Optimization and Decision Making

Exercise Sheet 2

Exercise 2.1 (Setup VPN and API Access)

In this project, you are tasked to solve continuous black-box optimization problems. You cannot access to the underlying problems, but you get access to an API endpoint that you can query with decision vectors to retrieve the evaluated objective vector or gradient.
To access the API, you need to be logged into the Uni Münster VPN.[1] Then, you can query a point in the black-box service by sending a POST request to `http://ls-stat-ml.uni-muenster.de:7300/OP/FID` (with placeholders OP and FID) containing the decision vector and a group-specific token.

A Jupyter notebook with the basic setup to connect to the black-box service and perform requests is provided in the LearnWeb!

As the token, always use your LearnWeb submission group number, e.g., if you are Lecture Working Group 42, you use `token=42`. For OP, you have the following choices:

- `compute`: Computes the objective value at the queried point.
- `compute_gradient`: Computes the gradient at the queried point.

For FID, you can substitute: `Function1`, `Function2`, `Function3`, `Function4`, or `Function5`. That is, there are five different black-box problems in total. The underlying problems are unconstrained and can be evaluated at any (two-dimensional) real-valued vector. However, for all problems, the “area of interest” guaranteed to include the global optimum is given by $\mathcal{X} = [-5, 5]^2$. So, for example, `http://ls-stat-ml.uni-muenster.de:7300/compute_gradient/Function2` is the endpoint for the gradient of the second problem. With the VPN and API connection set up, perform the tasks on the following page.

[1] To setup your VPN connection, please follow the instructions provided at: `https://www.uni-muenster.de/IT/en/services/kommunikation/vpn/index.html`.

Exercise 2.2 (Implement and Compare Optimizers)

Compare how well the optimization approaches you learned about in the last two lectures (04 and 05) work on the problems provided by the API. Use these guiding questions:

- Are there problems favoring a particular optimizer over another?
- How do you compare the results between two optimizers?
- How can you fairly compare approaches using the gradient with those that don't?
- Is the result sensitive to the starting point?
- How does the configuration of the optimizer influence the result?

You can implement some algorithms yourself or use external libraries / sources.

Exercise 2.3 (Analyze Problem Properties)

With the results from the previous exercise, can you come up with problem groups that share major characteristics? Think about problem landscape visualizations (such as contour plots), optimizer traces, etc., to get a better understanding of the problem characteristics.