NaCo 22/23 assignment 2 report, group number

First and last name of each student in the group

Leiden Institute of Advanced Computer Science, The Netherlands

Abstract. This document contains the instructions and the format for the report required for submission of the practical assignment for the course Natural Computing.

1 Introduction

This document serves as a *description of the practical assignment* for the course Natural Computing in academic year 2022/2023. For this assignment, you will extend your work from assignment 1 to a different problem. For this assignment you are asked to provide:

- Modified implementation of a GA in plain Python,
- a report, written and formatted as a *scientific paper* containing:
  - Description of the modified GA and its implementation
  - Description of the problem (CA inversion) and a review of an application of CA in practice
  - Clear comparisons of your GA variants on the different versions of the CA inversion.

To help you structure your report, we provide you with a *brief report outline* in this document. Please complete the following sections with your own results, explanations and conclusions. This includes the abstract and this introduction! For this section: introduce what the paper is about and provide a background to any relevant literature, such as the Genetic Algorithm [1].

2 Problem: Cellular Automata

For the second part of the assignment, you will need to implement a basic 1-dimensional Cellular Automata (CA) of size N (non-connected boundaries, fixed to 0), which can follow a transition rule Θ, with a neighbourhood size $r = 1$. You should create a function to run your CA and provide an output state after T time steps for a given input state. The number of possible values k at each site for your CA should be flexible. We will use both $k = 2$ (the binary case shown in the lecture, with $a_t^t = \{0, 1\}$) and $k = 3$ ($a_t^t = \{0, 1, 2\}$) in this assignment. Note that you should implement a CA from scratch, following the explanations of CA in the slides.

With this CA, you will try to solve a version of the *Inverting Problem*: find a state C^0 such that when you use it as the initial state of your CA, after T time steps you get the given state C^T as output. You will do this for a number of different CA configurations, which will be given in the supplementary material. In there, you are given the final state C^T, number of time steps T, and transition rule Θ to use for each configuration. Note that both $k = 2$ and $k = 3$ are considered in these configurations.

Your implementation should be in Python, since you will also use your implemented GA in this assignment. In a similar fashion as with the GA in part 1, you should provide a description in your implementation of the CA in your report.

3 Solving the Inverting Problem

In order to find this input state C^0 for a given output state C^T, you will use the GA you described in part 1.

The individuals in your GA will correspond to variations of input states $C^{0'}$ for your CA. Note that for a 2-state CA you can use the default binary operators, but for a 3-state CA you should make sure to modify your GA such that it can handle this different representation!

To run your GA, you need to define an *objective function* for the inverting problem given a configuration of your CA. This should be function which calculates a similarity between a given input state $C^{0'}$ as suggested by the GA, and the actual input state C^0 of the CA. Since many different implementations are possible, you are asked to describe and implement *two different objective functions* here, and compare the performance of your GA on these two versions of the problems. In the file `objective_function.py` in supplementary material, an example is given of creating your own objective function using the `ioh` framework, in addition to running a simple experiment.

In the file `ca_input.csv`, your input is given. The file contains 10 states and rules for your CA. Note that in addition, the number of time steps T to run a given rule and state combination is also given. You should process all these 10 CA states in your experiments, for both of the two aforementioned objective functions, giving you effectively 20 functions to conduct experiments on. Moreover, you should compare the performance for the different operator combinations. You don't have to run all combinations on every function, but you should try to conduct at least 3 main sets of experiments (as an example, one experiment could be comparing the performance of all operator combinations on one problem, or comparing the performance of a fixed combination of operators across all 2-state problems). [1].

To analyze the performance of your GA, you should analyze the data you collected with `ioh` in the `IOHAnalyzer`. In this report, include the performance curves of your GA for each main experiment. Make sure to in at least one experiment include a comparison to a basic random search method. Clearly motivate each experiment and explain the setup, results and possible conclusions we can draw from this.

4 Algorithm Description

Give a general overview of the working principles of a Genetic Algorithm (GA). This can be copied from your first assignment, but you should describe how you modified your previous operators to deal with the ternary representation (needed for the 3-state CA). Make sure to stick to the notation as described in the first assignment wherever possible.

[1] For these experiments, you should use a budget of 10000 evaluations, and do 5 independent runs of the GA on each function.

5 Application of CA in practice

Search in the literature for one paper which makes use of a CA outside of the field of computer science. Minor students are encouraged to take the lead in writing this section. Summarize the paper and describe the application, and add any relevant literature. In this section, be sure to answer at least the following questions about the paper:

- What kind of CA was used (dimensionality, states, transition rules,. . .)?
- Describe the field and context.
- Why did the researchers choose a CA for their application, and where there any alternatives?
- Was their approach successful? Interpret their results.
- Give your opinion on their approach. Would you have you have used a CA in their situation?
- How would you improve on their setup?

6 Conclusion

Write a short conclusion summarizing the most important findings of this assignment.

A Appendix

A.1 Submission, review and grading

For this assignment, you should submit the full report following this template. Please be sure not to exceed the **limit of 12 pages** for this report, excluding references.

Submission should be done on Brightspace, and should include your report in PDF format and your code as a Python file in a single zipfile. Make sure that the code is readable: clear variable names, comments,. . . .

References

1. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975), second edition, 1992