Barbara Hammer, Fabian Hinder
Universität Bielefeld

Applied Optimization (WS 2023/2024)
Exercise Sheet No. 1

Upload Date: 2023-10-12.
Submission Deadline: In groups of three until **2023-10-26, 7:55** to the Moodle.
Return date: 2023-11-02 in the tutorials.

In case you have questions do not hesitate to ask your tutor or to contact the tutor team at apopt@techfak.uni-bielefeld.de.

Remark: Be advised that...
- ... any exercise (sheet) handed in after the deadline will *not* be graded and is thus effectively marked with zero points. You require 50% of the total points of all exercise sheets to pass.
- ... all texts and images have to be complied into a single PDF-document or Jupyter-Notebook. If you are asked to write any code, hand it in, too. If you end up having more than one file (e.g. a PDF and a source file) create a single ZIP-file.
- ... you should always hand in your code and your obtained results, too. For example: if you are asked to plot a function you have to hand in both your code and the plot (as an image). If you use a Jupyter-Notebook, don't clean it before uploading.
- ... code that can not be executed (e.g. compiler errors) or crashes will give you zero points.

Remark: Recall the standard form of an optimization problem:

$$\begin{aligned} \min_{x\in\mathcal{X}} \quad & f(x) \\ \text{s.t.} \quad & g_i(x) \geqslant 0 && \forall i \in \{1,\dots,m\} \\ & h_j(x) = 0 && \forall j \in \{1,\dots,n\} \end{aligned}$$

where m and n are natural numbers (including zero) and $f, g_1, \dots, g_m$, as well as $h_1, \dots, h_n$ are all functions mapping from $\mathcal{X}$ to the real numbers.

Remark: Notice that you can turn an constrained optimization problem of the form

$$\begin{aligned} \min_{x\in\mathbb{R}^d} \quad & f(x_i, ..., x_d) \\ \text{s.t.} \quad & x_{j_i} \geqslant 0 && \forall i \in \{1,\dots,m\} \end{aligned}$$

into an unconstrained optimization problem by replacing x_{j_i} by $\max(0, x_{j_i})$ and adding an error term $-\min(0, x_{j_i})$ to f.

1 Warm up

(*8 points*)

Consider the functions

$$f(x,y) = \frac{1}{100}(x^2+y^2) - \frac{1}{2}(\cos(3x-3y)+\cos(3x+3y)),$$
$$g(x,y) = (x-1)^2 + 100(x^2-y)^2.$$

(a) *(1 pts.)* Create a heat map[1] of the functions in range $-10 \leqslant x, y \leqslant 10$ for f and $-3 \leqslant x, y \leqslant 3$ for g.

(b) *(2 pts.)* Describe the shape of the two functions: How do they look like? Are there "jumps" or "flat regions"? What about the optima? How many are there? Are those global or local?

(c) *(3 pts.)* Write a Python-script that randomly generates a point p_0 in the areas as specified above and optimize the function using the Nelder-Mead algorithm[2], using $p_0 \in \mathbb{R}^2$ as a starting point, to obtain a point p_1.

(d) *(2 pts.)* Repeat the procedure from *(c)* 10 times and add the resulting pairs of points to the heat maps from *(a)* (connect pairs of p_0 and p_1 by lines). What do you observe? Does the method converge to a unique optimum or are there many? Did the algorithm produce good results or not? And why?

[1] See `https://matplotlib.org/stable/gallery/images_contours_and_fields/image_demo.html` for a useful demo.
[2] You may use the SciPy implementation (see `scipy.optimize.minimize`). Do *not* implement the method yourself!

2 Modeling

(*12 points*)

Remark: The following problem is not strictly specified and you cannot find a "correct" solution, and it is unlikely that your solution will be "beautiful". Instead, think of it as a task that allows you to practice your problem-solving skills. Imagine the following: Your boss or a customer comes to you and asks you to solve this problem. Since reality is usually quite messy and it is never really right or wrong, it is plausible that the answer to this question will be the same.

Consider the following problem: You want to sell paper boxes. To make a box, take two identical open boxes (boxes without lid side) and put them together by sliding them into each other. Each open box you use is made from a single sheet of paper (see Figure 1 for an illustration. Note, that the sheet does not have to be a square). You get $2.50\$$ for every cubic centimeter of volume! However, you will have to pay $0.75\$$ for each square centimeter of paper you use. The larger the paper, the more complicated it becomes to make a single box. Up to a constant time you need to grab the paper, you observe that if you double the length of the longest side, it will take you about four times as long to make a box;[3] the shorter side of the sheet doesn't seem to matter. Keep in mind that your workday has a limited number of hours. How to get rich?

(a) *(4 pts.)* Formalize the problem described above. Explain your choices!
Note that the above description does not lead to a unique formalization, but there are a few issues which you can decide on in a reasonable way (such as labor costs to make the box). In particular, the explanation is just as important as the formula you come up with.

(b) *(2 pts.)* Turn your formalization into an optimization problem in standard form.

(c) *(3 pts.)* Implement your approach: Write a Python-script that optimizes a reasonable starting point using the Nelder-Mead algorithm.

(d) *(3 pts.)* Discuss your results. Do you obtain a good solution? Can you find an analytic solution or get the same insight without the help of a computer?
Also criticize the setup and the decisions you made in part (a): Is there enough information in the task description? Do you need additional (world) knowledge? If "yes", which could be the most relevant? And what are indicators for that? What is the advise you can give on this problem?

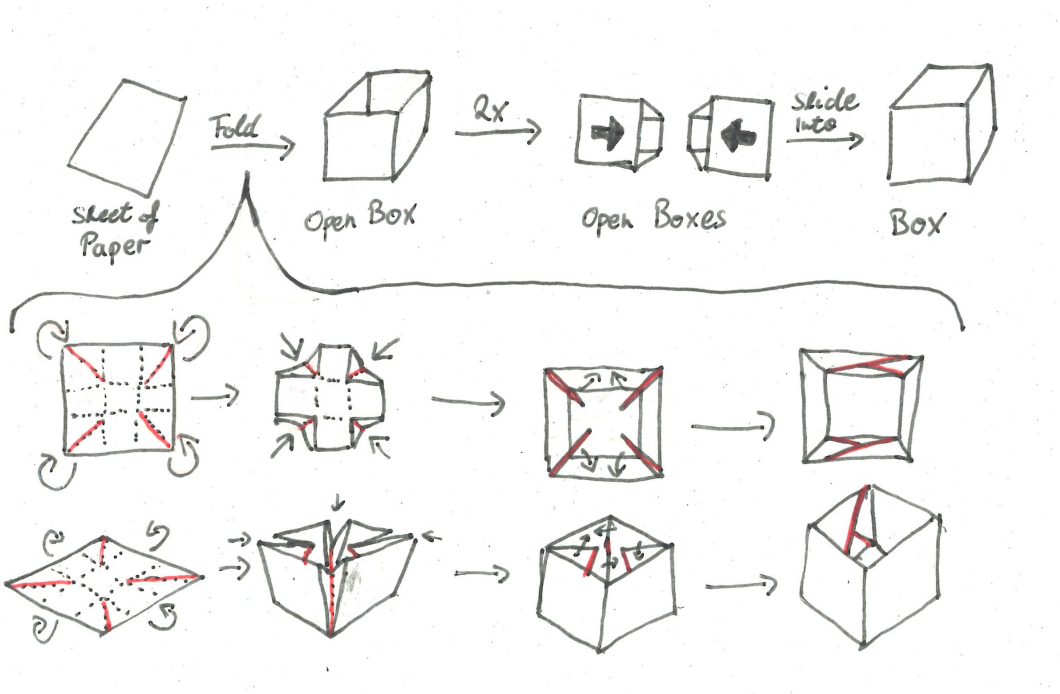

Figure 1: How to make a box

[3] Recall that $(2x)^n = 2^n x^n$. You may assume that the relation of time to the longer side is of this type.