

Traveling Salesman Problem (TSP)

Develop a Genetic Algorithm (GA) to solve an asymmetric version of the Traveling Salesman Problem (TSP) using the provided dataset of cities and their travel times. The goal is to find the shortest possible route that visits each city exactly once and returns to the starting city, considering the direction-dependent travel times.

Problem context:

- In the asymmetric TSP (ATSP), the distance (or time) from city A to city B is not necessarily the same as from B to A.
- The provided dataset ([data/tsp.csv](#)) includes pairs of Catalan cities with travel times in minutes, differing for each direction.
- The cities are fully-connected, that is, there is always direct route to go from any city A to any city B.

Ideas:

- Individual: Represent each individual as a sequence of city indices, indicating the order in which the cities are visited. For example, [2, 0, 3, 1] might represent a route that visits the second city first, then the first city, and so on. Ensure Validity: each city must appear exactly once in the sequence.
- Fitness: Calculate the fitness of an individual by summing the travel times between consecutive cities in the route. Don't forget to include the return trip to the starting city.
- Crossover: Check and implement "Order Crossover (OC)", which avoids duplicate cities in offspring. Another good option could be "Partially Mapped Crossover (PMX)".
- Mutation: Two interesting options:
 - Swap Mutation: Randomly select two cities in the route and swap their positions. It's simple and effective for maintaining route validity.
 - Scramble Mutation: Randomly select a subset of the route and shuffle the cities within this subset. This can introduce more diversity.
- For the whole problem, take into account that now we want to MINIMIZE the fitness (minutes from travelling through all cities).