


---

# Texts in Computer Science

## Series Editors

David Gries, Department of Computer Science, Cornell University, Ithaca, NY, USA

Orit Hazzan , Faculty of Education in Technology and Science, Technion—Israel Institute of Technology, Haifa, Israel

More information about this series at <http://www.springer.com/series/3191>

---

R. M. R. Lewis

# Guide to Graph Colouring

Algorithms and Applications

Second Edition

R. M. R. Lewis  
Cardiff School of Mathematics  
Cardiff University  
Cardiff, UK

ISSN 1868-0941

ISSN 1868-095X (electronic)

Texts in Computer Science

ISBN 978-3-030-81053-5

ISBN 978-3-030-81054-2 (eBook)

<https://doi.org/10.1007/978-3-030-81054-2>

1<sup>st</sup> edition: © Springer International Publishing Switzerland 2016

2<sup>nd</sup> edition: © The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG  
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*For Fifi, Maiwen, Aoibh, and Maccy*  
*Gyda cariad*

---

## Preface

Graph colouring is one of those rare examples in the mathematical sciences of a problem that, while easy to state and visualise, has many aspects that are exceptionally difficult to solve.

In this book, our goal is to examine graph colouring as an *algorithmic* problem, with a strong emphasis on implementation and practical application. To these ends, in addition to providing a theoretical treatment of the problem, we also dedicate individual chapters to real-world problems that can be tackled via graph colouring techniques. These include designing sports schedules, solving Sudoku puzzles, timetabling lectures at a university, and creating seating plans.

Portable source code for all of the algorithms considered in this book is also available for free download.

---

## Organisation and Features

The first chapter of this book is kept deliberately light; it gives a brief tour of the graph colouring problem, avoids jargon, and gives plenty of illustrated examples.

In Chap. 2, we discover that graph colouring is a type of “intractable” problem, meaning that it usually needs to be tackled using inexact heuristic algorithms. To reach this conclusion, we introduce the topics of problem complexity, polynomial transformations, and  $\mathcal{NP}$ -completeness. We also review several graph topologies that are easy to colour optimally.

Chapter 3 of this book starts with some theory and uses various techniques to derive bounds on the chromatic number of graphs. It then looks at five well-established constructive heuristics for graph colouring (including the Greedy, DSatur, and RLF algorithms) and analyses their relative performance. Source code for these algorithms is provided.

The intention of Chap. 4 is to give the reader an overview of the different strategies available for graph colouring, including both exact and heuristic methods. Techniques considered include backtracking, integer programming, column generation, and various metaheuristics. No prior knowledge of these techniques is assumed. We also describe ways in which graph colouring problems can be reduced in size and broken into smaller parts, helping to improve algorithm performance.

Chapter 5 gives an in-depth analysis of six high-performance algorithms for the graph colouring problem. The performance of these algorithms is compared using several different problem types, including random, flat, scale-free, planar, and timetabling graphs. Source code for each of these algorithms is also provided.

Chapter 6 considers several problems, both theoretical and practical, that can be expressed through 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, Latin squares, and Sudoku puzzles. The problems of colouring graphs where only limited information about connectivity 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, weighted graph colouring, and chromatic polynomials.

The final three chapters of this book examine three separate case studies in which graph colouring techniques can be used to find high-quality solutions to real-world problems. Chapter 7 looks at the problem of designing seating plans for large gatherings; Chap. 8 considers the creation of league schedules for sports competitions; Chap. 9 looks at timetabling in educational establishments. Each of these chapters is written so that, to a large extent, they can be read independently of the other chapters of this book.

---

## Audience

This book is written for anyone with a background in mathematics, computer science, operational research, or management science. Initial sections are particularly appropriate for undergraduate learning and teaching; later sections are more suited for postgraduate and research levels.

This text has been written with the presumption that the reader has no previous experience of graph colouring or graph theory more generally. However, elementary knowledge of the notation and concepts surrounding sets, matrices, and enumerative combinatorics (particularly combinations and permutations) is assumed.

---

## Supplementary Resources

All algorithms reviewed and tested in this book are available for free download at <http://www.rhysdewis.eu/resources/gCol.zip>. These implementations are written in C++ and can be compiled on Windows, macOS, and Linux. Full instructions on how to do this are provided in Appendix A. Readers are invited to experiment with these algorithms as they make their way through this book.

This book also shows how graph colouring problems can be generated and tackled using Sage and Python. Both of these programming languages are free to download. We also show how graph colouring problems can be solved via linear programming software, in this case using the commercial software FICO Xpress.

---

An online implementation of the seat-planning algorithm presented in Chap. 7 can be accessed at <http://www.weddingseatplanner.com>. C++ code for the algorithms described in Chaps. 8 and 9 can also be downloaded using the links given in the text.

I hope you find these implementations as useful as I did.

Cardiff, Wales, UK  
June 2021

R. M. R. Lewis



---

# Contents

<b>1</b>	<b>Introduction to Graph Colouring</b>	<b>1</b>
1.1	Some Simple Practical Applications	2
1.1.1	A Team Building Exercise	2
1.1.2	Constructing Timetables	4
1.1.3	Scheduling Taxis	5
1.1.4	Compiler Register Allocation	6
1.2	Why Colouring?	7
1.3	Problem Description	9
1.4	About This Book	12
1.4.1	Algorithm Implementations	13
1.5	A Note on Pseudocode and Notation	15
1.6	Chapter Summary	16
	References	16
<b>2</b>	<b>Problem Complexity</b>	<b>17</b>
2.1	Algorithm Complexity and Big O Notation	18
2.2	Solving Graph Colouring via Exhaustive Search	20
2.3	Problem Intractability	23
2.3.1	$\mathcal{P}$ and $\mathcal{NP}$	23
2.3.2	Polynomial-Time Reductions	26
2.3.3	$\mathcal{NP}$ -Completeness	27
2.3.4	Boolean Satisfiability Problems (SAT)	28
2.4	Proofs of $\mathcal{NP}$ -Completeness	30
2.5	Graphs that are Easy to Colour Optimally	33
2.5.1	Complete Graphs	34
2.5.2	Cycle, Wheel, and Planar Graphs	35
2.5.3	Grid Graphs	36
2.6	Chapter Summary and Further Reading	37
	References	38
<b>3</b>	<b>Bounds and Constructive Heuristics</b>	<b>39</b>
3.1	The Greedy Algorithm for Graph Colouring	41
3.2	Bounds on the Chromatic Number	45

3.2.1	Lower Bounds . . . . .	45
3.2.2	Upper Bounds . . . . .	49
3.3	The DSATUR Algorithm . . . . .	54
3.4	Colouring Using Maximal Independent Sets . . . . .	58
3.4.1	The RLF Algorithm . . . . .	59
3.5	Empirical Comparison . . . . .	62
3.5.1	Experimental Considerations . . . . .	63
3.5.2	Algorithm Complexities . . . . .	65
3.5.3	Results and Analysis . . . . .	68
3.6	Chapter Summary and Further Reading . . . . .	71
	References . . . . .	75
<b>4</b>	<b>Advanced Techniques for Graph Colouring . . . . .</b>	<b>77</b>
4.1	Exact Algorithms . . . . .	77
4.1.1	Backtracking Approaches . . . . .	78
4.1.2	Integer Programming . . . . .	81
4.1.3	Minimum Coverings and Column Generation . . . . .	91
4.2	Inexact Heuristics and Metaheuristics . . . . .	96
4.2.1	Feasible-Only Solution Spaces . . . . .	97
4.2.2	Spaces of Complete, Improper $k$ -Colourings . . . . .	102
4.2.3	Spaces of Partial, Proper $k$ -Colourings . . . . .	105
4.2.4	Combining Solution Spaces . . . . .	106
4.2.5	Problems Related to Graph Colouring . . . . .	106
4.3	Reducing Problem Size . . . . .	107
4.3.1	Removing Vertices and Splitting Graphs . . . . .	107
4.3.2	Extracting Independent Sets . . . . .	108
4.4	Chapter Summary . . . . .	110
	References . . . . .	110
<b>5</b>	<b>Algorithm Case Studies . . . . .</b>	<b>113</b>
5.1	The TABUCOL Algorithm . . . . .	113
5.2	The PARTIALCOL Algorithm . . . . .	115
5.3	The Hybrid Evolutionary Algorithm (HEA) . . . . .	117
5.4	The ANTCOL Algorithm . . . . .	118
5.5	The Hill-Climbing (HC) Algorithm . . . . .	121
5.6	The Backtracking Algorithm . . . . .	122
5.7	Algorithm Comparison . . . . .	124
5.7.1	Random Graphs . . . . .	125
5.7.2	Flat Graphs . . . . .	126
5.7.3	Planar Graphs . . . . .	129
5.7.4	Scale-Free Graphs . . . . .	134
5.7.5	Exam Timetabling Graphs . . . . .	137
5.7.6	Social Network Graphs . . . . .	141
5.7.7	Comparison Discussion . . . . .	143

5.8	Further Improvements to the HEA . . . . .	145
5.8.1	Maintaining Diversity . . . . .	146
5.8.2	Recombination . . . . .	149
5.8.3	Local Search . . . . .	151
5.9	Chapter Summary and Further Reading . . . . .	152
	References . . . . .	153
<b>6</b>	<b>Applications and Extensions . . . . .</b>	<b>155</b>
6.1	Face Colouring . . . . .	156
6.1.1	Dual Graphs, Colouring Maps, and the Four Colour Theorem . . . . .	158
6.1.2	Four Colours Suffice . . . . .	163
6.2	Edge Colouring . . . . .	166
6.3	Precolouring . . . . .	171
6.4	Latin Squares and Sudoku Puzzles . . . . .	172
6.4.1	Relationship to Graph Colouring . . . . .	173
6.4.2	Solving Sudoku Puzzles . . . . .	174
6.5	Short Circuit Testing . . . . .	179
6.6	Graph Colouring with Incomplete Information . . . . .	182
6.6.1	Decentralised Graph Colouring . . . . .	182
6.6.2	Online Graph Colouring . . . . .	185
6.6.3	Dynamic Graph Colouring . . . . .	187
6.7	List Colouring . . . . .	188
6.8	Equitable Graph Colouring . . . . .	189
6.9	Weighted Graph Colouring . . . . .	193
6.9.1	Weighted Vertices . . . . .	193
6.9.2	Weighted Edges . . . . .	195
6.9.3	Multicolouring . . . . .	196
6.10	Chromatic Polynomials . . . . .	196
6.11	Chapter Summary . . . . .	200
	References . . . . .	200
<b>7</b>	<b>Designing Seating Plans . . . . .</b>	<b>203</b>
7.1	Problem Background . . . . .	203
7.1.1	Relation to Graph Problems . . . . .	205
7.1.2	Chapter Outline . . . . .	206
7.2	Problem Definition . . . . .	206
7.2.1	Objective Functions . . . . .	207
7.2.2	Problem Intractability . . . . .	208
7.3	Problem Interpretation and Tabu Search Algorithm . . . . .	208
7.3.1	Stage 1 . . . . .	209
7.3.2	Stage 2 . . . . .	210
7.4	Algorithm Performance . . . . .	212
7.5	Comparison to an IP Model . . . . .	214

7.5.1	Results	217
7.6	Chapter Summary and Discussion	219
	References	220
<b>8</b>	<b>Designing Sports Leagues</b>	<b>221</b>
8.1	Problem Background	221
8.1.1	Further Round-Robin Constraints	223
8.1.2	Chapter Outline	225
8.2	Representing Round-Robins as Graph Colouring Problems	225
8.3	Generating Valid Round-Robin Schedules	226
8.4	Extending the Graph Colouring Model	227
8.5	Exploring the Space of Round-Robins	233
8.6	Case Study: Welsh Premiership Rugby	236
8.6.1	Solution Methods	237
8.7	Chapter Summary and Discussion	244
	References	245
<b>9</b>	<b>Designing University Timetables</b>	<b>247</b>
9.1	Problem Background	247
9.1.1	Designing and Comparing Algorithms	249
9.1.2	Chapter Outline	250
9.2	Problem Definition and Preprocessing	251
9.2.1	Soft Constraints	254
9.2.2	Problem Complexity	255
9.2.3	Evaluation and Benchmarking	255
9.3	Previous Approaches to This Problem	256
9.4	Algorithm Description: Stage One	257
9.4.1	Results	259
9.5	Algorithm Description: Stage Two	261
9.5.1	SA Cooling Scheme	261
9.5.2	Neighbourhood Operators	262
9.5.3	Dummy Rooms	264
9.5.4	Estimating Solution Space Connectivity	265
9.6	Experimental Results	266
9.6.1	Influence of the Neighbourhood Operators	266
9.6.2	Comparison to Published Results	269
9.6.3	Differing Time Limits	271
9.7	Chapter Summary and Discussion	271
	References	274
	<b>Appendix A: Computing Resources</b>	<b>277</b>
	<b>Appendix B: Table of Notation</b>	<b>297</b>
	<b>Index</b>	<b>299</b>