Beihang University
School of General Engineering

Beijing, May 25th, 2021
Lin Wei and Yida Ding

Data Management and Artificial Intelligence
Lab Class 12

Task 1 ***N-Queens*** *Implement a solution procedure for the N-Queens problem from the lecture. Note that N can be any positive integer number.*

1. Implement an initialization function, which generates samples of a random population (=random assignments of queens).
2. Implement a function to visualize a specific sample with matplotlib.
3. Implement a fitness function, based on the number of threatening pairs, which assesses the quality of a given solution.
4. Implement a mutation function which modifies a given sample by moving one queen only.
5. Implement a cross-over function which generates a new offspring, based on two parents.
6. Implement an overall algorithm, which computes a number of given generations n and returns the best found solution. For the selection of parents, you can use the Roulette Wheel approach from the lecture.
7. Try to derive the overall pattern for a solution of the N-Queens problem. Can you find it?

Task 2 ***Traveling Salesman Problem*** *Implement a solution procedure for solving traveling salesman problem: Given a list of cities and the distances between each pair of cities, find the shortest possible route that visits each city and returns to the origin city (See Figure 22). The data of distance between each pair of nodes are shown in 'distance.csv'.*

1. Implement an initialization function, which generates samples of a random population. Each chromosome is represented as a list of cities. Note that the the selection of the origin city does not affect the result. Thus, you can keep the first element fixed in the list.
2. Implement a function to visualize a specific sample with matplotlib. The locations of cities are shown in 'location.csv'.
3. Implement a fitness function, based on the total length of the route to assess the quality of a given solution.
4. Implement a mutation function which swap the position of two randomly selected cities in a chromosome.

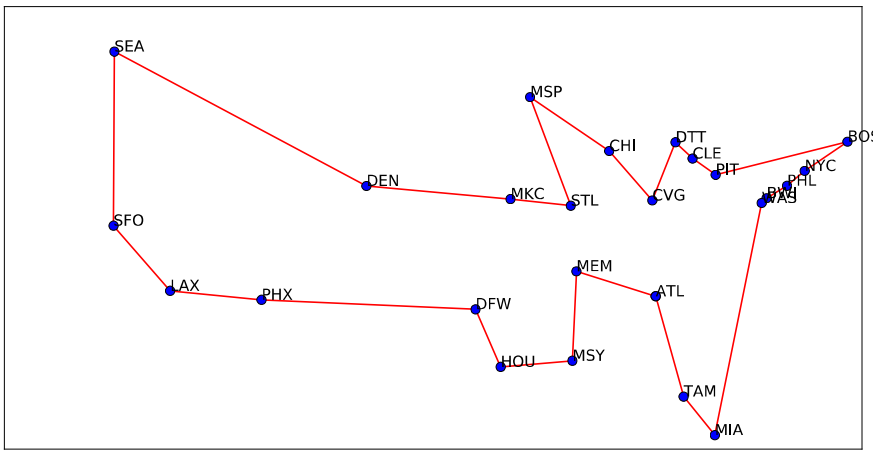

Figure 22: Traveling salesman problem

5. Implement a cross-over function which generates a new offspring, based on two parents. Note that you much keep each chromosome to be a list of all cities without duplicate.

6. Implement an overall algorithm, which computes a number of given generations n and returns the final grid. For the selection of Ma and PA, you can use the Roulette Wheel approach (See `https://en.wikipedia.org/wiki/Fitness_proportionate_selection`).

7. Use matplotlib to visualize the solution (Figure 22).