 0 to represent night shifts in all my constraints. I quickly updated that and then proceeded to check the validity of all the other constraints. Through this process, I found myself changing my constraints around due to both silly mistakes on my part, but also not really understanding the constraints entirely. Once I made sure all the constraints worked, I moved on to to qualitatively evaluating the constraints.

Analysis of Schedules

I am not going to lie. I was somewhat confused by this part initally. It also took me a while to understand what these pretty print graphs represented. I decided to try and see what happens from a business perspective where we try to increase output. Lets say we want the min number of hours to go up per day. I did the analysis with the 7-14 schedule to make drawing inferences a little easier. I ran it initially and got a sound result that satisfied all the constraints. I ran it again with some changes that would make production improve. The following are the changes made:

minDailyOperation 80

minWeeklyWork 25

MaxTotalNightShift 3

I initially thought that there might be some UNSAT induced into these constraints that would result in the solver not being able to solve it, but I was surprised that it gave me an output promptly. Also thinking about these constraints, they don't look so bad. Increasing min daily operations with only such small tweaks could be really beneficial from the employers point of view. Min weekly work has only gone up by 5 hours which is not a lot. In fact, I would go out on a limb and say the number can be as high as 30. Working anywhere between 30-40 hours a week still gives you a pretty good work life balance. There was a noticeable increase in the number of night shifts, but if you have a night shift you dont have to work for the rest of the day and you get your 16 hours back of which you can sleep for 8 and do misc activities for the remainder. If there was a constraint that said if after a night shift you have to take an off shift, this really doesn't seem so bad. And I

implemented that constraint and we also get good results that are sound. Furthermore if thats the case then you can have more consecutive night shifts as well, because you will be rewarded with a day to set your schedule back to normal. I also find the min demand day shift constraint kind of obscure. If we are already constraining the number of hours of work we need why do we need to fix a split of those hours between shifts. Even if we need to get work done sequentially I am sure most of the factories these days have ways of ensuring these things can happen either through robots or remote access to these machines. I ran the solver without this constraint as well and it provided an output that passed all my constraint satisfiability checkers.

Search Guidance

This was an absolutely difficult challenge for me. I was stuck several times and wanted to just give up. I tried so many different things and in the end was simply not able to catch all instances. I tried over 6 different search paradigms. Initially with just domain sizes and value selection. Then I tried with phases. I had an initial split between shifts and durations. Then added a split between training and remaining days. Then tried day by day. Then I added durations and shifts together but spliting only on training days and non training days. I tried a custom value selection for duration in the order of 4, 8, 0, 5, 6, 7. None of the above worked well at all. I then tried with randomized restarts and that gave me some progress. Still nothing to write home about. I finally used the impact functions and that's when things got a little better. I still wasn't happy given that I spent two full days in the SunLab especially on the weekend of St. Patricks Day and still didn't get any result to feel proud about, but I think at this point I can't justify spending more time on this. I would truly love some feedback on how I could maybe improve the search paradigm to catch more instances and it doesn feel a little UNSATisfying (Pun intended) but we move.

Time

I would have spent over 32 hours on this project. Almost all of it on the search guidance part.