Graph Coloring Using Genetic Algorithm

Jakub Kocalka, xkocal00

April 3, 2022

1 Graph Coloring

Graph Coloring is the assignment of colors (labels) to elements of a graph subject to certain constraints. In it's simplest form (and the form that this project deals with), it is the assignment of colors such that two adjacent elements don't have the same color. The Graph Coloring Problem is a decision problem asking whether a coloring exists for a specific graph and a specific number of colors. Furthermore, in optimization mode, it asks what is the smallest number of colors required for a specific graph to be colored.

In this project, I will concern myself with vertex and edge coloring.

2 Genetic Algorithm

Genetic Algorithm is an algorithm belonging to the larger class of evolutionary algorithms. The base building blocks of a genetic algorithm are a population of chromosomes, selection according to fitness, crossover to produce new offspring, and random mutations of new offspring's chromosomes[1].

3 Graph Coloring Using Genetic Algorithm

The form a genotype for this problem should take is clear: a color for each element (vertex or edge). The fitness function is also simple: the reciprocal of the number of element pairs that are in conflict. However, the choice of genetic operators is a little more complicated, at least when it comes to crossover. We can't perform a simple single point cross over, as the genes don't have local dependencies. Instead, we should perform a crossover based on the fenotype.

4 Implementation

I will be using $PyGAD$[2] a genetic algorithm implementation for python[1]. This library presents an easy to use framework for creating and running genetic algorithms, along with the collection of evaluation metrics, such as fitness changes.

[1] https://github.com/ahmedfgad/GeneticAlgorithmPython

It provides a lot of default implementation, but allows the user to customize the genetic operators, parent selection and fitness functions.

I have already implemented a simple version of the algorithm using *PyGAD*. However, I don't have much to present, as it was not tested extensively, and for small graphs tends to guess the correct answer while constructing the initial population.

References

[1] Mellanie Mitchels. *An Introduction to Genetic Algorithms*. 1996. ISBN: 9780585030944.

[2] Ahmed Fawzy Gad. *PyGAD: An Intuitive Genetic Algorithm Python Library*. 2021. arXiv: `2106.06158 [cs.NE]`.