NAT & NAT-DL 2023/2024: Assignment[1]

Natural Computing

--- Preliminary Version ---

This is an exercise worth 40% of the course mark. The maximal score you can get for this assignment is 100 points. The assignment is to be completed by yourself individually. You are asked to submit an individual solution, and not to discuss or share your solution with other students. The assignment requires you to write programs in Python[2] (you can use Jupyter notebooks), with which you need to carry out a number of investigations related to three problems. You will need to submit your program code (zip) and a report (pdf). It can be useful, if you include some comments in your code, but there will be no marks specifically for the code, see the marking breakage below.

Problem 1 (30/100): The effect of the population size.

In this task you are asked to investigate the effect of cooperation within a population. Choose a population-based metaheuristic optimisation algorithm (however, do not use a GA for this problem) and a non-trivial objective function that allows for a variation of difficulty of the problem. For example, if you choose the Rastrigin function

$f_2(x) = 10d + \sum_i (x_i^2 - 10\cos(2\pi x_i))$, where $x = (x_1, \ldots, x_d)$, $x_i \in [-5.12, 5.12]$, $i = 1,\ldots,d$

as your objective function, you can set d=1, 2, … in order to vary the search space dimension and, thus, the problem difficulty.

a) For a constant total number of evaluations of the cost function and for a fixed problem difficulty, determine (approximately) the optimal population size. (20/30)

b) How does this optimal number depend on the problem complexity? (10/30)

Problem 2 (40/100 marks): *Sumplete* is a game similar to Sudoku, and, allegedly, it was invented by ChatGPT earlier this year. The task in this game is to de**lete** some of the entries of a k-by-k matrix of random numbers so that the rows and columns **sum** u**p** to given values, see Fig. 1.

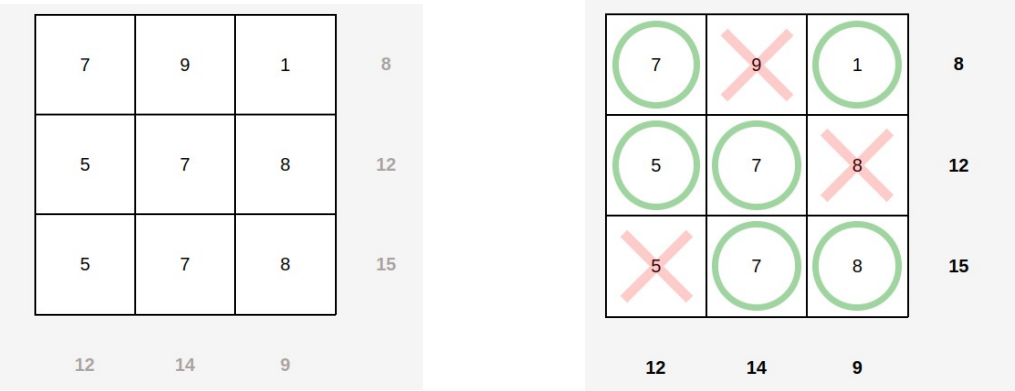

Figure 1. A 3-by-3 Sumplete example problem (left) and its solution (right).

Use a genetic algorithm to realize a solver for the Sumplete game. You can find sample problems at sumplete.com, but it is easy to generate problems yourself (suggested deletion rate: 1/3, entries: small positive random integers, $k \geq 3$), just make sure not to use any information from the problem generation in the solution in an unfair way.

1 The text of the final version of the assignment may differ in details from this preliminary version.
2 Alternatively, you can use C/C++.

After explaining (10/40 marks) your choice of an encoding, of algorithmic details and parameters,

a) run the algorithm with a fitness function that assigns a fitness of 1 to a correct solution and a fitness of 0 to all incorrect solutions. Show the performance of your genetic algorithm for various values of $k \geq 3$. (10/40 marks)

b) Find a fitness function for the game so that the behavior of the algorithm scales more favorably with k. Explain your choice. (10/40 marks)

c) Discuss the effect of the GA parameters on the solution. (10/40 marks)

Problem 3: Genetic Programming (30/100)

The On-Line Encyclopedia of Integer Sequences® (OEIS®) (see also Wikipedia) presents many examples of interesting number sequences. Choose a squence and try to find a rule that generates this series by genetic programming.

a) Explain your choice of such a sequence, and discuss your GP design for this problem. (10/30)

b) Show and explain the results of the GP on this problem. (10/30)

c) Test and discuss the generality of your approach. (10/30)

You may find the following comments useful for your work on the assignment:

For most questions, computer experiments will help to substantiate your answer. If you answer any questions solely based on literature, you may loose marks. Likewise, you will not get full marks, if you submit numerical results without any explanation.

If you need to choose values of parameters of the algorithms, explain and justify your choice at least briefly.

If you perform comparisons, make sure that the comparison is reasonable and fair, and discuss your evaluation procedure.

Explain your decisions on the termination criterion, and on the number of repetitions of runs.

Wherever possible, use graphical representations of your numerical results.

Although the report needs to be concise, try to include justification and explanation of your work, rather than just a list of the steps that you have carry out.

Structure of the report

In the report on your work on this assignment, start a new page for each of the three questions. If you choose to add a front page or a preface, restart the page numbering that Question 1 starts on page 1. There is no page limit for the report, but please take some effort to keep your answers and the total report short and concise. If you prefer to include material which is redundant or not directly related to the tasks, please put it in an appendix.

Feel free to use any existing resources and literature to prepare your work and to support your argument, but make sure to cite all papers and resources that you have used for this assignment. Place this list of references at the end of the report or after the appendix.

Submit your report as a PDF file and your code as a separate zipped file via Learn. Deadline for submissions is **12 noon of Thursday, 16. November 2023.**

In the tutorial sessions and in the classroom Q&A sessions will be time to clarify your questions related to the assignment. The final version of the assignment will be available by 12/10/23 . For critical questions, please contact michael.herrmann@ed.ac.uk.