Evolutionary Algorithms

Practical Assignment 2

Real-Valued Optimization Using Evolution Strategies

In this Practical Assignment you are presented with five *black-box* optimization problems. Each problem consists of a function with real-valued input parameters that is to be minimized:

$f(\mathbf{x}) \rightarrow \min$, $\mathbf{x}$ in R^N

subject to $-100 < x_i < 100$, for all i in$\{1,\ldots,N\}$

with $N = 30$ (i.e., a 30-dimensional search space)

The five black-box problems can be found in the file **PA2.zip**, to be downloaded from blackboard.

Assignment

Implement an Evolution Strategy (ES) in MATLAB using (μ +/, λ)-selection to optimize these problems. The choice between plus or comma selection can be made beforehand, or depending on what you discover to deliver the best results w.r.t. the given optimization problems. Your algorithm should be able to operate on arbitrary values of μ and λ.

Describe your implementation by means of pseudocode, accompanied by a textual explanation, providing the possibility for the reader to easily re-implement the ES exactly as you did and re-creating/verifying your results. Report on the performance of your algorithm, presenting the average performance over 20 runs on each of the five test problems, using a budget of 10,000 function evaluations per problem.

Deadline: before 7 December 2018, 23:59.

Your submission should consist of two files (no more, no less): a report in PDF format (≥4 pages) and the ES implementation in MATLAB. These are to be uploaded to blackboard.

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

MATLAB Implementation Details

Your ES implementation should consist of **one** .m file named ***studentnumber1_studentnumber2*_es.m** (replacing lastname1 and lastname2 by your own names) and should be structured as follows:

```
function [xopt, fopt] =
studentnumber1_studentnumber2_es(@fitnessfct, N, lb, ub,
eval_budget)

…

end
```

Here, *fitnessfct* is a handle to the fitness function, *N* denotes the dimensionality of the input to *fitnessfct*, *lb* is an *N*-dimensional row vector of lower bounds, *ub* is an *N*-dimensional row vector of upper bounds, and *eval_budget* denotes the function evaluation budget; *xopt* is the best solution found and *fopt* the accompanying best fitness value.

Each black-box fitness function is of the following form:

```
f = bbf1(x)

…

end
```

To run your ES on *bbf1* with N=30, $lb=[-100]^N$, $ub = [100]^N$, and a budget of 10,000 function evaluations you type

```
[xopt, fopt] = studentnumber1_studentnumber2_es(@bbf1, 30, -100 *
ones(1,30), 100 * ones(1,30), 10000)
```

The *bbf* functions expect an *N*-dimensional row vector *x* as input (i.e., a vector with a single row and *N* columns).

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! Submit one ES, and configure it to the tested settings that you found to perform best.

Report Structure Guidelines (≥4 pages)

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

Introduction

Introduction text here.

Problem Description

Brief description of the optimization problem 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 following tables and figures to report on the results:

	ES configuration A		ES configuration B	
Benchmark	Avg	Std dev	Avg	Std dev
1				
2				
3				
4				
5				

Table 1: Final solution quality after 10,000 function evaluations, averaged over 20 runs

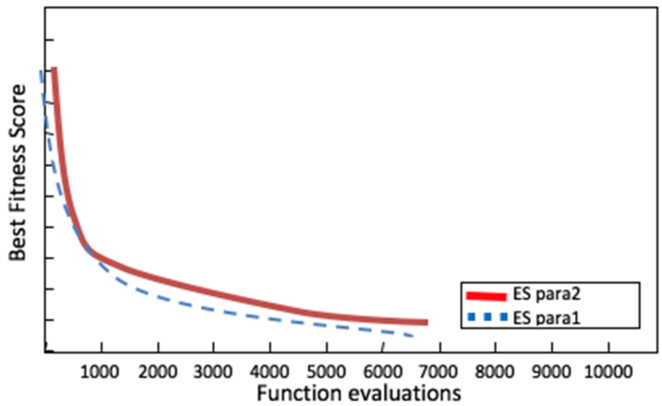

Figure 1: Algorithm convergence, averaged over 20 runs

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.