Applied Combinatorial Optimization

Dr. Bogdan Savchynskyy <bogdan.savchynskyy@iwr.uni-heidelberg.de>
Sebastian Stricker <sebastian.stricker@iwr.uni-heidelberg.de>

⋆

General information The goal of this course is to obtain the theoretical as well as practical skills. The exercises accompanying the lecture will mainly focus on programming aspects.

It is planned that you work on each exercise sheet for roughly 3 weeks. Deadlines for handing in your solution is usually 3–4 weeks. Information about exercises and deadlines are provided via exercise announcement e-mails, distributed through Müsli. Make sure to register for the course there.

Oral examination The final **oral examination** focuses on the theoretic foundation presented in the lecture whereas the practical exercises focus on programming the techniques discussed in the lecture. To **get accepted** for the oral examination at the end of the course you need to aquire **8 points** by submitting your solutions for the exercise sheets. For each complete and correct submission of one exercise sheet you will get **4 points**.

Weekly consultations During the semester we provide a dedicated time slot for consultations. If you have questions or you get stuck with an exercise, please come to the next consultation to ask questions. Participation is optional: If you do not have any questions, you do not have to attend the consultations.

Automated tests For the programming exercises we provide test instances so that you can verify your implementation before submitting. Additionally, we provide an online verification tool[1] where you can upload you source code and run automated tests. It is expected that you use this tool to check your assignment before handing it in. Any detected problem will reduce the number of points you receive.

As the exercises were changed quite a lot since the last iteration of this lecture, not all test cases will work. Please refer to the exercise announcement e-mail for up-to-date information on which online tests are still relevant. If you do not understand why the verification tool does not accept your code, you can get help during the weekly consultations. Hard-coding input or output is obviously not permitted.

Working in groups To increase exchange with other students, we allow working in a group of up to two students. While working in a group is possible it is not required. If you prefer to work alone, you can submit your solutions individually.

Setting up your computer We recommend using a Linux distribution with Python 3 installed. On Windows you can use **Anaconda**[2] to setup a working Python 3 environment.

[1] https://ofml.iwr.uni-heidelberg.de
[2] https://www.anaconda.com/download

Submission Submissions will be handled via HeiBox. Please register your exercise group to the tutor via e-mail. Be sure to mention all participants of the group and everyone's student e-mail address. You will receive a dedicated folder on HeiBox for your group to upload your submission files.

You should upload your solution before the deadline. Each submission should contain two files (where n is the number of the exercise sheet):

`exercise_n.py`:	Your Python source code.
`exercise_n.pdf`:	Non-code answers (text, formulas, figures, etc.).

Please **do not** rename the files. Ensure that the name of the authors is present in every file.

Make sure to guard any code that is not function definitions. Otherwise it can not be imported for testing:

./exercise_n.py

```
1 def fun_1():
2     ...
3
4 def fun_1():
5     ...
6
7 if __name__ == '__main__':
8     // INSERT CUSTOM SCRIPT CODE HERE
```