Evolutionary Computation: Theory and Application

Assignment - 02

Travelling Salesman Problem

Parameter values used for computation and analysis:

number of cities: 100
Population size in each generation: 200
mutation rate: 0.01
crossover rate 0.9
elitism: Best fitness
number of generations: 1500

Randomly initialized population

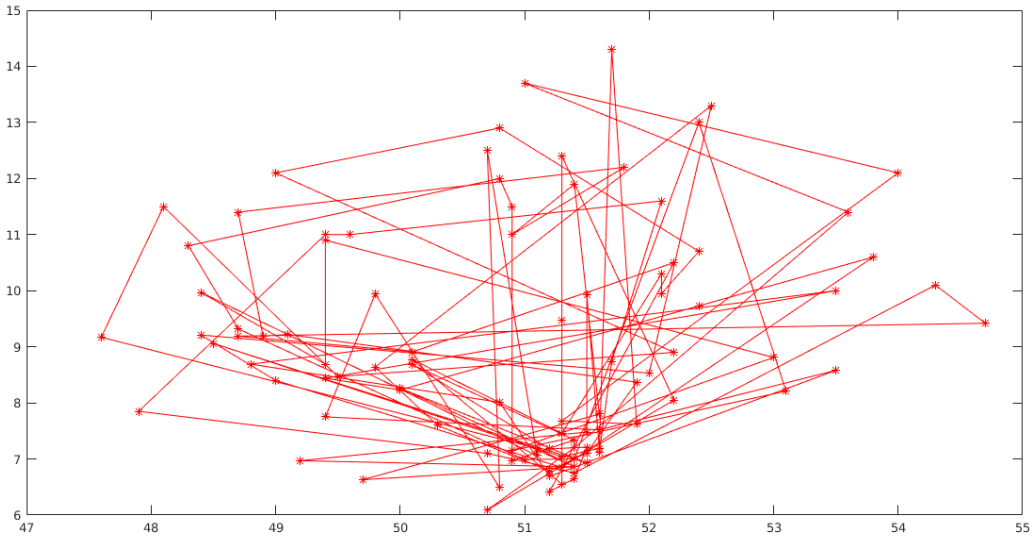

Figure 1: Edge map of individual with the best fitness from the initialized population

Best fitness after 1500 generations

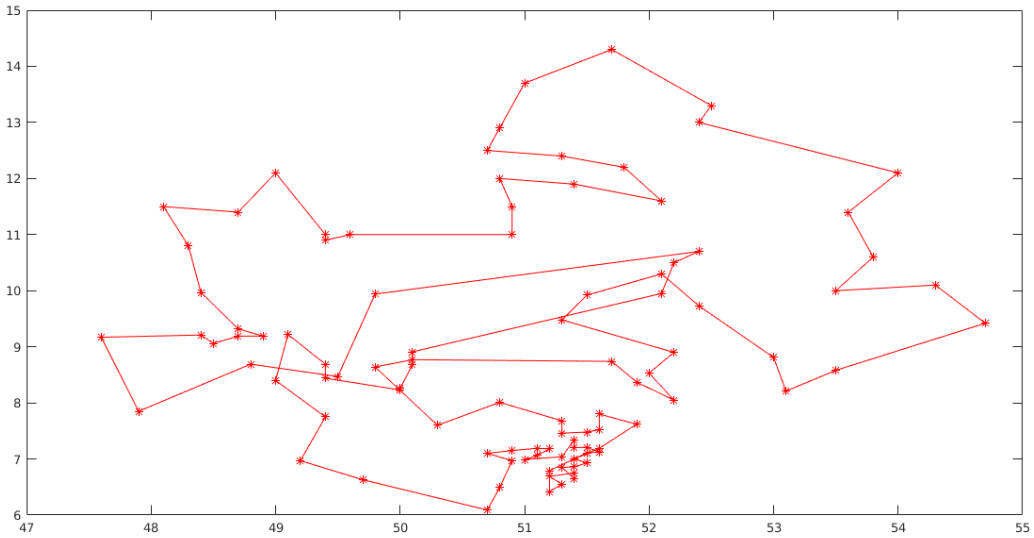

Figure 2: Edge map of individual with the best fitness after 1500 generations, using random crossover and neighbor swapping mutation.

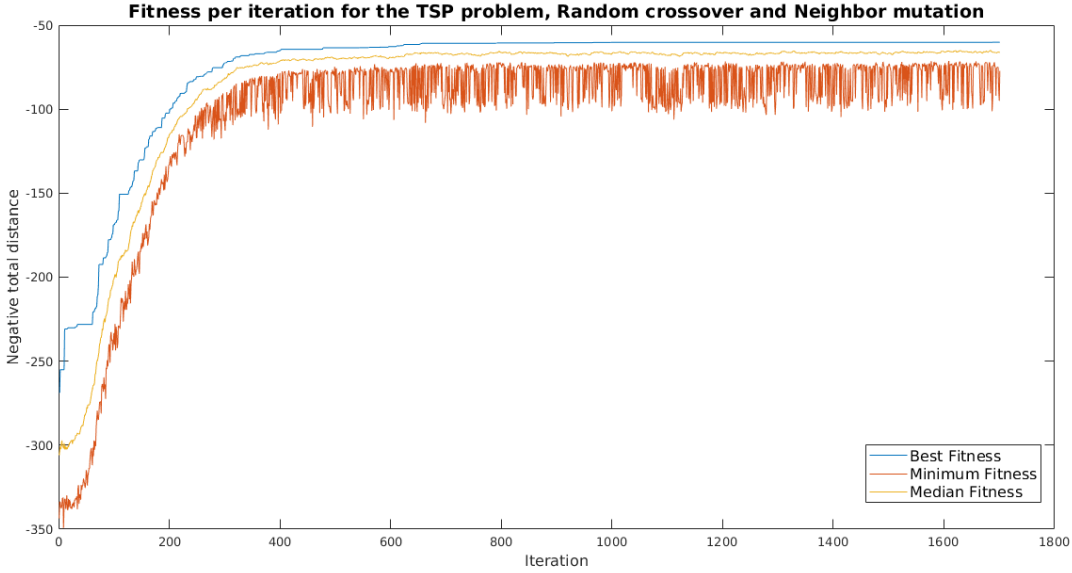

Figure 3: Evolution of fitness in each generation. We see that all best, mean and minimum fitness increases gradually over generations before we converge to a solution. This means that we have a reasonably well-conditioned mutation rate for the problem. The algorithm looks for maxima on the fitness landscape, thus our distance values have negative sign.

Experimentation with mutation and crossover mechanisms

number of cities: 100
Population size in each generation: 200
elitism is present
number of generations: 800
crossover rate: 0.9
mutation rate: 0.01

Experimentation Description

Setup

- Two mutation and two crossover methods are implemented
- Each combination of mutation and method is run 30 times
- The best fitness of the population is recorded for each iteration, and the median over 30 runs is recorded.
- The child with the best fitness of 30 runs is also recorded for each combination of crossover and mutation method.
- The same initialization is used for each combination of techniques.

Mutation methods

Each gene is chosen to mutate or not using the mutation rate as the probability.

- Order change: if gene mutate, it will swap place with a randomly selected gene in the genome.
- Neighbor swapping: if gene mutate, it will swap place with the adjacent gene (wrap around).

Crossover methods

- Cycle crossover: implement the algorithm as described on the tutorial from rubicite.com[1]
- Random crossover: a set number of genes are chosen randomly from first parent, then the rest are picked from the second parent so that all genes has an unique value (all cities are traveled only once).

Mapping of best solution for

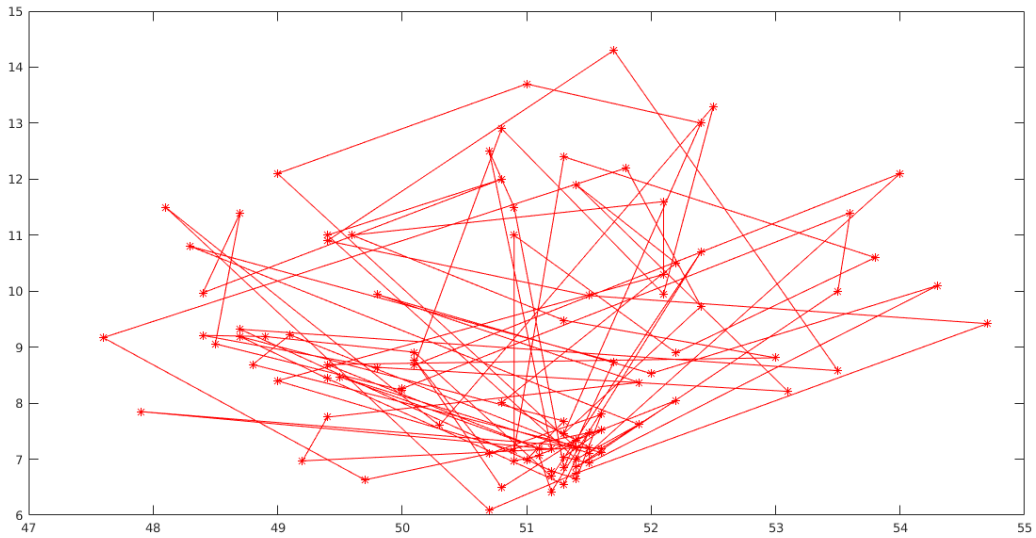

Figure 4: Edge map of individual with the best fitness from the initialized population

[1] http://www.rubicite.com/Tutorials/GeneticAlgorithms/CrossoverOperators/CycleCrossoverOperator.aspx

Best fitness after 800 generations

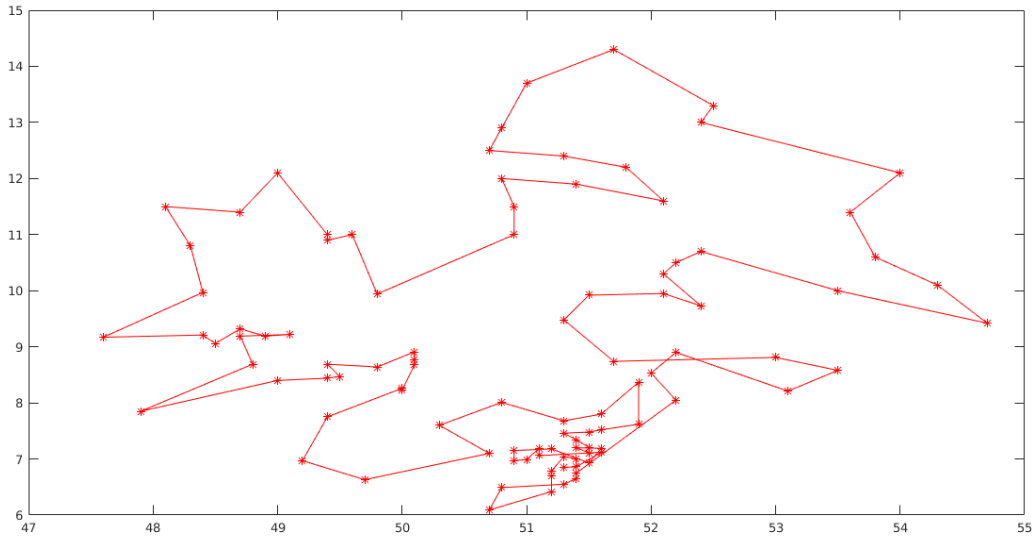

Figure 5: Best solution from 30 runs of each combinations for 800 generations each. The solution was acquired using a combination of random crossover and neighbor swapping mutation. The total distance for this solution is 57.9948.

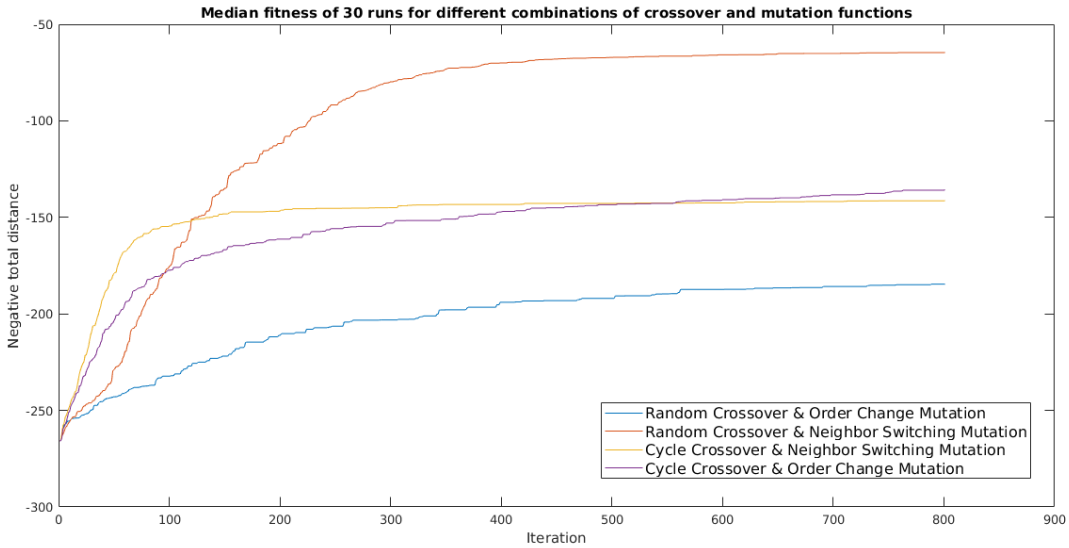

Figure 6: Evolution of median fitness in each generation over 30 runs. We tried different combinations of crossover and mutations as explained in the figure. We observe that random crossover and mutation by swapping neighbors gives us the best evolution in the fitness. The algorithm looks for maxima on the fitness landscape, thus our distance values have negative sign.