

Assignment 5

For this assignment, form groups of two. This assignment is worth a total of 40 points. Solutions must be submitted by 1.6.2025. Use the link on e-ucilnica to submit your work. The **report** must be in .pdf format. You should also submit **source code** of your implementation and **results** in .txt format.

Continuous optimization of CeC 2022 functions

The aim of this assignment is to find the best result for 12 optimization (minimization) functions that are available in *opfunu*[2] package. The functions used will be from Cec2022 benchmark found here[1].

Instructions

To test the results use all 12 functions with 20 dimensions with bounds in default ranges from -100 to +100 for each dimension. You can see the lower and upper bounds of each function in variables *lb* and *ub*. Check appendix and supplementary code for example.

As a team you need to implement at two or more optimization programs. One of the optimization programs must be local search (best descent local search, tabu search, guided local search, variable neighborhood search, simulated annealing, etc...) and the second one can be any optimization approach of your choice (genetic algorithm, differential evolution, whale algorithm, ant colony optimization, etc...). The second approach can also be local search.

You need to implement the approaches yourself and not use already build optimizers from other packages. There is no time limit on how long you can let the algorithms running.

Reporting

Write your results into the Google Spreadsheet available here. Each group is encouraged to fill their results as soon as it has them available so that peers can see what are currently the best obtained results. The groups should be written in spreadsheet no later than 25.5.2025.

Write a report with your results and description of used approaches. Report should include the results for each tested method separately and a short description of each method (1-2 pages for each method). Submit your code and report on e-ucilnica.

You should also submit coordinates of the found minimums for each method in a .txt file. The file should contain 12 lines, each line representing its own function. The values of coordinates should be separated by tabulator. See example on e-ucilnica.

Grading

Final grade will be based on quality of results, quality of report, oral presentation, number of methods tested and code quality.

Appendix

Python example

```
1 from opfunu.cec_based import cec2022
2
3 # Choose the first five benchmark functions
4 functions = [cec2022.F12022, cec2022.F22022, cec2022.
5             F32022, cec2022.F42022,
6             cec2022.F52022, cec2022.F62022, cec2022.
7             F72022, cec2022.F82022,
8             cec2022.F92022, cec2022.F102022, cec2022.
9             F112022, cec2022.F122022]
10
11 for f in functions:
12     try:
13         # Initialize the 20D function
14         func = f(ndim=20)
15
16         # Find the recommended bounds and evaluate
17         # function at point 0
18         lower = func.lb
19         upper = func.ub
20         x = [0] * 20
21         # Evaluate the function
22         fitness = func.evaluate(x)
23
24         # Print and store results
25         print(f"{f.__name__}: {fitness}={fitness:.3f}")
26     except Exception as e:
27         print(f"Error: {f.__name__}: {e}")
```

References

- [1] Opfunu Cec2022 based module. https://opfunu.readthedocs.io/en/latest/pages/cec_based.html#module-opfunu.cec_based.cec2022. Accessed: 2025-05-05.
- [2] Nguyen Van Thieu. Opfunu: an open-source python library for optimization benchmark functions. *Journal of Open Research Software*, 12(1), 2024.