Practical Visualization of Genetic Algorithms

Hudson Taylor Lekunze[1] and **Olga Chemodanova**[2]

[1]Institute of Computer Science, University of Tartu
[2]Institute of Computer Science, University of Tartu

UNIVERSITY OF TARTU

Introduction

Genetic algorithms are a type of evolutionary algorithms that are based on biologically inspired operators and are used to develop solutions to optimization and search problems (Melanie, 1996). The genetic algorithms are used for solving different problems, such as function optimization, Sudoku solving (Mantere and Koljone, 2006), music generation (Lopez-Rincon, Starostenko, Lopez-Rincon, 2021), and so on.

Objectives

The goal of the project is to provide students of the Institute of Computer Science and everyone interested in the topic of Genetic Algorithms (GAs) with a tool for studying them. The tool is intended for beginners, so everything is written in fairly understandable language. The tool includes a web application part with theoretical information about GAs and an interactive visualization of GA, as well as a python code for the knapsack problem via GA.

Algorithm

GAs Visualization Visualizing GAs is a useful step in understanding how the optimization algorithm behaves, given the different configuration of key parameters such as population size, generations, mutation and crossover rates, and even selection criteria. With increasing data, the cost of optimizing some objective function becomes computationally expensive and it is useful to explore which evolutionary pathways lead to the best results. In this demo, we show how a GA can be visualized as it is being applied to an automated clustering task. Users can configure GA parameters, select out of 8 clustering algorithms, and 2 classes of fitness algorithms (single objective or multi-objective), each with different indices for cluster evaluation.

Knapsack problem via GA To show students that different problems can be solved via GA, the knapsack problem solution is presented. It incorporates the important steps of a GA, such as creating an initial population, calculating the fitness value, selection of solutions, and performing generic operations (namely, crossover and mutation).

Results

The main result of the project is a web tool (Figure 1). It includes theory about GAs and GA interactive visualization. As for the interactive visualization of GAs, with a selection of provided benchmark cluster datasets, a user is able to see how the selection of chromosomes (i.e. hyper-parameter values) over time yields either better or worse outcomes, in the Visualization Panel. At the end of a defined number of generations, the user is also presented with the best results from the optimization process, their respective hyper-parameter values and labeled clusters, which they can save to compare with future runs of the GA with different parameter configurations of fitness function, mutation or crossover rates, amongst others.

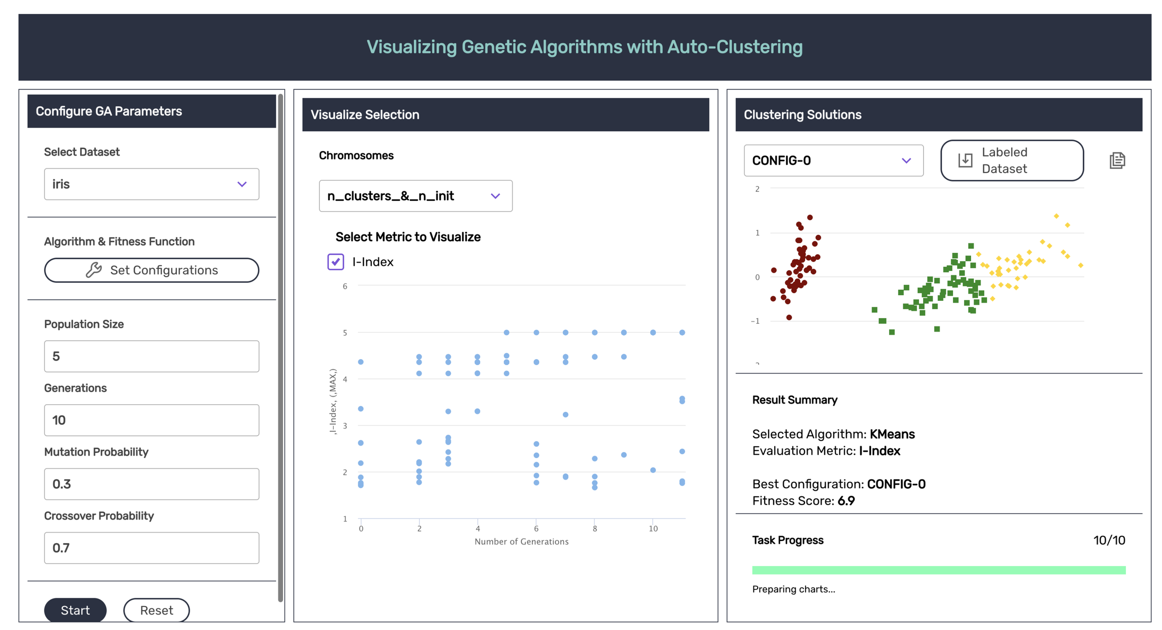

Fig. 1: GA Visualization.

The knapsack problem solution via a GA is implemented in the form of code (with the user's input). Students can receive help with choosing courses to study - which courses to take so that the value for them is maximum and the maximum amount of credits is not exceeded (Figure 2). For that, they should input the maximum amount of credits they can take, as well as the values of each course they are considering to choose (a subjective measure), and the number of credits each course takes. Then, via a GA algorithm, they receive a solution to this problem. The code includes comments so that students can understand the operating principle of the algorithm.

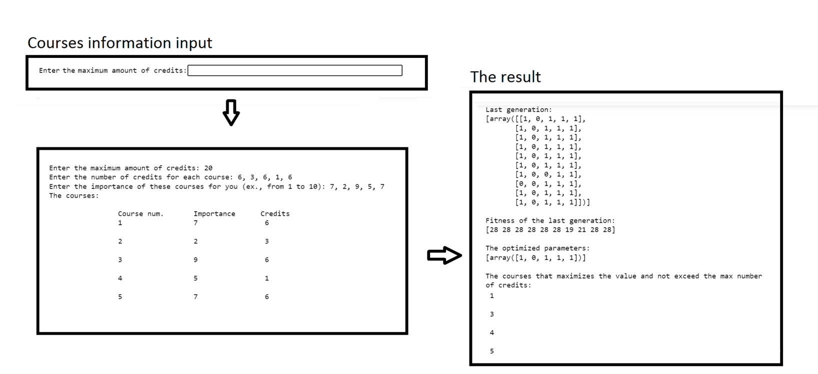

Fig. 2: Knapsack problem with GA.

Future work

The web application is a complete tool to start learning the algorithm. However, it can be improved by incorporating the code of the knapsack problem solution. Another improvement, for instance, can be making the knapsack problem more complex by including parameters other than weight and value. Thus, there are opportunities for further work on this application.

References

Mantere, T., Koljone, J. (2006). Solving and rating sudoku puzzles with genetic algorithms. In New Developments in Artificial Intelligence and the Semantic Web, Proceedings of the 12th Finnish Artificial Intelligence Conference STeP, 86–92.

Melanie, M (1996). An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press. ISBN 9780585030944.

Lopez-Rincon, O., Starostenko, O. and Lopez-Rincon, A., (2021). Algorithmic music generation by harmony recombination with genetic algorithm. Journal of Intelligent Fuzzy Systems, pp.1-13.