Random search

1 WORK DURING THE LAB

1. **Exhaustive search**: generate all possible binary strings of size N and print them on the screen. Test the code for different values of N.
2. Read data for the knapsack problem.
3. Generate a **random solution** for the *knapsack problem* and verify its quality.

Points for the work during the lab: **25p**

2 ASSIGNMENT A1

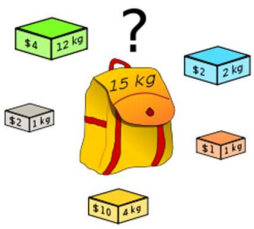
1. Implement a **random search method** for the *knapsack problem*.
   a. From k random solutions, the method should return the best one.
   b. Test the method for different values of k.
   c. Perform experiments for knapsack instances of size 20 and 200.
2. Submit source code and report.

Deadline to submit A1: **Lab 2**

Points for A1: **25p**

3 KNAPSACK PROBLEM

- n objects, each has a value (v) and a weight (w)
- *Objective: the knapsack to contain max value without exceeding max weight W*
- $x_i = 1$ means object *i* is selected
- $x_i = 0$ means object *j* is not selected



$$\text{maximize} \sum_{i=1}^{n} v_i x_i$$

$$\text{subject to} \sum_{i=1}^{n} w_i x_i \leq W \text{ and } x_i \in \{0,1\}.$$

4 REQUIREMENTS

1. Source code (notebook) needs to be documented.
2. Algorithms have to be tested for several parameter values (sufficient to clearly determine performance).
3. Experiments must be performed for all available problem instances and results compared for diferent parameter settings.
4. Results of the experiments need to be saved in output files, indicating solution quality, parameter values used, number of runs.
5. A report should capture the following: problem definition, algorithm used (name, steps/pseudocode), parameter setting, comparative results of experiments, discussion of results.