

# Knapsack problem

---

Develop a Genetic Algorithm to solve the Knapsack problem. The goal is to select a set of items, each with a specified weight and value, to include in a knapsack such that the total weight is less than or equal to the maximum capacity of the knapsack, and the total value is maximized.

Problem context:

- You have a knapsack with a maximum weight capacity of 15kg.
- You are given a list of items (`data/knapsack.csv`), each with a weight and a value in euros.
- Maximize the value of the items included in the knapsack without exceeding the maximum weight capacity.

Ideas:

- Individual:
  - Each individual in the population represents a possible solution to the Knapsack problem and is encoded as a binary string. Each bit in this string corresponds to an item in the list, where `1` means the item is included in the knapsack and `0` means it is not.
  - Therefore, the sequence length of each individual is equal to the total number of available items.
- Fitness function:
  - Each individual can be evaluated based on the total value of the items it includes in the knapsack.
  - However, if the total weight of such items is more than the maximum weight of 15kg, the fitness should be 0.

Extension ideas:

- Crossover: Implement uniform crossover.
- Mutate: Each child has a certain probability of being mutated. In case of mutation, flip a single bit.
- Keep best: At each new population, keep the K best individuals from the previous generation.
- Selection: Implement "Tournament Selection".