Evolutionary Algorithms (1)

1 WORK DURING THE LAB

1. Implement an **evolutionary algorithm** for the *knapsack problem*.
    a. Population initialization for binary codification
    b. Crossover operator
    c. Mutation operator
    d. Fitness function
2. Test the algorithm for the two problem instances (size 20 and 200), considering different parameter settings.

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

2 ASSIGNMENT A5

1. Implement an **evolutionary algorithm** for TSP.
2. Perform experiments for the two TSP instances selected in the previous lab and compare results, considering different parameter settings for the algorithm.
3. Compare **at least 2 different crossover** operators.

Deadline to submit A5: **Lab 6**

Points for A5: **25p**

3 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.

4 EVOLUTIONARY ALGORITHM

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```