Assignment 3: TSP with Genetic Algorithms

Abhijeet Prajapati

November 13, 2021

1 Introduction

The traveling sales man problem is a famous mathematical optimization problem, that is being tackled here with a genetic algorithm in this assignment. In the problem, the sales man must travel to every given city, only once and back to the original city, with the goal of finding the shortest travel route.

2 Background

2.1 Algorithm

The algorithm used in this problem will take a random order of the cities of a given population, then run the order through for some number of generations. In each generation i=1...n where n is the generation length, 1 order in the population will have its fitness value evaluated and based on that evaluation, it will be chosen to pass on it's order or not, possibly in a mutated order. Repeat until shortest path is found or end of generations.

```
BEGIN
i=0                         set generation number to zero
initpopulation P(0)         INITIALIZE usually random population
                            of individuals representing candidate solution
evaluate P(0)               EVALUATE fitness of all initial individuals of population
 while (not done) do test for termination criterion (time, fitness, etc)
 begin
     i = i + 1              increase the generation number
     select P(i) from       SELECT a sub-population for offspring
     P(i-1)                 reproduction
     recombine P(i)         RECOMBINE the genes of pairs of selected parents
     mutate P(i)            MUTATED perturb the mated population (i.e., resulting
                            offspring)
     evaluate P(i)          EVALUATE the new individuals
end
END
```

Figure 1 Basic general GA

B.Ombuki-Berman cosc 3p71

Figure 1: This is the pseudo code of a basic Genetic Algorithm.

[]

3 Experimental Setup

My experimental parameters were mostly set, they only changed for the 5 crossover and mutation options I was supposed to use. There were 2 options for how the crossover would take place, either through Uniform Order crossover or a Two-point-Crossover. But to keep my results easily replicable, I only changed the Crossover Rate and Mutation Rate as needed. For the last run where the parameters where up to me, I selected to have a high Crossover rate of 90% and also a high mutation rate of 90%. The high Crossover rate is useful so that most of the population is selected to reproduce, and a high Mutation rate was selected so that the algorithm would cover most of the solution space. Results were not as expected but I learned that I should have the Mutation rate to change dynamically, start at a high % to still cover as much of the solution space and then decrease to a very low amount and stay there to prevent too much change/decrease the fitness.

4 Results

4.1 Summary of results

Based on my results, I learned that runs that have any amount of a mutation rate, slowly increase in fitness compared to runs that have a 0% mutation rate. Runs that have no mutation rate very quickly rise in fitness, but also plateau out very quickly compared to runs with a mutation rate which slowly increase and have a lot more room to grow/improve the fitness value.

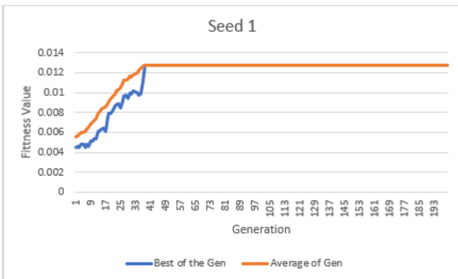

Figure 2: This is a line graph of the fitness of a run with High crossover rate and No Mutation rate.

4.2 Review

As seen, the run with a High crossover rate and any amount of a mutation rate, they slowly increase in fitness (figure 3). But compared to a run where both the Crossover rate and the Mutation rate was high, the run stays/hovers around the same fitness value for a long time (figure 4).

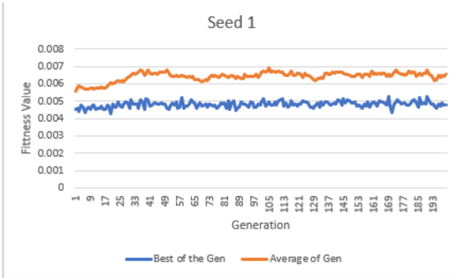

Figure 3: This is a line graph of the fitness of a run with High crossover rate and a low Mutation rate.

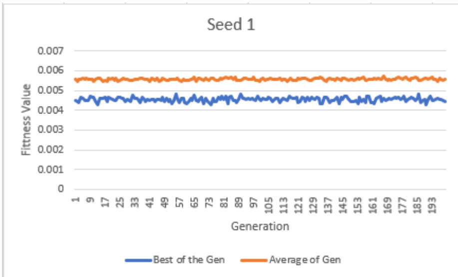

Figure 4: This is a line graph of the fitness of a run with High crossover rate and a High Mutation rate.

5 Conclusion

The tests I performed are ones with a high Crossover Rate and no mutation rate, high Crossover Rate and low mutation rate, and high Crossover Rate and high mutation rate. The results conducted of a high fitness growth that quickly plateaued, a fitness rate that slowly increased and did eventually result in a better route than the no mutation rate, but took much longer, and the last run resulted in a fitness rate that didn't improve much at all.

5.1 Afterthought

So, the run that performed the best and presented the fastest and reasonable distance answer were with high crossover and no mutation, but the run that should give the best route, at a much longer run time then the others, is the one with high crossover and low mutation rate. But the one that should give an even better result is the run that has a dynamic mutation rate that decreases over time to cover a wide solution space.

6 Sources

1. COSC 3P71 Powerpoint slides by Professor Beatrice Ombuki-Berman