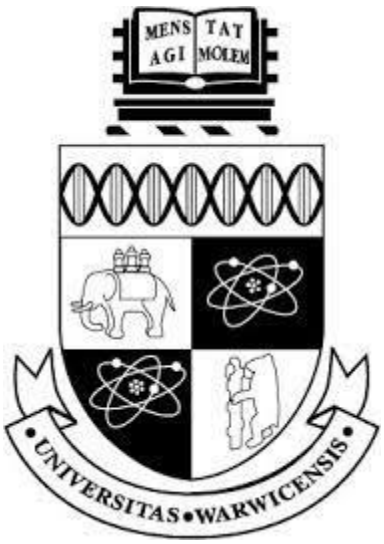
New Developments in Graph Editing Algorithms

Project Specification

Sam Moon [2001779]



Contents

1 Abstract

An investigation into New Developments in Graph Editing Algorithms, studying various algorithms relating to the vertex cover problem, including the König-Egerváry class of graphs, parameterised vertex cover problems and the use of linear programming techniques in efficient algorithms.

2 Project Problem

2.1 Background

In recent years there have been many developments in our understanding of the relationship between Matchings and Vertex Covers; The well-known König-Egerváry Theorem (or simply König's Theorem) [1] asserts that the size of a maximum matching equals the size of a minimum vertex cover in bipartite graphs, and we denote the class of all graphs with this property as König-Egerváry graphs (of which bipartite graphs are a proper subset). More recently there have been further developments including a complete characterisation of such graphs which have a unique minimum vertex cover [2], this characterisation can be efficiently algorithmically checked. It is also known that the Vertex Cover problem is NP-Complete in general graphs however we can more efficiently solve parameterised versions of the problem (known as Fixed-Parameter Tractable or FPT problems) [3]. There have also been advances in using Linear Programming techniques to produce faster parameterised algorithms [4].

2.2 Objective

This project will attempt to produce an implementation of some previously theoretical algorithms in order to showcase them and assess their performance in a practical scenario. More specifically, the following algorithm ideas are proposed in their respective papers:

- (2.2.1a) A complete characterisation of König-Egerváry graphs with unique vertex covers [2]
- (2.2.1b) An FPT algorithm for Local Search Vertex Cover problem on planar graphs [5]
- (2.2.1c) The "Simple Algorithm" for Vertex Cover Above LP [4]
- (2.2.1d) The "Improved Algorithm" for Vertex Cover Above LP [4]
- (2.2.1e) An algorithm for König-Egerváry vertex deletion problem [4]

In addition:

- The program should be efficient enough to run smoothly, assuming the theorised algorithm in question is efficient
- The project will produce a summary of the various algorithms considered
- And a brief review of each, evaluating its strengths and weaknesses

3 Methods

3.1 Research

Before developing anything for the project, it will be necessary to complete significant research, so that a thorough understanding of the topics within the project is gained. Several weeks of research will be conducted, through Google Scholar, books within the department and University library, various academic texts, and online sources. This will ensure that the project is thoroughly considered and minimise the amount of reading to be done later in the project when the focus will be on the project and portfolio itself.

All research notes will be stored online, where they can be accessed and added easily, complete with references – this means if further clarification is required on an area, the original source can be returned to.

3.2 Development

In the development stage of the project, C++ will be used, along with various libraries (SNAP, BGL) to implement different algorithm ideas proposed by the papers above. Each algorithm will be fully implemented and tested before proceeding to the next, this will ensure that if the project is unable to fulfill all objectives set out above then there still exists an implementation and full review of some of the objectives, rather than potentially the implementations completed with no report written about any of them.

3.3 Testing

The various algorithms will be tested rigorously with various graphs, to ensure that it works and is functional. This will also show where some algorithms have weaknesses and could be improved, as they may be optimal for some specific graphs, but be inefficient in other cases. As more research is done into the algorithms, a more concrete test table will be produced for each covering a range of cases including typical and extreme inputs, such as very large graphs or very dense graphs.

In order to test the performance of the implemented algorithms, GNU Profiler, an open-source tool for performance profiling and optimisation, will be used to give accurate and accessible data and to ensure that all tests are carried out in a fair environment so that they are comparable.

4 Timetable

The project timetable will be displayed as a Gantt Chart (Appendix A). The following table shows the hours per week which will be dedicated to Project Work. This approximates the amount of work per week necessary. The progress of the Project will be regularly monitored, both individually, and via weekly meetings with the project supervisor. In the weekly meetings, the progress and direction of the project will be discussed, so as to ensure that it is on track, and any necessary adjustments made, be that to the timescale or the project itself. Therefore, the provisional timetable is subject to change, with further understanding of the processes required, the limitations of the project, and any further research that may be necessary.

In Term 1, less time will be dedicated to the project, due to module selection. Consequently, progress during Term 1 will be less than what was initially hoped to achieve, so it will be necessary to set realistic and achievable goals. Considerable time will be spent during the Christmas break to work on the project, to ensure sufficient progress is made. As there is no specific software or material that is only on Campus or available during Term time requirements, this will be possible and achievable. Term 2 will also have considerably more time dedicated to the project.

Each task will be inserted into Trello.com, a web-based list-making application, which enables you to make lists, and sub-lists, and move items between these. This will be a necessary visual display of the tasks: what is done, what is left, and what is in progress. It will also aid in time management and prioritisation.

Term 1	10	11	12	1	2	3	4	5
Monday			x	x		x	x	
Tuesday	x							
Wednesday						x	x	x
Thursday				x	x	x		
Friday								
Saturday								
Sunday								

5 Risk Assessment

The main risks will be loss of data or inability to work due to a laptop malfunction. To mitigate this, backups of the data will be regularly taken to ensure that any possible issues are minimised – Git will be used for this. This should ensure that there is little work lost due to hardware issues. There are possibilities of theft, fire, and vandalism, so there will be a backup completed somewhere away from where the project is being completed (namely the personal residences of the student completing the project). This backup will be on a computer in the CS department.

There is also the risk of illness, which is incontrollable, so appropriate measures will be in place should this occur, such as giving earlier deadlines than necessary.

6 Resources to be used

The resources that are going to be used are all freely available to use, with prior experience in using some of the following.

- **Git** will be used for version control, to store a backup of the project, and allow it to be worked on from multiple computers, be that at home or on the DCS machines on campus.
- **C++** will be used to implement the algorithms mentioned in the Objectives.
- Two libraries for graph data structures and standard algorithms may be used, depending on the requirements for each algorithm:
    - The **BGL** (Boost Graph Library) may also be used for implementing the relevant algorithms. [8]
    - The **SNAP (**Stanford Network Analysis Platform) will also be used in the project for implementing the algorithms. [9]
- **gProfiler**, an open-source performance profiling, and optimisation tool, will be used in the testing phase for each algorithm. [10s]
- **Computers in the DCS** will be used during the project so that the project can be worked on during breaks between lectures, and spare time on campus. Backups will also be completed on a Computer in DCS to mitigate any setbacks due to any hardware issues on personal devices.
- **Trello.com** will be used to coordinate and work through the requirements of each task in a visual way, monitor progress made, and decompose each section into manageable chunks that are achievable, attainable, and measurable. This ensures that the progress is maintained, and the workload manageable.

7 Legal, Social, Professional, and Ethical Issues

This project does not entail any legal, social, professional, or ethical issues as the product will not involve other people or be released into the public. The project will also use only free licence and/or open-source software to avoid legal issues.

8 References

[1] D. König, "Graphs and matrices" Mat. Lapok, (1931)

[2] V. Raman et al., A characterization of König-Egerváry graphs with extendable vertex covers, Inf Process. Lett. (2020)

[3] Christian Komusiewicz, André Nichterlein, and Rolf Niedermeier, Parameterized Algorithmics for Graph Modification Problems: On Interactions with Heuristics (2016)

[4] D. Lokshtanov, N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh. Faster parameterized algorithms using linear programming. ACM Transactions on Algorithms, (2014)

[5] M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. A. Rosamond, S. Saurabh, and Y. Villanger. Local search: Is brute-force avoidable? Journal of Computer and System Sciences, (2012)

[6] A. Chandra Babu, P.V. Ramakrishnan. New Proofs of Konig-Egervary Theorem and Maximal Flow - Minimal Cut Capacity Theorem using O.R Techniques, (1991)

[7] Vadim E. Levit, Eugen Mandrescu. A Characterisation of Konig-Egervary Graphs Using a Common Property of All Maximum Matchings, (2011)

[8] Boost Graph Library, Accessed 10th October 2022

[9] SNAP Graph Library, Jure Leskovec, Accessed 11th October 2022

[10] gProfiler, an open-source, continuous profiler for production across any environment, at any scale. Accessed 10th October 2022

9 Appendix A

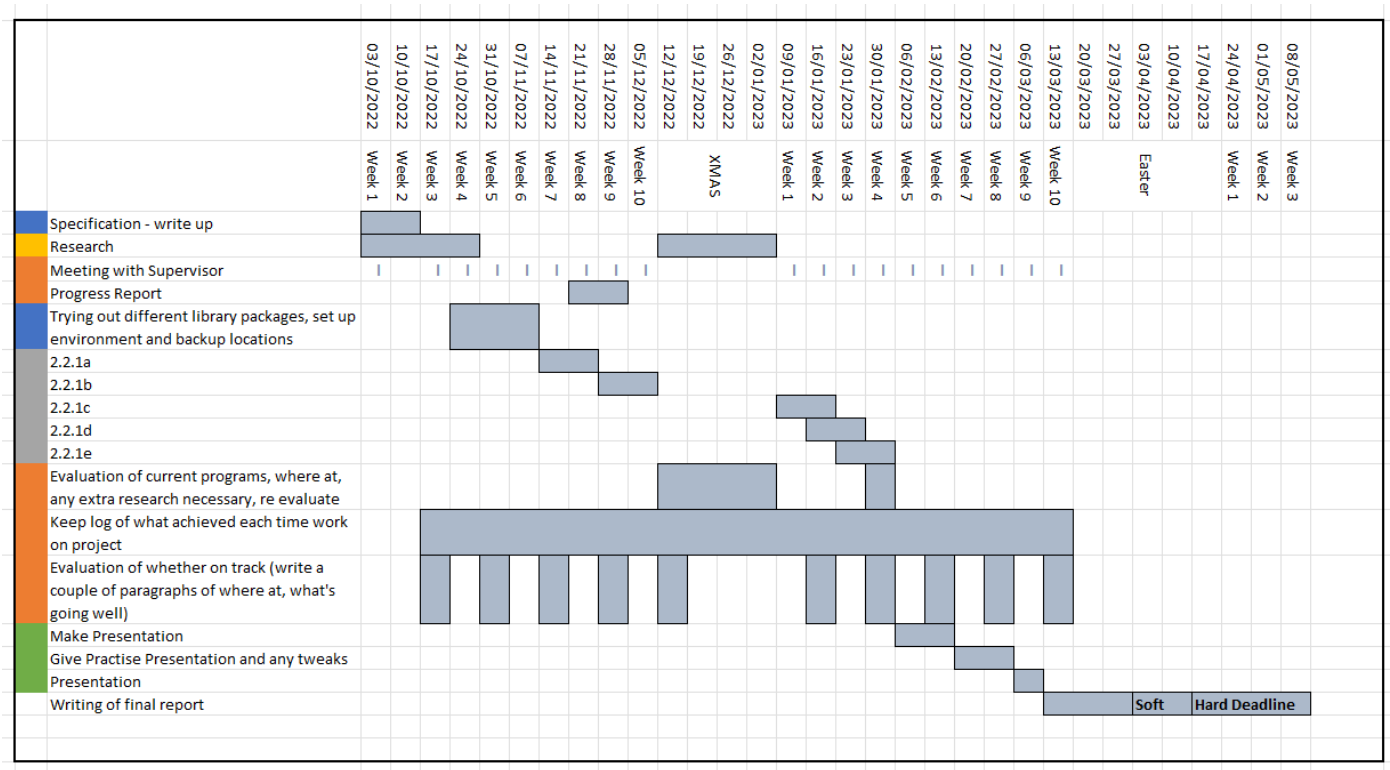


Figure 1: A Gantt Chart of the project's time management