

Knapsack problem

Deadline - March 30, 2025

A set of n objects is given with weights w_i and prices c_i . The goal in the knapsack problem is to find a subset of these objects that maximizes the cumulative price (i.e., the sum of the prices of objects in this subset), such that their total weight (i.e., the sum of the weights of objects in the subset) is less than or equal to a given limit W .

The task is to implement an evolutionary algorithm to solve this problem. You can either program the entire algorithm from scratch or use something we implemented in the exercises.

You will receive four files specifying the input. Each of these files contains several lines. The first line contains two numbers: the number of objects n and the capacity of the knapsack W . Then there are n lines, each containing the price c_i and weight w_i of one object. The values on individual lines are separated by a space.

For the input files `debug_10.txt` and `debug_20.txt`, the cumulative price of the optimal solution is 295 and 1024, respectively. You can use this knowledge to check the correctness and debug your implementation. Your goal is then to find the solution (cumulative price of the subset) for the files `input_100.txt` and `input_1000.txt`.

Submit your solution via email. Your submitted solution should include:

- Description of the algorithm (encoding of individuals, genetic operators, chosen selection method, etc.)
- Algorithm code + instructions for running it
- Graph showing how fitness changed over generations
- The best solution you discovered (especially its price)

Evaluation criteria

1. Find the best solution for `debug_10.txt` and `debug_20.txt`
2. A good score on `input100.txt` `input1000.txt` (Even relatively weak results can pass if the solution is well-documented, and you tried and documented several settings of the algorithm - hyperparams, different operators/fitness...)

Mgr. Martin Pilát, Ph.D.
Malostranské náměstí 25
118 00 Prague, Czech Republic
Martin.Pilat@mff.cuni.cz