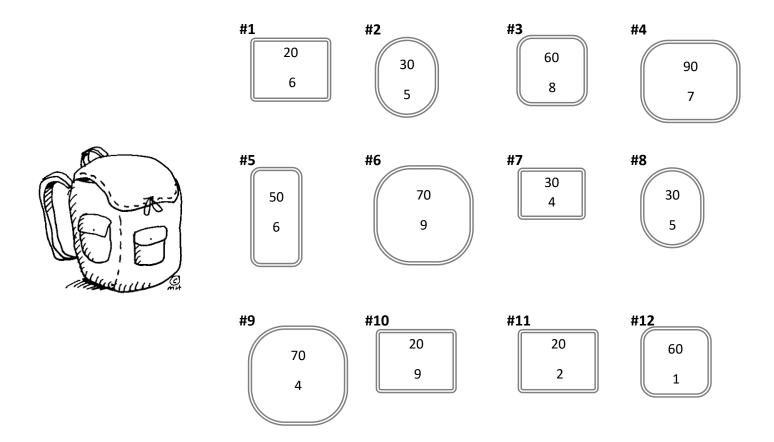# Genetic Algorithms

THE KNAPSACK PROBLEM (100 Points)

You are going on a hiking trip, and there is a limit to the things 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
20
6

#2
30
5

#3
60
8

#4
90
7

#5
50
6

#6
70
9

#7
30
4

#8
30
5

#9
70
4

#10
20
9

#11
20
2

#12
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++ are the preferred implementation languages. If you are writing in C++, please include a Makefile as well as any other instructions for compilation. For Python, simply provide a plain PY file (no Jupyter Notebook).

Your solution may make use of any numerical libraries for pre-processing, fundamental calculations (i.e., linear algebra), and visualization. However, the core portion of your solution must be implemented from scratch.

Submit your solution via Canvas and include a README file that clearly explains its assumptions.