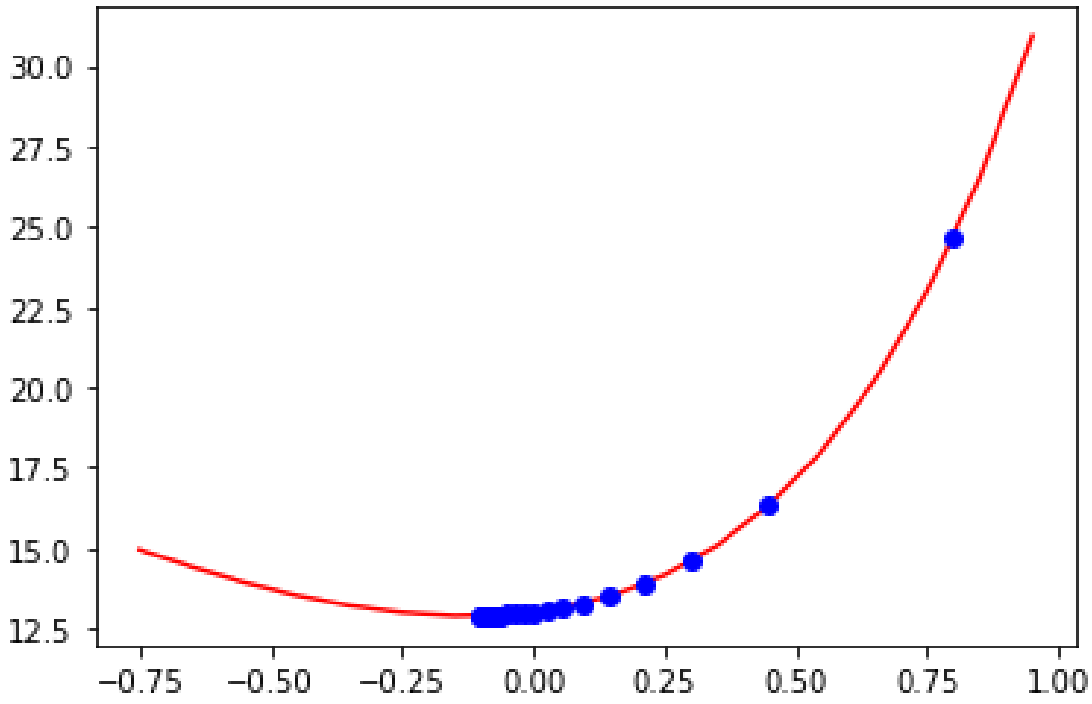
MECS4510 Evolutionary Computation, Fall 2023
Problem Set #1

Bole (James) Pan - bp2632@columbia.edu

September 17, 2023

- UNI: bp2632
- Instructor: Prof. Hod Lipson
- Grace hours used: 0
- Grace hours remaining: 96

Performance Plot



Column1	Column2	Column3	Column4
Row1C1	Row1C2	Row1C3	Row1C4
Row2C1	Row2C2	Row2C3	Row2C4
Row3C1	Row3C2	Row3C3	Row3C4
Row4C1	Row4C2	Row4C3	Row4C4
Row5C1	Row5C2	Row5C3	Row5C4

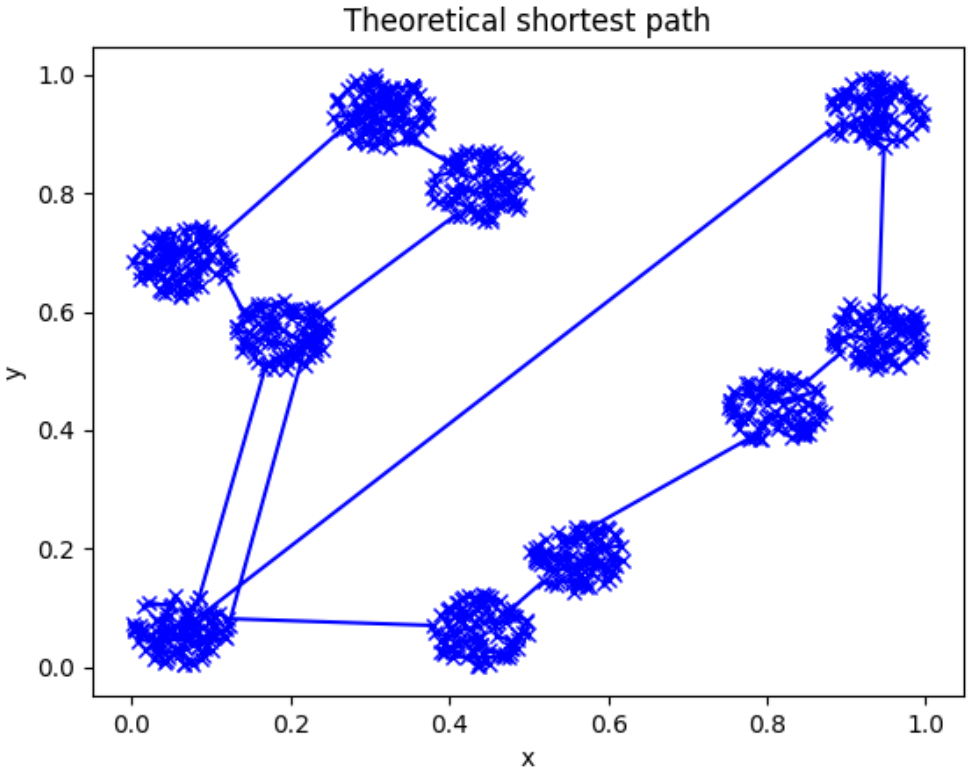
Table 1: Sample table with 4 columns and 5 rows.

Column1	Column2	Column3	Column4
Row1C1	Row1C2	Row1C3	Row1C4
Row2C1	Row2C2	Row2C3	Row2C4
Row3C1	Row3C2	Row3C3	Row3C4
Row4C1	Row4C2	Row4C3	Row4C4
Row5C1	Row5C2	Row5C3	Row5C4

Table 2: Sample table with 4 columns and 5 rows.

General

- Summary table of results:
- **Theoretical shortest path**: 12.775068874879208, found using Christofides algorithm.



Methods

- **Representation Used:** We represent each city as a tuple with two elements. A route (individual) is an ordered array of all the cities. A population is an array of routes (individuals). We are using direct representation, and more specifically, index representation.
  - **Mutation**: We use the swap mutation method. For an individual route, we loop through each city, and with a certain mutation rate we probablistically swap this city with another random city.
  - **Crossover**: We use the order crossover method. For two parent routes, we randomly select a subset of cities from one parent, and make the child's corresponding segment the same as that parents'. then fill in the rest of the cities in the order of the other parent.
- **Random Search**: Randomly generate a permutation of the cities and calculate the total distance. Repeat this process for a number of times and record the best result.
- **Random Mutation Hill Climbing**: Randomly generate a permutation of the cities and calculate the total distance. Then, for a number of times, apply mutation on the route and calculate the fitness of that route. If the new fitness is higher, keep the new route. Repeat this process for a number of times and record the best result.
- **EA variation and selection methods used**: We use the roulette selection method. From the initial population, we first select a small population of elites. The higher the fitness, the more likely an individual would be selected for this group. We put all our elites in the new population. Then, until the new population reaches the predetermined populationwe size, we randomly select two individuals from the elites and apply crossover to produce a child, apply mutation on the child, and put it into the new population. This process, going from the initia population to the new population, is repeated for a number of times (called generations).
- **Analysis of Performance**:
- **Methods compared**: we compare the performance of random search, random mutation hill climbing, and EA.