**Program 2 - Genetic Algorithms Report for CS461**

Trinity McCann

**i) What were the challenges in writing the program? Or did it seem to go smoothly from the beginning?**

This program was definitely outside of my comfort zone and familiarity with coding. I am very comfortable with single class projects in Java, such as scripting, which this project does kind of fall into that category. However, in terms of Genetic Algorithms, I have never actually coded one before. I did have the assistance of Claude, the AI tool, for my GeneticScheduler class (which was used to set up POPULATION\_SIZE, NUM\_GENERATIONS, and MUTATION\_RATE), the mutate method, and the tournamentSelection method as those relied heavily on genetic algorithm knowledge. After that I was able to set up the main method and calculateFitness methods fairly easily. Main was a simple for loop comparing and contrasting the data given from calculateFitness and GeneticScheduler. And calculateFitness was simple boolean checks into basic addition or subtraction. Finally came storing, parsing, and printing data which was all relatively easy, mostly just data entry of room names, names, times, and class sizes from the instructions. I definitely encountered some struggles, but StackOverflow and Claude kind of made those a non factor in my issues. This task seemed a lot more daunting when I first looked it over, but it kind of came together into one piece by the time I started working on it. I think this is due to the minimal requirements that did not have the actual restrictions a full on scheduling system would need (Such as when professors are available, multiple companies vying for the same time/space, extended hours, specific exceptions, and many more).

**ii) What do you think of the schedule your program produced? Does it have anything that still looks odd or out of place?**

I am extremely happy with what my schedule produced! The program seems to cap out at a maximum fitness score of 8.49 repeating. I believe this is the highest max efficiency you can get, but I would be really interested if anyone else in the class got a higher number. I double checked some of the 8.49 fitness schedules and they all seem accurate (As in, no classroom is being used at the same time, no professor is double booked, and there are no class sizes too small). I also noticed that room Logos 325 is never used, which makes sense as it is much too large for all of the classes that need it. Simple checks like this really make me confident in my program's output. Once you hit about 300 population size, you get the 8.49 fitness score regularly. By the time you hit 500, it has not failed to hit maximum fitness.

**iii) How would you improve the program, or change the fitness function?**

I think the point system that the fitness function uses is a really good start. Honestly, I would just keep using the fitness point system even if there are better options because it is a quick computational way to get a general idea of a good schedule. As for the program, I think there’s many more things that could be done to greatly improve this. For instance, you could include classes that are too far away from each other and cannot be reached in a 10 minute walk, you could include professors that are only available at certain times or days, you could have classes that require two professors, you could have 2 or 3 hour long classes. I think there is a lot of room for improvement in a scheduling system like this.

**iv) Anything else you feel like discussing, asking about, bragging about, etc.**

Not particularly, I am feeling very excited at taking my first broach into AI and genetic algorithms. I always thought coding was daunting and scary but the more and more I get into coding, the more I feel like all of these abstract, complex topics are actually achievable with time and dedication.