



An investigation into Chromatic Graph Theory and optimal Graph Colouring techniques

Developing and implementing a graph coloring tool and analysing it's performance on a range of generic and special-case graphs

Harry Alexander Richards under the supervision of Professor Daniel Paulusma

A project plan contributing towards a bachelors degree in
Computer Science and Mathematics within the Natural Sciences programme
Bachelor of Science School of Engineering and Computing Sciences

Durham University
23/10/2018

Abstract

In preparation for this project a substantial amount of research has taken place in the field of chromatic graph theory and particularly in the graph colouring problem and gaining an understanding of approaches to solving the graph colouring problem that have already been explored. This will prove useful as one of the project aims to explore known approaches such as the first-fit or brute force colouring, analyse the effectiveness of these approaches on different graph classes. Beyond this I have explored more practical implementations and details such as efficient graph representation and this should prove useful in developing my own graph colouring tool(s).

As the graph colouring problem is a well established, known problem it is not expected that I will be able to produce a tool which works perfectly for all graphs G regardless of the graph class that G belongs to. However, through a series of trial and error and systematic analysis of the results yielded by the tool I expect to be able to produce some relatively low upper bounds for the chromatic number $\chi(G)$ - at least on cases where we are considering a graph $G \in X$ where X is a graph class exhibiting a characteristic we can take advantage of. In cases where the colouring yielded is not near-optimal I expect there will still be a considerable amount of reasoning we can explore behind this and perhaps an opportunity to improve the tool.

Preliminary Preparation

The following are a list of requirements and recommended objectives completed in preparation of beginning the project and beginning development of my graph colouring tool:

- Revise current understanding of graph theory and some of the concepts with chromatic graph theory.
- Develop a further understanding of chromatic graph theory and gain an understanding of the Graph Colouring Problem.
- Investigate a number of specific graph classes and characteristics of these graph classes that make it particular methods of colouring them optimal or near-optimal (in terms of obtaining a lowest upper bound for the chromatic number).
- Due to the substance of this project being in the analysis of the results we obtain and the quality of the results I will be using Python as this is what I'm most familiar with. Should the speed at which results are obtained have been of more importance then I would have used C++ (as I previously alluded to in my Literature Review).
- In order to offer a suitable compromise between ease of implementation and the speed at which my results are acquired I will familiarise myself with techniques of efficient graph representation and methods of implementing my algorithms such that they still run relatively quickly.
- As mentioned earlier I will research a number of basic (usually poor) algorithms for colouring graphs and implement these myself in order to provide a suitable benchmark for comparison after I have developed my algorithm.
- Ensure that I am able to input benchmark graphs from web pages (these pages were linked in my literature review) as this provides me with a far larger pool of graphs at my disposal.
- Gain a deeper understanding of the evolutionary, simulated annealing and ant-based algorithms that I will be implementing and why each of them could prove effective in yielding solutions to the Graph Colouring Problem (at least in some cases). In order to do this it may be useful to analyse

my previous implementations (of the simulated annealing and genetic algorithms for solving the travelling salesman problem) as well as outsourcing example implementations (as referenced in the literature review).

- In addition to the above it will be necessary to familiarise myself with how best to create a hybrid algorithm, since the literature review this now seems likely to be a combination of the genetic algorithm and the ant colony algorithm.

Project Aim (Summary)

Here I will state the project aim clearly and concisely.

I am to produce a graph colouring tool to yield near-optimal solutions to the Graph Colouring problem. This graph colouring tool should take a graph G as an input, find an n -colouring of said graph and output the lowest n for which it is able to find an n -colouring. My hope is that after much work the n outputted will be equal to the chromatic number ($\chi(G) = n$).

As stated above I do not expect the n returned by the tool to satisfy $\chi(G) = n$ for any graph G . However, it is my hope that for some graph classes this will be the case, and regardless of the output that the trends yielded by the graph tool will be of interest to us when performing analysis.

Objectives

The following is an outline of the necessary objectives in order to complete the aim outlined above, having identified the 'level' of the objective in terms of the difficulty of fulfilling the objective(s).

Basic

These are basic objectives that I would expect to be completed in the early stages of the project (early stages).

- Set up a tool in which one can input the benchmark graphs from the web pages in my previous email.
- Implement the first-fit algorithm and see how far the results it yields are away from the optimal solutions when run on the benchmark graphs. First-fit depends on the ordering of the vertices, if we alter the order in which the algorithm colours vertices, for instance, based on vertex degrees it will be useful so that we can compare the results yielded and analyse any differences exhibited.
- Implement a number of other graph colouring heuristics (ones that I find through research and perhaps my own custom heuristics), again I will analyse the effectiveness of these heuristics and the causal factors in poor and near-optimal results.
- Implement a graph colouring checker, which verifies if the output of a colouring algorithm is really a graph colouring (that is, there are no vertices connected by an edge that have the same colour).
- The above ensures that we are able to colour a vast range of graphs, including some special-case graphs that we should be able to input in the same manner as other graphs. However, it provides us with limited progress in solving the Graph Colouring Problem (finding the chromatic number).

Intermediate

These are intermediate objectives that I would expect to be completed in the mid-stages of the project and would consider essential in order for my project aim to be fulfilled. I expect a considerable amount of my time will be spent on fulfilling these (intermediate) objectives. The theme of this section shall be to optimise our graph colouring techniques, as this should provide us with more interesting results which we can analyse in turn, and will hopefully provide some useful insight for our conclusions.

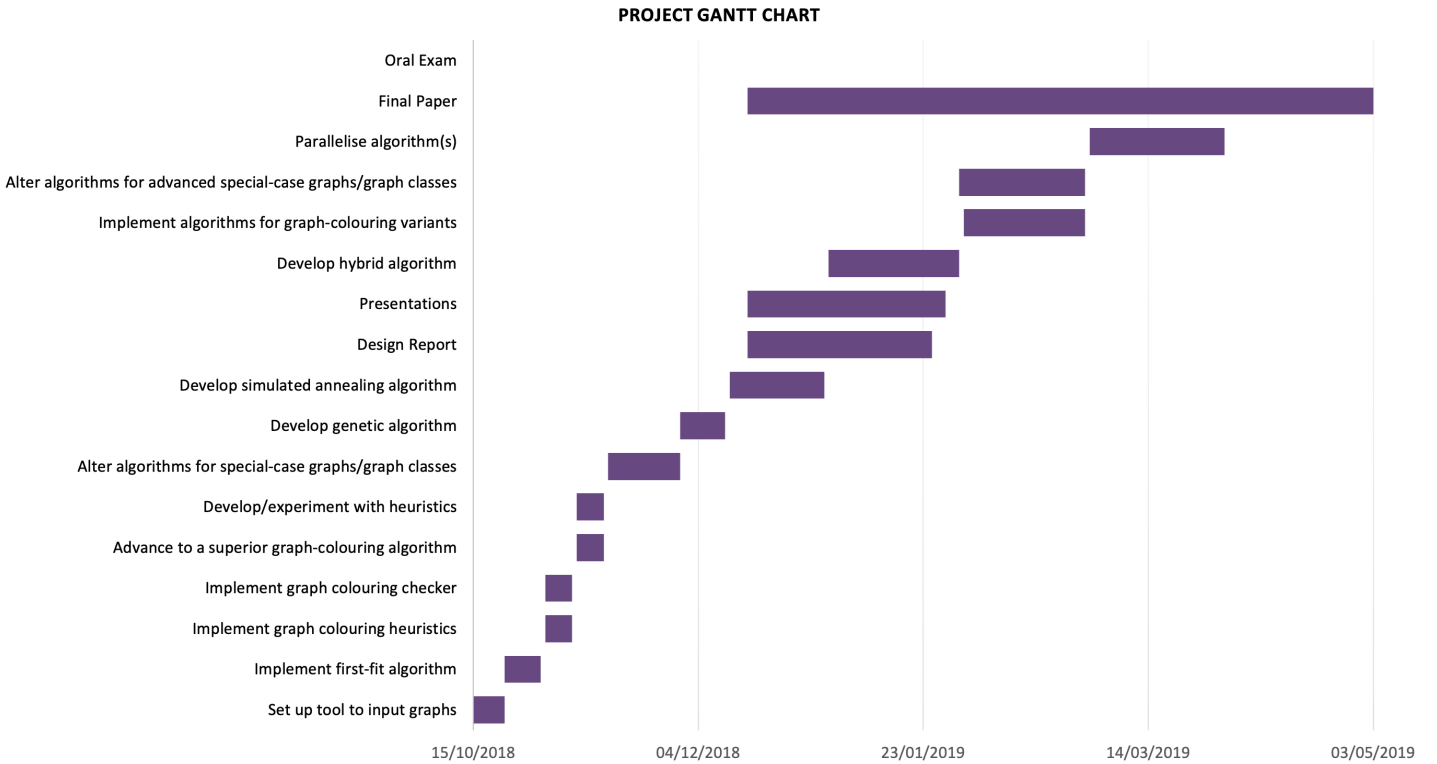
- We must to the best of our ability improve the algorithms/heuristics that we developed as part of our basic objectives, as mentioned about this will help to acquire better results and offer more insight as to how the Graph Colouring Problem is best approached.
- As an extension of the algorithms we will have designed up to this point, develop new algorithms designed with the specific purpose of solving the Graph Colouring Problem for special case graph classes. An example graph class provided by the supervisor was graphs that are almost bipartite and graphs with bounded maximum degree. Our aim in this objective is to use the understanding of graph structures and how this effects our algorithms that we will have developed at this point to exploit properties of graph classes in order to find better/near-optimal colourings.
- With the aim of obtaining better colourings I will be developing a genetic algorithm as well as a simulated annealing algorithm for solving the Graph Colouring Problem - this is inline with the above objectives.
- Proceeding from the above algorithms I aim to develop a further algorithm, which is a hybrid of the genetic algorithm and some other (from previous research and my literature review this is likely to be an ant-based algorithm).

Advanced

These are loose outlines of advanced objectives that I would expect to be completed in the latter stages of the project but will add vastly to the quality of the project as a whole. They are not as fixed as the basic and intermediate objectives and as more of the project is completed we may see some of these altered/replaced.

- I shall look into graph colouring variants and implement algorithms for such variants.
- Parallelise the algorithms or design a new parallel algorithm. The aim behind this is to improve the efficiency of my algorithm(s) and improve the rate at which they work and thus we will be able to yield greater results in the same time.
- There are a number of interesting graph classes and graph colouring variants of interest and currently being studied within the department - this includes a conjecture on a graph colouring variant for planar graphs. It will be interesting to see how my algorithms (after some modifications) would perform on this (or other) graph class(es) and this will add a further focus to the project.

Project Plan



A Gantt chart of expected progression as well as both formal and informal deadlines for the project.

Above we have a reasonably detailed Gantt chart of both university and self-imposed project deadlines. It is not highlighted in the diagram the priority of the deadlines but if necessary see the Objectives section for reference. The project is broken down in the above section to some clear objectives however, there as mentioned previously there is some flexibility to the order of their completion and in fact whether some of the objectives are in time replaced with others that may be more rewarding to pursue. The Gantt chart is a guideline in order to make sure that I am using my time productively and that the project is completed before time is exhausted. The purpose of this is in order to make sure I am managing my time efficiently and meeting regular deadlines.

| GANTT CHART | | | |
|---|------------|---------------------|-----------------|
| OBJECTIVES | START DATE | COMPLETION DURATION | COMPLETION DATE |
| Set up tool to input graphs | 15/10/2018 | 7.00 | 22/10/2018 |
| Implement first-fit algorithm | 22/10/2018 | 8.00 | 30/10/2018 |
| Implement graph colouring heuristics | 31/10/2018 | 6.00 | 06/11/2018 |
| Implement graph colouring checker | 31/10/2018 | 6.00 | 06/11/2018 |
| Advance to a superior graph-colouring algorithm | 07/11/2018 | 6.00 | 13/11/2018 |
| Develop/experiment with heuristics | 07/11/2018 | 6.00 | 13/11/2018 |
| Alter algorithms for special-case graphs/graph classes | 14/11/2018 | 16.00 | 30/11/2018 |
| Develop genetic algorithm | 30/11/2018 | 10.00 | 10/12/2018 |
| Develop simulated annealing algorithm | 11/12/2018 | 21.00 | 01/01/2019 |
| Design Report | 15/12/2018 | 41.00 | 25/01/2019 |
| Presentations | 15/12/2018 | 44.00 | 28/01/2019 |
| Develop hybrid algorithm | 02/01/2019 | 29.00 | 31/01/2019 |
| Implement algorithms for graph-colouring variants | 01/02/2019 | 27.00 | 28/02/2019 |
| Alter algorithms for advanced special-case graphs/graph classes | 31/01/2019 | 28.00 | 28/02/2019 |
| Parallelise algorithm(s) | 01/03/2019 | 30.00 | 31/03/2019 |
| Final Paper | 15/12/2018 | 139.00 | 03/05/2019 |
| Oral Exam | 05/04/2018 | 31.00 | 06/05/2019 |

A detailed table of the objectives and expected dates of completion.