Evolutionary Algorithms: Group report

Group Member 1, Group Member 2, and Group Member 3

October 15, 2024

Formal requirements

Please respect the structure of this template. You can remove the instructions in this section from your report. The blue text should be replaced with your discussion. Your report can be **at most** 3 **pages** long.

It is recommended that you use this LaTeX template, but you are allowed to reproduce it with the same structure in a WYSIWYG-editor. You should replace the blue text with your discussion. The questions we ask in blue are there to help you decide which topics to discuss, rather than an exact list of questions that must be answered.

This report should be uploaded to Toledo by November 8, 2024 at 9:00 CET. It must be in the **Portable Document Format** (pdf) and must be named `r0123456_intermediate.pdf`, where r0123456 should be replaced with your student number. **Each group member should hand it in individually on Toledo.**

1 An basic evolutionary algorithm

1.1 Representation

How do you represent the candidate solutions? What is your main motivation to choose this one? What other options did you consider? How did you implement this specifically in Python (e.g., a list, set, numpy array, etc)?

1.2 Initialization

How do you initialize the population? How do you generate random cycles?

1.3 Selection operators

Which selection operators did you consider? If they are not from the slides, describe them. Which one did you implement? Can you motivate why you chose this one? Are there parameters that need to be chosen? What do you think are reasonable parameter values?

1.4 Mutation operator

Which mutation operator(s) did you consider and implement? If they are not from the slides, describe them. Which one did you implement? Why did you choose that one specifically? Do you believe it will introduce sufficient randomness? Can that be controlled with parameters?

1.5 Recombination operator

Which recombination operator(s) did you consider? If they are not from the slides, describe them. Which one did you implement? Why did you choose that one specifically? Explain how you believe that this operator can produce offspring that combine the best features from their parents. How does your operator behave if there is little overlap between the parents? Can your recombination be controlled with parameters; what behavior do they change? Do you use self-adaptivity?

1.6 Elimination operators

Which elimination operators did you consider? If they are not from the slides, describe them. Which one did you implement? Can you motivate why you chose this one? Are there parameters that need to be chosen? What do you think are reasonable parameter values?

1.7 Stopping criterion

When do you think the evolutionary algorithm should stop? Which stopping criterion did you implement? Did you combine several criteria?

2 Numerical experiments

2.1 Chosen parameter values

What parameters are there to choose in your basic evolutionary algorithm? How did you approach parameter selection for now? Did you experiment a bit with different values? What was the influence of this choice? Which are the parameters that you will use for the experiments below?

2.2 Preliminary results

Run your algorithm on the smallest benchmark problem. Include a convergence graph, by plotting the mean and best objective values in function of the time. How long did your program run until convergence? How do you rate the performance of your algorithm (time, memory, speed of convergence, diversity of population, etc)? What was the best fitness value you found? Do you think this is the global optimum?

When you solve this problem several times, how much variation do you observe in the best solution? Include an informative plot of this.