General constraints for code submissions Please adhere to these rules to make our and your life easier! We will deduct points if your solution does not fulfill the following:

- If not stated otherwise, we will use exclusively Python 3.5.
- If not stated otherwise, we expect a Python script, which we will invoke exactly as stated on the exercise sheet.
- Your solution exactly returns the required output (neither less nor more) – you can implement a `--verbose` option to increase the verbosity level for developing.
- Add comments and docstrings, so we can understand your solution.
- (If applicable) The `README` describes how to install requirements or provides addition information.
- (If applicable) Add required additional packages to `requirements.txt`. Explain in your `README` what this package does, why you use that package and provide a link to it's documentation or GitHub page.
- (If applicable) All prepared unittests have to pass.
- (If applicable) You can (and sometimes have to) reuse code from previous exercises.

Now that you have learned about evolutionary algorithms such as Differential Evolution (DE) you will implement the following yourself.

1. **Differential Evolution for HPO** [5 points]
   We provide you with the objective function to be optimized (see `src/main.py`). You will implement DE to **minimize** this function.
   (a) Implement the evolutionary loop of DE which consists of initialization, mutation, crossover and selection operations. Use random initialization and a population size of 10 individuals (i.e. NP=10). For the mutation and crossover rates, use F=0.5 and CR=0.5 respectively.[1] [3.5pt.]
   (b) Generate plots that demonstrate the functionality of DE algorithm, e.g. best-so-far seen function value over time. Compare your results of DE against BO, Random Search and Gird Search for 100 function evaluations. Add all plots to a PDF and briefly discuss what you can learn from these plots, and add some observations about computational time for DE versus other compared optimizers.[2] [1.5pt.]

2. **Feedback** [Bonus: 0.5 points]
   For each question in this assignment, state:
   - How long you worked on it.
   - What you learned.
   - Anything you would improve in this question if you were teaching the course.

This assignment is due on 31.05.19 (10:00). Submit your solution for the tasks by uploading a PDF to your groups BitBucket repository. The PDF has to include the name of the submitter(s).

[1] Hint: Your implementation of DE should perform boundary checking after mutation to be within the search range of the problem being solved (use random re-generation for that).
[2] Hint: Your implementation of DE should run for 20 generations.