Evolutionary Algorithms

Practical Assignment 2

Black-Box (Continuous) Optimization Using Evolution Strategy

In this practical assignment you are presented with 24 black-box continuous problems. Each problem consists of a function that is to be minimized:

$$f: [-5,5]^n \rightarrow \mathbb{R}$$

with *n* = {2,5,20} (i.e., 2-dimensional, 5-dimensional and 20-dimensional search space)

The introduction of problems can be found at
http://coco.lri.fr/downloads/download15.01/bbobdocfunctions.pdf.

Your algorithm is to be evaluated with empirical cumulative distribution function (ECDF), which allow to aggregate performance across different functions and dimensions. For each of the 24 different problems $i = [1..24]$ and dimension $n = \{2,5,20\}$, there is a list of target values $\varphi_{i,n,j}$, $j = \{1, \dots, m\}$ are chosen. The ECDF value at budget t is defined as the fraction of (run,target)-pairs $(s, \varphi_{i,n,j})$, which satisfy that in run s on problem i, a solution has been identified that is at least as good as $\varphi_{i,n,j}$. All submitted algorithms will be ranked by the area under ECDF, which is averaged ECDF value at a pre-defined set of budgets. You can achieve the value of the area under ECDF from IOHanalyzer.

Assignment

Implement an Evolution strategy (ES) using IOHexperimenter according to the generational ES model.

Test the performance on BBOB suite, analyze your ES implementation with IOHanalyzer, and provide the setting of ESs that you suggest for the suite.

Use a maximum of 10,000n function evaluations for each instance, and report your results averaged over 25 instances.

Deadline: before 7 December 2020, 23:59.

Your submission should consist of two files (no more, no less): a report in PDF format and the ES implementation. These are to be uploaded to Brightspace.

Read carefully the *Implementation Details* and the *Report Structure Guidelines* provided below.

IOHexperimenter Implementation Details

Your ES implementation should consist of **one** file named as ***studentname1_studentname2*_ES** (replacing studentname1 and studentname2 by your own last names) and should be structured as follows:

For C++:

```
void
studentname1_studentname2_ES(std::shared_ptr<IOHprofiler_problem<dou
ble>> problem, std::shared_ptr<IOHprofiler_csv_logger> logger) {
…
}
```

```
For Python:
def studentname_studentname_ES(problem):
…
```

Here, *problem* is the problem class to be tested, *logger* is used to generate csv files that are available for IOHanalyzer. There are many useful member functions of *problem* and *logger*, please follow the given example files and watch the videos introducing IOHprofiler.

We will compare all submissions using an automated script, this requires all plotting, printing to the command line etc. to be disabled in the submitted version of your work! If you apply specific parameters setting on different problems, please make those parameters as variables that can be configured easily, and provide the settings in the report clearly.

Report Structure Guidelines

Title + Authors (names, email addresses, and student numbers)

Introduction

Introduction text here.

Implementation

Outline of your algorithm, algorithm parameters, and settings used for those parameters. Make sure that the algorithm and the results are reproducible from your description.

Experiments

Description of the experiments and the results. Use the tables and figures generated from IOHanalyzer. Describe your suggested algorithm settings for each problem.

Make sure to present your results in a way that is convenient to the reader, do not blindly include plots of all your experiments, try to combine data in figures!

Discussion and Conclusion

Summarize the results and conclude your report.