Christopher Goes

Assignment 3 – Evolution Strategies

Algorithm

My algorithm uses the PLUS operator. Children are selected from the previous population, mutated, then the new population is selected from both the children and the previous population. Selection is a simple tournament with 3 contestants. Fitness is the Euclidian distance between all points in a chromosome. Mutation is done by incrementing the theta and radius of points in the chromosome using a random normal distribution, controlled by a sigma, with each point having a chance of 1/N to be mutated. This process happens for k generations. Then, I select the best individual in the population, and do a simple local search using the same mutation function.

Issues

For some time, my fitness evaluations would all converge to zero, which was quite strange. This was due to checking points against themselves, which of course is zero, thus making the fitness zero after the first evaluation.

Future expansion

The COMMA operator is partially implemented but buggy, so I'm planning to try using this once fixed. Another future expansion of this would be to evolve variables such a mutation probability and population size a sort of meta-evolution to generate an optimal algorithm. The codebase is architected such that this would not be difficult, and easy to parallelize onto a computational cluster (such as Amazon AWS or BigSTEM).