

# Preface

Graph colouring is one of those rare examples in the mathematical sciences of a problem that is very easy to state and visualise, but that has many aspects that are exceptionally difficult to solve. Indeed, it took more than 160 years and the collective efforts of some of the most brilliant minds in nineteenth and twentieth century mathematics just to prove the simple sounding proposition that “four colours are sufficient to properly colour the vertices of a planar graph”.

Ever since the notion of “colouring” graphs was first introduced by Frances Guthrie in the mid-1800s, research into this problem area has focussed mostly on its many theoretical aspects, particularly concerning statements on the chromatic number for specific topologies such as planar graphs, line graphs, random graphs, critical graphs, triangle free graphs, and perfect graphs. Excellent reviews on these matters, together with a comprehensive list of open problems in the field of graph colouring, can be found in the books of Jensen and Toft (1994) and Beineke and Wilson (2015).

In this book, our aim is to examine graph colouring as an *algorithmic* problem, with a strong emphasis on practical applications. In particular, we take some time to describe and analyse some of the best-known algorithms for colouring arbitrary graphs and focus on issues such as (a) whether these algorithms are able to provide optimal solutions in some cases, (b) how they perform on graphs where the chromatic number is unknown, and (c) whether they are able to produce better solutions than other algorithms for certain types of graphs, and why.

This book also devotes a lot of effort into looking at many of the real-world operational research problems that can be tackled using graph colouring techniques. These include the seemingly disparate problem areas of producing sports schedules, solving Sudoku puzzles, checking for short circuits on printed circuit boards, assigning taxis to customer requests, timetabling lectures at a university, finding good seating plans for guests at a wedding, and assigning computer programming variables to computer registers.

This book is written with the presumption that the reader has no previous experience in graph colouring, or graph theory more generally. However, an elementary knowledge of the notation and concepts surrounding sets, matrices, and enumerative

combinatorics (particularly combinations and permutations) is assumed. The initial sections of Chapter 1 are kept deliberately light, giving a brief tour of the graph colouring problem using minimal jargon and plenty of illustrated examples. Later sections of this chapter then go on to look at the problem from an algorithmic point of view, looking particularly at why this problem is considered “intractable” in the general case, helping to set the ground for the remaining chapters.

Chapter 2 of this book looks at three different well-established constructive algorithms for the graph colouring problem, namely the GREEDY, DSATUR, and RLF algorithms. The various features of these algorithms are analysed and their performance (in terms of running times and solution quality) is then compared across a large set of problem instances. A number of bounds on the chromatic number are also stated and proved.

Chapters 3 and 4 then go on to look at some of the best-known algorithms for the general graph colouring problem. Chapter 3 presents more of an overview of this area and highlights many of the techniques that can be used for the problem, including backtracking algorithms, integer programming methods, and metaheuristics. Ways in which problem sizes can be reduced are also considered. Chapter 4 then goes on to give an in-depth analysis of six such algorithms, describing their relevant features, and comparing their performance on a wide range of different graph types. Portions of this chapter are based on the research originally published by Lewis et al. (2012).

Chapter 5 considers a number of example problems, both theoretical and practical, that can be expressed using graph colouring principles. Initial sections focus on special cases of the graph colouring problem, including map colouring (together with a history of the Four Colour Theorem), edge colouring, and solving Latin squares and Sudoku puzzles. The problems of colouring graphs where only limited information about a graph is known, or where a graph is subject to change over time, are also considered, as are some natural extensions to graph colouring such as list colouring, equitable graph colouring and weighted graph colouring.

The final three chapters of this book look at three separate case studies in which graph colouring algorithms have been used to solve real-world practical problems, namely the design of seating plans for large gatherings, creating schedules for sports competitions (Lewis and Thompson, 2010), and timetabling events at educational establishments (Lewis and Thompson, 2015). These three chapters are written so that, to a large extent, they can be read independently of the other chapters of this book, though obviously a full understanding of their content will only follow by referring to the relevant sections as instructed by the text.

## **A Note on Pseudocode and Notation**

While many of the algorithms featured in this book are described within the main text, others are more conveniently defined using pseudocode. The benefit of pseudocode is that it enables readers to concentrate on the algorithmic process without

worrying about the syntactic details of any particular programming language. Our pseudocode style is based on that of the seminal textbook *Introduction to Algorithms* by Cormen, Leiserson, Rivest and Stein, often simply known as the “The Big Book of Algorithms” (Cormen et al., 2009). This particular pseudocode style makes use of all the usual programming constructs such as while-loops, for-loops, if-else statements, break statements, and so on, with indentation being used to indicate their scope. To avoid confusion, different symbols are also used for assignment and equality operators. For assignment, a left arrow ( $\leftarrow$ ) is used. So, for example, the statement  $x \leftarrow 10$  should be read as “ $x$  becomes equal to 10”, or “let  $x$  be equal to 10”. On the other hand, an equals symbol is used only for equality *testing*; hence a statement such as  $x = 10$  will only evaluate to true or false ( $x$  is either equal to 10, or it is not).

All other notation used within this book is defined as and when the necessary concepts arise. Throughout the text, the notation  $G = (V, E)$  is used to denote a graph  $G$  comprising a “vertex set”  $V$  and an “edge set”  $E$ . The number of vertices and edges in a graph are denoted by  $n$  and  $m$  respectively. The colour of a particular vertex  $v \in V$  is written  $c(v)$ , while a candidate solution to a graph colouring problem is usually defined as a partition of the vertices into  $k$  subsets  $\mathcal{S} = \{S_1, S_2, \dots, S_k\}$ . Further details can be found in the various definitions within Chapters 1 and 2.

## Additional Resources

This book is accompanied by a suite of nine graph colouring algorithms that can be downloaded from [www.rhydLewis.eu/resources/gCol.zip](http://www.rhydLewis.eu/resources/gCol.zip). Each of these heuristic-based algorithms are analysed in detail in the text and are also compared and contrasted empirically through extensive experimentation. These implementations are written in C++ and have been successfully compiled on a number of different compilers and platforms. (See Appendix A.1 for further details.) Readers are invited to experiment with these algorithms as they make their way through this book. Any queries should be addressed to the author.

In addition to this suite, this book’s appendix also contains information on how graph colouring problems might be solved using commercial linear programming software and also via the free mathematical software Sage. Finally, an online implementation of the table planning algorithm presented in Chapter 6 can also be accessed at [www.weddingseatplanner.com](http://www.weddingseatplanner.com).

Cardiff University, Wales.  
August 2015

*Rhyd Lewis*



<http://www.springer.com/978-3-319-25728-0>

A Guide to Graph Colouring  
Algorithms and Applications

Lewis, R.M.R.

2016, XIV, 253 p., Hardcover

ISBN: 978-3-319-25728-0