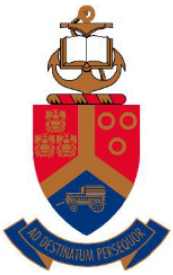

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 314: Artificial Intelligence
Assignment 2: Sudoku
Due Date: 17 June 2021

1 Assignment Outline

Sudoku is a well known Japanese logical puzzle. The puzzle consists of a 9x9 grid divided into nine grids of 3x3. Each of the 3x3 grids contain numbers 1 to 9 arranged in different configurations. Some of the numbers in the 3x3 grids are missing and solving the puzzle involves inserting the missing numbers so that each column, row and 3x3 grid has the numbers 1 to 9, with each number appearing only once in the row, column and 3x3 grid and no missing numbers.

This assignment involves implementing a genetic algorithm to solve the Sudoku problem. Please use the benchmark sets at `http://lipas.uwasa.fi/~timan/sudoku/` to test your program. This site also lists the best known results which you can compare the performance of your program to. Two academic papers on applying GAs to solve Sudoku have also been uploaded alongside Assignment 2 as attachments.

You are required to write a python program to solve the Sudoku problem. You are allowed to use the DEAP `https://deap.readthedocs.io/en/master/examples/index.html` library in Python to code your GA. You can also write your own piece of code without using the recommended library.

2 Submission document/code

You must submit a report (in PDF) specifying:

- Representation used
- Initial population generation
- Fitness evaluation
- Selection method used
- Genetic operators used and the value of the parameters chosen

You must submit a *.py* file named sudoku.py which contains your code. Your *.py* will be tested as follows in the terminal:

sudoku.py puzzle.txt n_gen

,where *puzzle.txt* is a text file containing a sudoku puzzle in the format of the sudoku puzzles in `http://lipas.uwasa.fi/~timan/sudoku/` (eg s01a.txt) and n_gen is the number of generations.

Your program must output the final solves puzzle as a text file (i.e puzzle with best fitness score), the total fitness score of the population for each generation and when it is complete it must output a graph representing the total fitness score per generation.

Total: 50