

**Mazunki Hoksas**

**2022-03-02**

**UiO — IN3050**

For all algorithms, I've run the simulations ten times, and I've simulated the average time for 10 cities, and 24 cities on both Hill Climbing Algorithm and the Genetic Algorithm.

## **Exhaustive search**

For 10 cities: Called function 10 times. Took 10.53s on average!

Due to the nature of factorial growth of permutations, we can expect this number to grow by a factor of  $O(n!)$ . That means I'd expect this to take  $(24!/10!) * 10.53s$  for all 24 cities. Sufficient to say,  $10^{10}$  years is a bit too long for my own taste.

## **Hill climbing**

For 10 cities Called function 10 times. Took 0.00s on average!

For 24 cities Called function 10 times. Took 0.04s on average!

I was quite surprised by the effectiveness of this algorithm. Sadly it involves the issue of local maxima, resulting in imperfect runs.

Regardless, this is not a bad result: This generation ended at  $td=11523.65$ :

## **Genetic Algorithm**

For 10 cities Called function 10 times. Took 11.79s on average!

For 24 cities Called function 10 times. Took 19.64s on average!

I am not sure whether I've managed to solve this task successfully. The results I've gathered seem to be generally worse than those found by the hill climbing algorithm, and altogether being slower...

This algorithm depends a lot on the parameters selected, and does not really care much about the size of the input data... other than the fact that parameters need to be tuned properly.

The number of path checks done, and the runtime in general, of the genetic algorithms depends on the parameters we choose, and the luck we have while selecting random offsprings.

## **In general**

Running the algorithm shows more detailed information of the simulation, such as scores per generation. Furthermore, the source code should be well documented, and changing the input variables for the `Salesman()` should allow for easy testing.

I would like to know how to customize my parameters more accurately, and make sure I have implemented the breeding of two parents properly. I'm not sure I fully understand how it works.