

1 Detailed Instructions

The assignment requires you to implement the above four operators (Algorithms 2, 3, 4 and 5 in the appendix) within a simple (1+1) evolution strategy only using mutation (see Algorithm 1 in the appendix) on the SPHERE function in a programming language of your choice (e. g., Java, C, C++, Matlab). You will then run a set of experiments as detailed below to assess the performance of the mutation operators. Additionally, you are required to describe and discuss your implementation and your experimental findings in a report. The report should in particular discuss the following aspects (amongst any other points which you might consider important):

- How did you implement the algorithm and the fitness function (e. g., programming language, design decisions, . . .)? What data structures, algorithms or libraries did you use, e. g., for a random number generator? What other software/scripts did you use to facilitate your experiments/analysis?
- How does the performance of the different operators compare? How do parameters of the operators/the fitness function influence the performance?
- Try to explain why the operators behave the way you observe in your experiments. What properties, strengths or weaknesses can you observe? Can you name reasons for them?

As a minimum you need to run experiments using the following parameter settings. You are free to test additional settings if you feel these will help you with the discussion. Make sure to include these in your report as well. Note that uniform mutation does not have any parameters.

- A 10-dimensional SPHERE function (i. e., $n = 10$) with search space restricted to $[-100, 100]^{10}$
- Non-uniform mutation with parameter $b \in \{0.05, 1.0, 10.0, 20.0\}$.
- Gaussian mutation without 1/5-rule and parameter $\sigma \in \{0.05, 0.5, 1.0, 10.0\}$
- Gaussian mutation with 1/5-rule and initial uniformly distributed over $[1, 100]$.

You need to perform 30 independent runs for each of the above settings. Each run should have an upper limit of 5,000 iterations. For each of the runs you need to record the fitness value for each iteration of the run. Visualise and present your data as follows:

- Plot the average function value over time (i. e., iteration number on the x axis; average function value over the 30 trails on the y axis). Use logscale for the y axes to improve visibility. You should have such a graph for each of the following scenarios:

- a figure comparing different parameters for non-uniform mutation
- a figure comparing different parameters for Gaussian mutation without 1/5 rule
- a figure comparing uniform mutation, Gaussian mutation with 1/5 rule and the best (i. e., best fitness value after 5,000 iterations) parameter setting for non-uniform and Gaussian mutation without 1/5 rule
- Create a table stating the average fitness value and its standard deviation (over the 30 runs) after 5,000 iterations for each of the considered settings. You can determine these values with a statistics or spreadsheet software of your choice (e. g., R, Matlab, Microsoft Excel, LibreOffice).