Isabella Rocha

My algorithm for the Traveling Salesman Problem was very similar to the genetic algorithm explained in the video. For each set of points, I went through 1000 generations of a population of 150. The first 150 paths were just randomly shuffled routes, then mating of two paths worked similarly to the video, with a portion of the first path making up a random subset of the new path and then using the second path to fill in the rest. Each path's fitness was calculated based on its length and two paths with higher fitnesses had a higher chance of mating in hopes of getting fitter and fitter generations. If that generation produces a better path than the last generation, that path is stored and after all 1000 generations have been run through, the shortest path found is returned. One thing that I did not incorporate from the video, however, was mutations. I initially included mutations, but even when I had the mutation probability down to 1%, it seemed to do more harm than good, with the length of A50 being shorter with no mutations than with any mutation probability I tried.

For better results, I could raise the generations and population, however, this would take significantly longer to run. I tried raising the population and generations for A50 and while I did get better results for A50, I felt I should keep my initial results as the time it took to run with increased generations and population was much too long.