Exercise 1: Hill Climbing

Due date: *October 16th, 2025 (push to GitHub repo before the lecture)*

General Requests

- set up a GitHub repository in pairs of two and invite me to it (hannah.wimmer@fh-joanneum.at)
- use uv for package management (more information here)
- use streamlit for summarization and documentation (more information here). use a pages folder setup as shown in Create a multipage app to allow later additions of future exercises; a demo can be found here

2.1 Benchmark Functions

Implement the following test functions taking inputs $x \in \mathbb{R}$ or $\vec{x} \in \mathbb{R}^n$, respectively:

- Quadratic: $f(x) = x^2$
- Sinusoidal: $f(x) = sin(x)$
- Ackley: $f(\vec{x}) = -a \cdot \exp(-b \cdot \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(cx_i)) + a + \exp(1)$[1]
- (optional) Rosenbrock: $f(\vec{x}) = \sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2]$[2]
- (optional) Rastrigin: $f(\vec{x}) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$[3]

Visualize everything in a meaningful way (line plots, surface plots, contour plots - your choice). You're free to implement more test functions if you're interested; see Molga and Smitnicki for other functions and their definitions [1].

2.2 Hill Climbing Algorithm

Implement the **Simple Hill Climbing** algorithm and **one variant of your choice** (e.g., steepest ascent, next ascent, adaptive, or some design you can come up with)[2, Chapter 2].

- your implementation should take an initial solution candidate (n-dimensional), a fitness function, a neighborhood function (assume continuous domain), and a termination criterion as input values to allow later reuse
- track the number of objective function evaluations ("$f(x_{new}) > f(x_{old})$?") to justly compare between variants
- use your algorithms to find minima (!) of the benchmark functions (quadratic and sinusoidal in 1D, Ackley in 2D) and visualize your solution's progression (search trajectory) in a meaningful way

2.3 Analysis and Reporting

Use streamlit to document your insights on Hill Climbing. Feel free to go wild on layout and functionality; to make it as simplistic or fancy as you like. Your final documentation webpage should contain the following sections:

1. **Introduction** - Overview of the algorithm: basics, strengths/weaknesses, complexity, ...
2. **Methods** - Describe your implementations: pros and cons, relevant parameters, critical design choices, ...
3. **Results** - Display your results; allow adjustment of settings (test function, neighborhood size, ...)
4. **Discussion** - Analyze your findings: expected versus unexpected results, solution quality, efficiency, performance differences between variants, dependency on parameters, limitations, possible improvements, ...

The documentation doesn't need to be absurdly long, but it should give a comprehensive overview of the topic that you can use later on to study for the exam.

References

1. Molga, M. & Smutnicki, C. Test functions for optimization needs. 3 kwietnia 2005. https://robertmarks.org/Classes/ENGR5358/Papers/functions.pdf.
2. Simon, D. Evolutionary Optimization Algorithms https://research-1ebsco-1com-1195qzf320241.perm.fh-joanneum.at/c/kofjhs/search/details/4sh2uyq6wn?db=nlebk&db=nlabk (Wiley, 2013).

[1] parameters are classically set as $a = 20$, $b = 0.2$, $c = 2\pi$; test area around $10\pi \leq x_i \leq 10\pi$
[2] test area around $-2 \leq x_i \leq 2$
[3] test area around $-5 \leq x_i \leq 5$