Report Assignment 3.1
Yazeed Alsuboh

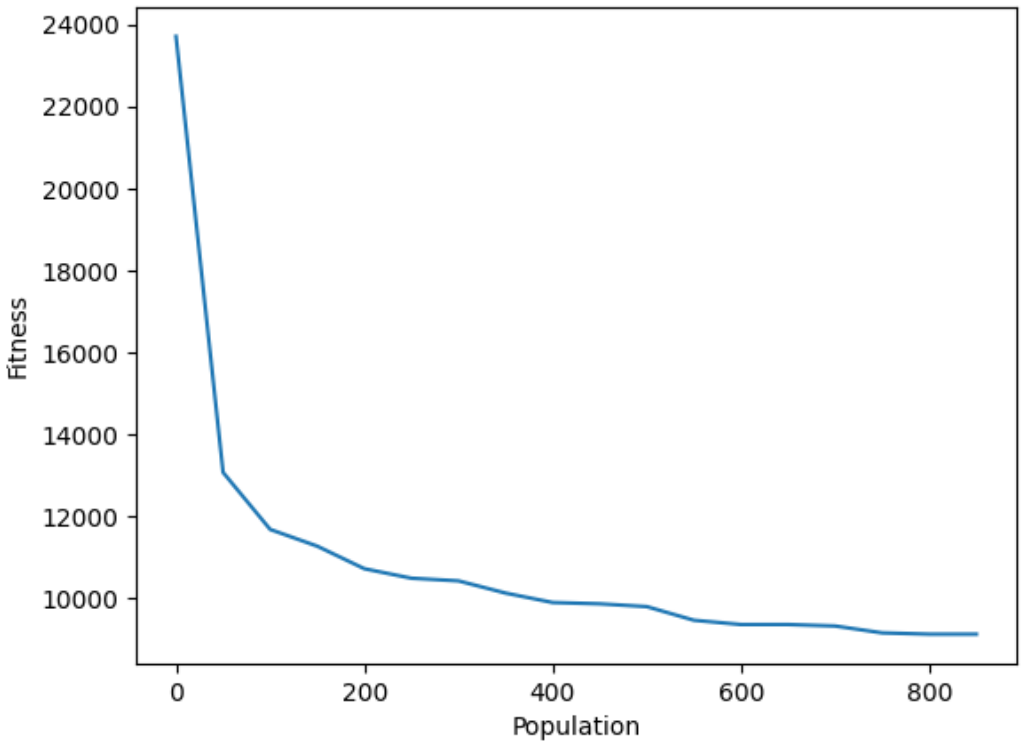
- **Explain the important operations of the employed algorithm.**
  - **parents_genesis(n_population)**
    - generates an initial population of parents and each parent is a path of cities and the fitness value is calculated using fitness_eval().
  - **offsprings_genesis(n_parents)**
    - generates a new set of offspring by randomly selecting two parents, applying crossover operation using *offspring_crossover()*, and mutating the offspring using *mutate()*. The fitness value of each offspring is calculated using *fitness_eval()*.
  - **offspring_crossover(parent1, parent2)**
    - applies a crossover operation between two parents by randomly selecting two crossover points and swapping the corresponding segments of the parents' paths. The function then returns the resulting offspring's path.
  - **mutate(n_offspring, end)**
    - mutates an offspring's path by swapping two random cities with a probability determined by the *mutation_rate* parameter.
  - **selection(n_parents, n_offsprings)**
    - selects the parents for the next generation by comparing each parent's fitness value with its corresponding offspring's fitness value. If the offspring's fitness value is better, the offspring is selected; otherwise, the parent is selected.

- **Explain the representation of the individual, a solution to the problem**
  - An individual solution is a path that includes all the 52 cities. The first city in the path is always the starting city with id 1, and the remaining cities are represented by a random permutation of the integers from 2 to 52, and in the end the path returns to the starting city.

- **Give the equation of the fitness function used by your algorithm.**
    - The fitness function used by the genetic algorithm is the total distance traveled by a given path that visits all cities in the offspring using the euclidean distance equation.

- **Give the parameters used in your algorithm**
    - **n_cities**: The number of cities in the TSP problem. It is set to 52.
    - **n_population**: The number of individuals in each generation of the genetic algorithm. It is set to 100.
    - **mutation_rate**: The mutation rate of the genetic algorithm. It is the probability of a mutation occurring in each offspring generated. It is set to 15.

- **Illustrate how the performance of the population evolves with generations**
    - A population of possible solutions is randomly initialized, often resulting in suboptimal solutions at the beginning. The algorithm then evaluates the fitness of each solution using a fitness evaluation function. The fittest individuals in the population are selected as parents for the next generation. The algorithm then generates offspring by combining or mutating the selected individuals. As the algorithm progresses through multiple generations, the population continually improves, and the solutions become better.
    - The figure in the next page □


24000
22000
20000
18000
16000
14000
12000
10000
Fitness
0
200
400
600
800
Population