Genetic Algorithms

THE KNAPSACK PROBLEM (100 Points)

You are going on a hiking trip, and there is a limit to what you can bring. You have two things: a backpack with a size (the weight it can hold) and a set of boxes with different weights and different importance values.

The goal is to fill the backpack to make it as valuable as possible without exceeding the maximum weight (250):

1. Define the problem as a genetic algorithm.
2. Provide the genome for the problem.
3. Define all the fringe operations.
4. Cull your population by 50% at every generation.

#1	#2	#3	#4
20 6	30 5	60 8	90 7

#5	#6	#7	#8
50 6	70 9	30 4	30 5

#9	#10	#11	#12
70 4	20 9	20 2	60 1

This list represents the complete set of boxes at your disposal: for each, the top value indicates the weight (higher means heavier); the bottom value represents the importance (higher means more important). The boxes are unique. Duplicates are not allowed in your solution.

SUBMISSION

Python or C++ is the preferred implementation language. For Python, provide a plain PY file (no Jupyter Notebook). If you are writing in C++, please include a `CMakeLists.txt` and any other compilation instructions. Your code should run immediately without any additional steps on our part.

Your solution may use any numerical libraries for pre-processing, fundamental calculations (e.g., linear algebra), and visualization. However, the core portion must be implemented from scratch. If you are unsure about a specific library, ask the teaching staff first.

Submit your solution as a ZIP file with all the files via Canvas. Include a `README.txt` file that clearly explains all its assumptions.