One of the biggest challenges I ended up facing was the tools I was using. I had to recreate the project multiple times to get it to run, and I was unable to implement the unit testing I was hoping for. I had a few issues with the scoring system, but no more debugging than usual.

The schedule below was the best schedule I’ve produced so far, and I think it looks pretty good based on the scoring mechanism. SLA191A and SLA191B are closer than 4 hours apart, but I believe it’s so that Banks can teach all of them. Otherwise, I’m fairly happy with what it came up with.

A screenshot of a computer

Description automatically generated

I did notice that sometimes the program gets stuck at a certain score, even with mutations, probably just because of the facilitators it has chosen. Since it’s in the program’s best interest to have facilitators chosen as 4-4-3, it probably quickly converges to a specific set of facilitators. Then, if a mutation changes the facilitator, it will quickly be bred out since there is now a facilitator with 1 or 5 classes. This would take a lot of effort to avoid, but I found it interesting.

I took some liberties with the fitness function just to improve the overall scoring (I marked these in the code with “CHANGED” comments). The following are the changes:

* 101 and 191 classes should be 4 or more hours apart (originally more than 4)
* Double-scheduling a facilitator reduces score by 0.5 (originally 0.2)
* Dr. Tyler should only have 2 or fewer classes (originally up to 4 like everyone else)
  + This was more for Tyler’s sake than the performance of the algorithm
* The simulation starts with 1000 schedules (originally 500)
* The mutation chance is 0.1 (originally 0.01)
  + This is to fight against minor changes being quickly bred out, like changing facilitators
* The minimum number of generations is 200 (originally 100)
* The selection process from the breeding pool is to breed the best and worst parents, working inwards
  + Not positive if this increases the performance, but I tried a couple methods.

Overall, I’m happy with how this turned out. While the schedules aren’t always perfect, they are always valid and score close to perfect (I calculated somewhere in the 20’s as being a perfect score). I probably should have given myself more time to do the project, but instead crammed in all the work in two days. I’m very tired now.