

On the Maximum Degree Chromatic Number of a Graph

Isabelle Nieuwoudt



Dissertation presented for the degree of
Doctor of Philosophy
in the Department of Mathematical Sciences
at Stellenbosch University

Promoter: Prof JH van Vuuren
Co-promoter: Dr PJP Grobler

December 2007

Declaration

I, the undersigned, hereby declare that the work contained in this dissertation is my own original work and that I have not previously in its entirety or in part submitted it at any university for a degree.

Signature: _____

Date: _____

Abstract

Determining the (classical) chromatic number of a graph (*i.e.* finding the smallest number of colours with which the vertices of a graph may be coloured so that no two adjacent vertices receive the same colour) is a well known combinatorial optimization problem and is widely encountered in scheduling problems. Since the late 1960s the notion of the chromatic number has been generalized in several ways by relaxing the restriction of independence of the colour classes. If this independence restriction is relaxed so that no colour class induced subgraph may have a maximum degree exceeding some specified number $d \in \mathbb{N}_0$, then the notion of a *maximum degree colouring* emerges. The minimum number of colours with which a graph G may be coloured in this way is the $\Delta(d)$ -chromatic number of G , denoted by $\chi_d^\Delta(G)$. Maximum degree colourings also arise in scheduling applications, where some threshold of conflict between different user groups of a shared resource may be tolerated.

In this dissertation both analytic and algorithmic approaches towards determining $\Delta(d)$ -chromatic numbers, or at least establishing upper bounds on $\chi_d^\Delta(G)$, are adopted. In the case of analytic approaches, the value of $\chi_d^\Delta(G)$ is determined for certain graph classes and an arbitrary value of d . An inverted strategy towards determining the $\Delta(d)$ -chromatic number of a graph is also established. This strategy is inverted in the sense that the number of colours, x say, is fixed and an attempt is then made to minimize the maximum degree of the colour class induced subgraphs. Values for this smallest maximum degree of a graph G , denoted by $D_x^\Delta(G)$, are also established analytically for various graph classes.

Algorithmic approaches towards computing the $\Delta(d)$ -chromatic number of a given graph for a given value of d , may be divided into two groups, namely those for general graphs and those exploiting for certain graph structures. In the latter case the inverted strategy described above is employed to determine an upper bound on $\chi_d^\Delta(G)$ for the class of complete balanced multipartite graphs. Algorithms for general graphs may be subclassified into two further categories: exact algorithms and heuristics. Two new exact algorithms and two new heuristics for the computation of $\chi_d^\Delta(G)$ (or bounds on this parameter) are developed and tested numerically in this dissertation.

The maximum degree chromatic sequence ($\chi_d^\Delta(G) : d \in \mathbb{N}_0$) of a general graph G (*i.e.* the values of $\chi_d^\Delta(G)$ as the parameter d increases) is also investigated. An open problem in maximum degree colourings is the characterization of such sequences (*i.e.* to determine which integral sequences are, in fact, maximum degree chromatic sequences of graphs). Although the characterization of maximum degree chromatic sequences is far from being resolved, the problem is placed on a firm mathematical foundation and the reader is provided with an idea of the difficulties and subtleties surrounding this problem.

Opsomming

Die bepaling van die (klassieke) chromatiese getal van 'n grafiek (naamlik die kleinste aantal kleure waarmee die punte van 'n grafiek gekleur kan word sodat geen twee naasliggende punte dieselfde kleur ontvang nie) is 'n bekende kombinatoriese optimeringsprobleem wat wyd in skeduleringstoepassings teëgekom word. Sedert die laat 1960s is die definisie van die chromatiese getal op verskeie maniere veralgemeen deur die vereiste van onafhanklikheid van die kleurklasse te verslap. Indien hierdie onafhanklikheidsbeperking verslap word sodat die maksimum graad van die kleurklas-geïnduseerde deelgrafieke nie 'n sekere waarde $d \in \mathbb{N}_0$ oorskry nie, word 'n sogenaamde *maksimale graad kleuring* verkry. Die kleinste aantal kleure waarmee die punte van 'n grafiek G op hierdie wyse gekleur kan word, staan as die $\Delta(d)$ -*chromatiese getal* van G bekend, en word deur $\chi_d^\Delta(G)$ aangedui. Maksimale graad kleurings speel 'n belangrike rol in skeduleringstoepassings waar een of ander drumpel van konflik tussen die lede van verskillende gebruikersgroepe van 'n gedeelde hulpbron aanvaarbaar is.

In hierdie proefskrif word beide analitiese en algoritmiese benaderings tot die bepaling van $\Delta(d)$ -chromatiese getalle, of minstens die daarstelling van bogrense op $\chi_d^\Delta(G)$ gevolg. In die geval van analitiese benaderings word die waarde van $\chi_d^\Delta(G)$ vir sekere klasse grafieke en arbitrêre waardes van d bepaal. 'n Inverse strategie tot die daarstelling van $\Delta(d)$ -chromatiese getalle word ook ontwikkel. Volgens hierdie strategie word die aantal kleure, x sê, vasgemaak, en word die maksimum graad oor al die kleurklas-geïnduseerde deelgrafieke geminimeer. Waardes van hierdie kleinste maksimum graad, aangedui deur $D_x^\Delta(G)$, word ook vir verskillende klasse grafieke analities bepaal.

Algoritmiese benaderings tot die berekening van die $\Delta(d)$ -chromatiese getal van 'n gegewe grafiek en 'n waarde van d val in twee klasse, naamlik algoritmes vir algemene grafieke en algoritmes geskoei op sekere klasse grafieke. Die inverse kleuringstrategie hierbo beskryf, word in die laasgenoemde algoritmiese klas gebruik om 'n bogrens op $\chi_d^\Delta(G)$ vir die klas van volledige, gebalanseerde veelledige grafieke te bereken. Algoritmes vir algemene grafieke kan in twee verdere klasse onderverdeel word, naamlik eksakte algoritmes en heuristieke. Twee nuwe eksakte algoritmes en twee nuwe heuristieke word in hierdie proefskrif vir die bepaling van $\chi_d^\Delta(G)$ (of bogrense daarop) vir algemene grafieke daargestel en numeries getoets.

Die maksimale graad chromatiese ry ($\chi_d^\Delta(G) : d \in \mathbb{N}_0$) van 'n algemene grafiek G (naamlik die waardes van $\chi_d^\Delta(G)$ soos die parameter d toeneem) word ook ondersoek. Die karakterisering van hierdie rye (met ander woorde om te bepaal watter heeltallige rye maksimale graad chromatiese rye van grafieke is) is 'n oop probleem in die literatuur oor maksimale graad kleurings. Alhoewel hierdie probleem onopgelos bly, word dit op 'n goeie wiskundige grondslag geplaas en word die leser 'n idee gegee van die komplekse en subtiele aard van die probleem.

Acknowledgements

During the course of most studies there are a number of people who play a significant role throughout the particular study and who makes it possible for the particular student to complete the necessary research. This study is no exception and the author wishes to express her deepest gratitude towards

- Prof JH van Vuuren, for his insight, brilliant ideas and constant motivation during the frustrating periods of this dissertation. It was a truly great privilege to study under his guidance, and to learn from his experience and general mathematical knowledge;
- Dr PJP Grobler, for his suggestions and interest;
- Dr AP Burger, for his many programming ideas, his support and interest in this dissertation;
- the National Research Foundation and the post-graduate bursary office of Stellenbosch University for financial support;
- my friend, Dr Werner Gründlingh, for the excellent graphics in this dissertation and his help with technical difficulties in \LaTeX ;
- fellow PhD-student Frank Ortmann for his help with the 3D plots in Chapter 5 and last minute \LaTeX problems as well as his support throughout the last two years of this study;
- students Ben Krige and Roxy Adams for their great help with computer programming implementations in Java and Mathematica, respectively;
- other friends, family members and fellow graduate students and colleagues at the Applied Mathematics Division at Stellenbosch University for their support; especially my parents for their continuous interest in my studies through many years; my mother for her proofreading and help; my husband who stayed up with me through seemingly endless nights; and friends Desmaré and Cecilia for their constant support and motivation through the difficulties of combining work, study and motherhood.

Dedicated to Dawid,

the light of my life, who had to share his mom for the first four years of his life with this study.

Table of Contents

List of Figures	v
List of Tables	ix
List of Algorithms	xi
Reserved Symbols	xiii
Glossary	xvii
1 Introduction	1
1.1 Map Colouring: The Origin of Graph Colouring	1
1.2 From Maps to Graphs: Other Types of Graph Colouring	4
1.3 Graph Colouring Applications	5
1.4 The Problem to be Considered in this Dissertation	7
1.5 Dissertation Objectives	8
1.6 Dissertation Layout	8
2 Fundamental Concepts and Notation	11
2.1 Basic Definitions in Graph Theory	11
2.2 Complexity of Graph Algorithms	19
2.3 Basic Concepts in Graph Colouring	23
2.4 Generalized Colourings	45
2.4.1 Requirements and Bounds for Generalized Colourings	45
2.4.2 Path Colourings	48
2.4.3 Clique Colourings	49
2.4.4 Chromatic Sequences	50
2.5 Chapter Summary	51
3 Maximum Degree Colourings	53
3.1 Maximum Degree Colourings in Graphs	53
3.2 Literature Survey on Maximum Degree Colourings	55

3.3	The parameters χ_d^Δ and D_x^Δ	60
3.4	Chapter Summary	61
4	Simple graph structure classes	63
4.1	Bipartite Graphs	63
4.2	Cycles and Wheels	64
4.3	Complete Graphs	66
4.4	Products of Paths, Cycles and Complete Graphs	67
4.5	Circulants	76
4.6	Complete Balanced Multipartite Graphs	81
4.6.1	Normalized k -partite Colourings	82
4.6.2	Discrete Colourings for $K_{k \times n}$	89
4.7	Chapter Summary	93
5	$\Delta(d, x)$-Colouring Algorithms	95
5.1	Heuristic Methods	95
5.1.1	A Procedure Based on Brelaz's Heuristic for Proper Colourings	95
5.1.2	A Tabu Search Approach	97
5.1.3	Results Obtained by the $\Delta(d, x)$ -colouring Heuristics	103
5.2	Exact Methods	124
5.2.1	The Irredundant χ_d^Δ -colouring Algorithm	125
5.2.2	The Critical χ_d^Δ -colouring Algorithm	128
5.2.3	Results Obtained by the χ_d^Δ -colouring Algorithms	132
5.3	Chapter Summary	142
6	Maximum Degree Chromatic Sequences	145
6.1	Basic Sequences	145
6.2	Δ -Chromatic Basic Sequences	148
6.3	The Basic Sequence 444422222221	152
6.3.1	On the Order of a Graph with Δ -chromatic Sequence 444422222221	152
6.3.2	On the Size of a Graph with Δ -chromatic Sequence 444422222221	153
6.4	In Search of Graphs realizing Δ -Chromatic Basic Sequences	155
6.4.1	The Product K_n^r	155
6.4.2	The Product $K_{k \times n} \times K_4$	157
6.4.3	Circulants	159
6.4.4	Other constructions	165
6.5	Chapter Summary	166

7 Conclusion	167
7.1 Dissertation Summary	167
7.2 Further Work	169
References	173
Appendices	179
A The Four-colour Theorem for Map Colouring	179
A.1 Kempe’s Attempted Proof of the Four-colour Theorem	179
A.2 The Error in Kempe’s Proof of the Four-colour Theorem	180
B The Five-colour Theorem	183
B.1 Heawood’s Cartographic Proof of the 5-colour Theorem	183
B.2 Heawood’s Graph Theoretic Proof of the 5-colour Theorem	184
C Sequences Satisfying the Conditions of Theorem 3.13	187
D On the Placement of Strings in a Collection of Hats	193
D.1 Article Problem Description	193
D.2 Upper and Lower Bounds on the Parameter $\alpha(x, k)$	194
D.3 A Recursive Algorithmic Solution Approach	196
E The Tabu Search Methodology	199
F Benchmark Instances for the $\Delta(d, x)$-Colouring Algorithms	201
F.1 Certain Graphs of Order at Most 10	201
F.2 Graphs from Various Structure Classes	202
F.3 Proper Colouring Benchmark Graphs	204
F.4 Random Graphs	205
G The CD Accompanying this Dissertation	207

List of Figures

1.1	Graph representation of a map	4
1.2	Illustration of different types of vertex colourings	5
1.3	An application of maximum degree colourings	7
2.1	An example of a $(7,8)$ -graph and its complement	12
2.2	The notions of an isomorphism and of equality in graphs	13
2.3	The notion of a subgraph	13
2.4	The notions of internal vertices, leaves, children and the root in a rooted tree	14
2.5	The notions of a bridge and a cut-vertex in a graph	15
2.6	The notions of the union, the join and the cartesian product of two graphs	15
2.7	The notions of a regular graph, vertex-transitivity and a perfect matching	16
2.8	The notions of a complete graph and of complete multipartite graphs	17
2.9	Some special graphs, namely a star, a wheel and a circulant	18
2.10	Proper colourings of the Grötzsch graph	24
2.11	Colouring of the Grötzsch graph via the sequential and largest-first colouring algorithms	27
2.12	Subgraphs during the course of Brelaz's heuristic and the resulting proper colouring	28
2.13	Colouring of the Grötzsch graph via the smallest-last colouring algorithm	29
2.14	A graph used to illustrate the working of Brooks' algorithm	30
2.15	Proper colourings of two subgraphs obtained during the course of Brooks' algorithm	32
2.16	The resulting proper colouring of the graph in Figure 2.14 obtained via Brooks' algorithm	32
2.17	Tree construction during execution of the algorithm by Brown	34
2.18	Tree construction during execution of the algorithm by Brelaz	35
2.19	Proper colourings obtained via Brelaz's heuristic in the Herrmann-Hertz algorithm	42
2.20	Proper colourings obtained in the descending phase of the Herrmann-Hertz algorithm	43
2.21	Proper colourings obtained in the augmenting phase of the Herrmann-Hertz algorithm	44
3.1	An illustration of Lovász's Theorem	57
3.2	A non- $\Delta(1,3)$ -colourable planar graph	58
3.3	A graph constructed by joining isolated vertices to a vertex of maximum degree	60
4.1	A χ_d^Δ -colouring of a tree	64

4.2	χ_d^Δ -colourings of cycles	65
4.3	χ_d^Δ -colourings of wheels	66
4.4	χ_d^Δ -colourings of K_8	67
4.5	χ_d^Δ -colourings of $P_3 \times P_4$ and $P_4 \times C_3$	69
4.6	χ_d^Δ -colourings of the cartesian product of two cycles	70
4.7	A χ_0^Δ -colouring of $K_5 \times K_7$	71
4.8	Number of vertices per colour class in a χ_1^Δ -colouring of $K_4 \times K_{12}$	72
4.9	Number of vertices per colour class in a χ_1^Δ -colouring of $K_{11} \times K_{12}$	74
4.10	Number of vertices per colour class in a χ_1^Δ -colouring of $K_4 \times K_7$ and $K_4 \times K_6$	75
4.11	A χ_1^Δ -colouring of $K_4 \times K_9$	75
4.12	χ_d^Δ -colourings of $C_n\langle 1, 2 \rangle$	78
4.13	χ_2^Δ -colourings of $C_{10}\langle i_1, i_2 \rangle$	79
4.14	χ_d^Δ -colourings of \overline{C}_{10}	81
4.15	Graphical representation of complete balanced multipartite graphs	82
4.16	Two $\Delta(d, x)$ -colourings of $K_{6 \times 15}$	82
4.17	Similarities between D_x^Δ -colourings and a normalized k -partite colouring	83
4.18	The trivial and balancing cases in normalized k -partite colourings	85
4.19	The reduction and composition cases in normalized k -partite colourings	86
4.20	Types of vertices in the progress tree of Algorithm $\overline{\alpha}(x, k)$	86
4.21	The progress tree of Algorithm $\overline{\alpha}(x, k)$ for Example 4.2	87
4.22	The normalized $\Delta(\tilde{d}, 10)$ 11-partite colouring obtained in Example 4.2	87
4.23	Rare cases where the strategy in Algorithm $\overline{\alpha}(x, k)$ may be improved upon	88
4.24	A $\Delta(18, 10)$ -colouring of $K_{11 \times 28}$	89
4.25	An ideal $\Delta(9, 10)$ -colouring of $K_{11 \times 14}$	90
4.26	Discretizing an ideal $\Delta(d, x)$ -colouring of a complete balanced multipartite graph	90
4.27	A $\Delta(10, 10)$ -colouring of $K_{11 \times 14}$ achieved via Algorithm $\overline{D}_x^\Delta(K_{k \times n})$	93
5.1	A graph illustrating the working of the colour degree heuristic	97
5.2	A $\Delta(1, 3)$ -colouring of the graph in Figure 5.1(a) obtained via the colour degree heuristic	97
5.3	Initial random colourings of the graph G_1 in Figure 5.1(a) obtained in Algorithm 9	102
5.4	Colourings of the graph G_1 in Figure 5.1(a) obtained during execution of Algorithm 9	103
5.5	Mean objective function values for the graph G_1 in Figure 5.1(a)	109
5.6	Mean objective function values for the random graph $G_{35,0.5}$ #2	112
5.7	Mean objective function values for the random graph $G_{65,0.5}$ #1	118
5.8	Trendlines for the data sets generated to determine values for the tabu parameters	119
5.9	Renumbering of the vertices of the graph G_1 in Figure 5.1(a)	130
5.10	Colourings of the subgraphs obtained during execution of Algorithm 11	133
5.11	Tree construction during execution of the irredundant χ_d^Δ -colouring algorithm	137

6.1	A $\Delta(2, 3)$ -colouring of K_4^3	157
6.2	A $\Delta(1, 5)$ -colouring of K_6^2	157
6.3	A χ_0^Δ -colouring of $K_{4 \times 3} \times K_4$	158
6.4	A $\Delta(3, 3)$ -colouring of $K_{4 \times 3} \times K_4$	158
6.5	The Filter 1 and Filter 2 strategies in search of a 3-partition of a circulant	160
6.6	The Filter 3 strategy in search of a 2-partition of a circulant	161
A.1	Illustration of Kempe chains	179
A.2	Recolouring part of a map using Kempe chains	180
A.3	Heawood's counter-example to Kempe's attempted proof	181
B.1	First steps towards colouring a minimal non-five-colourable map with 5 colours	183
B.2	Recolouring part of a graph using Kempe chains	184
B.3	Extending a Kempe chain in a graph to a cycle	185
F.1	A Mycielski construction	205

List of Tables

2.1	Truth table for the binary operations or and and	22
2.2	Graph theoretic decision problems in the classes P and NP-complete	22
2.3	Parameter values during execution of Brown’s modified colouring algorithm	39
4.1	Values of and upper bounds on $\alpha(x, k)$ for normalized $\Delta(\tilde{d}, x)$ k -partite colourings	84
5.1	Values of the parameters during execution of Algorithm 8 in Example 5.1	98
5.2	Results obtained by the colour degree heuristic for small graphs and structure classes	104
5.3	Results obtained by the colour degree heuristic for complete balanced multipartite graphs	105
5.4	Results obtained by the colour degree heuristic for proper colouring benchmark graphs	105
5.5	Results obtained by the colour degree heuristic applied to random graphs	105
5.6	Results of a sensitivity analysis performed on the colour degree heuristic	107
5.7	Tabu parameter values for which an objective function value of 16 was obtained	108
5.8	Results obtained by Algorithm 9 applied to small graphs and trees	110
5.9	The average number of iterations for the cases in Table 5.8	110
5.10	Results obtained by Algorithm 9 applied to graphs from various structure classes	114
5.11	The average number of iterations for the cases in Table 5.10	114
5.12	Results obtained by Algorithm 9 applied to complete balanced multipartite graphs	114
5.13	Results obtained by Algorithm 9 applied to some proper colouring benchmark graphs	115
5.14	Times and average number of iterations for the graphs in Table 5.13	115
5.15	Results obtained by Algorithm 9 applied to random graphs of orders 20 and 35	116
5.16	Execution times to determine tabu parameter values for graphs of order at least 50	117
5.17	Values of d when the time out condition was reached during execution of Algorithm 9	120
5.18	Results obtained by Algorithm 9 applied to proper colouring benchmark graphs	121
5.19	Results obtained by Algorithm 9 applied to random graphs of order 50	121
5.20	Results obtained by Algorithm 9 applied to random graphs of orders 65 and 85	122
5.21	Results of improvement on Algorithm 9 applied to graphs from structure classes	123
5.22	Results of improvement on Algorithm 9 applied to the benchmark graphs	124
5.23	Results of improvement on Algorithm 9 applied to the random graphs	125
5.24	Parameter values during the execution of the irredundant χ_d^Δ -colouring algorithm	129

5.25	Parameter values during the execution of the critical χ_d^Δ -colouring algorithm	132
5.26	Results obtained by the irredundant χ_d^Δ -colouring algorithm applied to small graphs . . .	134
5.27	Results obtained by Algorithm 10 applied to graphs from various structure classes	135
5.28	Results obtained by Algorithm 10 applied to complete balanced multipartite graphs . . .	136
5.29	Δ -chromatic sequences of some benchmark instances obtained by Algorithm 10	138
5.30	Results obtained by Algorithm 10 applied to benchmark instances resulting in a time out	138
5.31	Δ -chromatic sequences of certain random graphs obtained by Algorithm 10	139
5.32	Results obtained by Algorithm 10 applied to random graphs resulting in a time out	139
5.33	Results obtained by the critical χ_d^Δ -colouring algorithm applied to small graphs	140
5.34	Rankings of the four $\Delta(d, x)$ -colouring algorithms according to their performances	142
5.35	Results obtained by the critical χ_d^Δ -colouring algorithm applied to random graphs	143
6.1	Basic sequences	147
6.2	Δ -chromatic basic sequences and graphs realizing these sequences	151
6.3	Values of and upper bounds on $\chi_d^\Delta(K_{k \times n})$	152
6.4	Values of n and d for which $K_{k \times n}$ fails to classify basic sequences as Δ -chromatic	152
6.5	Upper bounds on the size of a graph realizing basic sequence 4444222222221	155
6.6	A selection of circulants of order 24 not realizing basic sequence 4444222222221	164
6.7	Circulants disregarded as graphs realizing the Δ -chromatic sequence 4444222222221 .	165
C.1	Sequences satisfying the conditions of Theorem 3.13	187
D.1	Values of and upper bounds on $\alpha(x, k)$	195
D.2	Values of $\bar{\alpha}(x, k)$ for $1 \leq x \leq 17$ and $2 \leq k \leq 18$	197
F.1	Δ -chromatic sequences of connected graphs of order at most 5	201
F.2	Δ -chromatic sequences of connected r -regular graphs of orders 6, 7 and 8	202
F.3	Δ -chromatic sequences of connected vertex-transitive, r -regular graphs of order 10	202
F.4	Δ -chromatic sequences of trees of order 8	202
F.5	Δ -chromatic sequences of cycles, wheels and complete graphs	203
F.6	Δ -chromatic sequences of complete bipartite graphs	203
F.7	A selection of complete balanced multipartite graphs	204
F.8	A selection of the DIMACS benchmark graphs	204
F.9	Random graphs	205

List of Algorithms

1	Maximum degree algorithm	19
2	Sequential colouring algorithm	27
3	Brooks' algorithm	31
4	Brown's modified colouring algorithm	36
5	Herrmann–Hertz algorithm	40
6	Algorithm $\bar{\alpha}(x, k)$ for normalized $\Delta(\tilde{d}, x)$ k -partite colourings	84
7	Algorithm $\bar{D}_x^\Delta(K_{k \times n})$	92
8	Colour degree heuristic	96
9	Tabu search $\Delta(d, x)$ -colouring heuristic	100
10	Irredundant χ_d^Δ -colouring algorithm	126
11	Critical χ_d^Δ -colouring algorithm	131
12	Filter 5 asymmetric 3-partition algorithm	162
13	Filter 4 asymmetric 2-partition algorithm	163
14	Upper bound on the optimal residual weight of a string distribution over a set of hats	196

Reserved Symbols

Non graph specific notation

\subseteq	the relation $\mathcal{A} \subseteq \mathcal{B}$ (defined between sets) states that the set \mathcal{A} is a subset of (or equal to) the set \mathcal{B} .
\mathbb{N}	the set of natural numbers $\{1, 2, 3, \dots\}$
\mathbb{N}_0	the set of counting numbers $\mathbb{N} \cup \{0\}$
$\gcd(a, b)$	the greatest common divisor of two integers a and b
$\binom{m}{n}$	the binomial coefficient $\frac{m!}{(m-n)!n!}$

Basic graph notation

$G(V, E)$	a graph G with vertex set $V(G)$ and edge set $E(G)$
(p, q) -graph	a graph of order p and size q
\mathcal{G}	a family of graphs
$p(G)$	order of a graph G , <i>i.e.</i> $p(G) = V(G) $
$q(G)$	size of a graph G , <i>i.e.</i> $q(G) = E(G) $
$A(G)$	$p \times p$ adjacency matrix of an order p graph G
$N_G(v)$	open neighbourhood set of a vertex v in a graph G
$N_G[v]$	closed neighbourhood set of a vertex v in a graph G
$d_G(u, v)$	distance between two vertices u and v in a graph G
$\deg_G(v)$	degree of a vertex v in a graph G
$\delta(G)$	minimum vertex degree of a graph G
$\Delta(G)$	maximum vertex degree of a graph G
$\rho(G)$	degeneracy of a graph G
$\tau(G)$	path number of a graph G
$g(G)$	girth of a graph G
$k(G)$	number of components of a graph G
$\nu(G)$	matching number of a graph G
$\beta(G)$	independence number of a graph G
$\omega(G)$	clique number of a graph G
$\mathfrak{c}(G)$	clique partition number of a graph G

Graph relations and operations

$\phi : V(G) \rightarrow V(H)$	isomorphism (function) between two graphs G and H
$G \cong H$	graphs G and H are isomorphic
\overline{G}	complement of a graph G
$H \subseteq G$	the graph H is a subgraph of the graph G
$H \subset G$	the graph H is a proper subgraph of the graph G
$\langle S \rangle_G$	subgraph of a graph G induced by a given subset S of $V(G)$
$G - S$ [$G - J$]	resulting subgraph of a graph G after the deletion of a vertex subset S [edge subset J] from the graph G
$G - v$ [$G - e$]	resulting subgraph of a graph G after the deletion of a single vertex v [edge e] from the graph G
$G + J$ [$G + e$]	resulting subgraph of a graph G after the addition of a edge subset J [edge e] of $E(\overline{G})$ to the graph G
$G \cup H$	union of two graphs G and H
nG	union of n isomorphic copies of a graph G
$G + H$	join of two graphs G and H
$G \times H$	cartesian product of two graphs G and H
G^n	cartesian product of n isomorphic copies of a graph G
$G \oplus H$	edge union of two graphs G and H

Specific types of graphs

C_n	a cycle of order n
$C_n \langle i_1, \dots, i_z \rangle$	a circulant of order n with connection set $\{i_1, \dots, i_z\}$
K_0	the empty graph
K_n	a complete graph of order $n \geq 1$
K_{n_1, \dots, n_k}	a complete multipartite graph with partite set cardinalities $n_1 \leq \dots \leq n_k$
$K_{k \times n}$	a complete, balanced multipartite graph with k partite sets of cardinality n
P_n	a path of order n
S_n	a star of order n
T_n	a tree of order n
W_n	a wheel of order n

Graph colourings

\mathcal{R}	colouring rule
$\chi(G)$	chromatic number of a graph G
$\chi_k^\lambda(G)$	$\lambda(k)$ -chromatic number of a graph G
$\chi_k^\tau(G)$	k -th path chromatic number of a graph G
$\chi_k^\omega(G)$	k -th clique chromatic number of a graph G
$\chi_d^\Delta(G)$	$\Delta(d)$ -chromatic number of a graph G
$D_x^\Delta(G)$	smallest value of d for which a $\Delta(d, x)$ -colouring of a graph G exists (for some fixed value of x)

Complexity theory

$\mathcal{D}_1 \rightsquigarrow \mathcal{D}_2$	\mathcal{D}_1 is polynomial time reducible to \mathcal{D}_2
$T(n)$	time complexity of an algorithm for input size n
$S(n)$	space complexity of an algorithm for input size n
$\mathcal{O}(g(n))$	a function $f(n)$ grows no faster than a multiple of $g(n)$ as $n \rightarrow \infty$ (denoted $f(n) = \mathcal{O}(g(n))$), if there are constants $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$
P	the class of polynomial time decision problems
NP	the class of non-deterministic polynomial time decision problems
CN	the chromatic number problem
– FxGenCN	the decision problem asking whether a graph may be coloured with at most x colours, such that each colour class induced subgraph is F -free
– K_{1,d+1}xGenCN	the decision problem asking whether a graph may be coloured with at most x colours, such that the maximum degree of each colour class induced subgraph is at most d

Glossary

Acyclic Said of a *graph* that contains no *cycles*.

Addition The *graph* $G + J$ derived from a *graph* G by adding a nonempty subset J of the edge set of the complement of G to the edge set of G , i.e. $V(G + J) = V(G)$ and $E(G + J) = E(G) \cup J$ where $J \subseteq E(\overline{G})$.

Adjacency Matrix A $p \times p$ symmetric matrix associated with a *graph* G of order p , where the (i, j) -th matrix element takes the value 1 if *vertex* i is *adjacent* to *vertex* j , and the value 0 otherwise.

Adjacent Said of two *vertices* of a *graph* if they are *joined* by an *edge*. Also said of two *edges* that are *incident* with the same *vertex*. Otherwise, the *vertices/edges* are nonadjacent.

Algorithm An ordered sequence of logical instructions for solving a given, well-defined problem within a finite number of steps.

Algorithmic Complexity A measure of the number of *basic operations* performed, or the memory expended by an *algorithm*.

Asymptotically Dominated Said of a function $f(n)$ that is less than a multiple of another function $g(n)$ for all values of n greater than some specified value.

Automorphism An *isomorphism* from a *graph* onto itself.

Basic Operation A single operation performed during the execution of an *algorithm*.

Bipartite Said of a k -*partite graph* if $k = 2$.

Boolean Variable A variable that may take on one of only two values, say **true** or **false**.

Bridge An *edge* e of a *graph* G with the property that the *graph* $G - e$ has more *components* than G .

Cardinality The number of elements in a set.

Cartesian Product The *graph* obtained from two *graphs* G and H with vertex set $V(G) \times V(H)$ and edges $(u_1, u_2)(v_1, v_2)$ where either $u_1 = v_1$ and $u_2 v_2 \in E(H)$ or $u_2 = v_2$ and $u_1 v_1 \in E(G)$.

$\chi(G)$ -**colourable** Said of a *graph* G when $\chi(G)$ colours may be used in a *proper colouring* of G .

$\chi_d^\Delta(G)$ -**colourable** Said of a *graph* G when $\chi_d^\Delta(G)$ colours may be used in a *maximum degree colouring* of G for a given value of d .

$\chi_k^\lambda(G)$ -**colourable** Said of a *graph* G when $\chi_k^\lambda(G)$ colours may be used in a k -*admissible colouring* of G with respect to a graph parameter λ .

$\chi(G)$ -**colouring** A *proper colouring* of a *graph* G using the smallest possible number of colours.

$\chi_d^\Delta(G)$ -**colouring** A *maximum degree colouring* of a *graph* G using the smallest possible number of colours for a given value of d .

$\chi_k^\lambda(G)$ -**colouring** A k -*admissible colouring* with respect to a graph parameter λ of a *graph* G using the smallest possible number of colours for a given value of k .

- Children** All the vertices in the *neighbourhood set* of an *internal vertex* v on a higher level than v in a *rooted tree*.
- Chromatic Number** The smallest possible number of colours with which a *graph* may be *coloured* in such a manner that no two *adjacent vertices* of the *graph* are assigned the same colour.
- Chromatic Number Problem** The *decision problem* “Given a *graph* G and an integer x , with $2 < x < p(G)$, is the *chromatic number* of G at most x ?”
- Circulant** A *graph* $C_n\langle i_1, \dots, i_z \rangle$ of order n with vertex set $V(C_n\langle i_1, \dots, i_z \rangle) = \{v_0, \dots, v_{n-1}\}$ and edge set $E(C_n\langle i_1, \dots, i_z \rangle) = \{v_\alpha, v_{\alpha+\beta(\text{mod } n)} \mid \alpha = 0, \dots, n-1 \text{ and } \beta = i_1, \dots, i_z\}$ where $n, z \in \mathbb{N}$ with $z < n$ and where $1 \leq i_1, \dots, i_z \leq n-1$ are z distinct integers.
- Class NP** Acronym for *Non-deterministic Polynomial*. The set of all *decision problems* that may be answered **true** by a *polynomial time algorithm*, given additional information, called a certificate to the problem at hand.
- Class NP-complete** The set of all **NP-complete** *decision problems*.
- Class P** Acronym for *Polynomial*. The set of all *decision problems* that may be solved by a *polynomial time algorithm*.
- Clique** A *complete subgraph* of a *graph*.
- Clique Colouring** A *colouring* of a *graph* in such a manner that each *colour class induced subgraph* has no *clique* of some specified order.
- Clique Number** The maximum *order* of a *clique* in a *graph*.
- Clique Partition Number** The minimum number of *cliques* into which a *graph* may be partitioned.
- Colour Class** A subset of the *vertex set* of a *graph* coloured with the same colour in a *colouring* of the *graph*.
- Colour Degree** The number of different colours already assigned to *vertices* in the *neighbourhood* of a *vertex* of a *graph* in a *partial colouring* of the *graph*.
- ($i - j$) Colour Interchange** All *vertices* previously coloured with colour i are recoloured with colour j and all *vertices* previously coloured with colour j are recoloured with colour i during the execution of a *colouring algorithm*.
- Colouring** An assignment of colours (elements of some set) to the *vertices* of a *graph*, one colour to each *vertex*.
- Colouring Rule** A rule to be satisfied by each of the *colour class induced subgraphs* in a *colouring* of a *graph*.
- Complement** A *graph* (denoted \overline{G}) associated with a given *graph* G whose *vertex set* satisfies $V(\overline{G}) = V(G)$ and which contains an *edge* if and only if the *edge* is not an *edge* of G .
- Complete, Balanced k -partite Graph** A *complete k -partite graph* in which all the partite sets have the same *cardinality*.
- Complete Graph** A *graph* of order n that is $(n-1)$ -regular.
- Complete k -partite Graph** A *k -partite graph* in which every pair of *vertices* not belonging to the same partite set is *adjacent*.
- Component** A *connected subgraph* of a *graph* G that is not a *subgraph* of any other *connected subgraph* of G .
- Composite** Said of a *circulant* with *connection set* of *cardinality* greater than 1.

- Computational Problem** A problem which has a real number (or collection of real numbers) as solution instead of a mere binary value.
- Connected** Said of a graph G if there exists a u - v path in G for every vertex pair $u, v \in V(G)$. Otherwise, the graph is said to be disconnected.
- Connection Set** The set $\{i_1, \dots, i_z\}$, where $1 \leq i_1, \dots, i_z \leq n-1$ are z distinct integers, that defines the edge set of the circulant $C_n\langle i_1, \dots, i_z \rangle$ of order n .
- Critical $\Delta(d, x)$ -chromatic** Said of a graph G with the property that for every proper subgraph H of G the $\Delta(d)$ -chromatic number of H is strictly less than the $\Delta(d)$ -chromatic number of G (which is x).
- Critical $\lambda(k, x)$ -chromatic** Said of a graph G with the property that for every proper subgraph H of G the $\lambda(k)$ -chromatic number of H is strictly less than the $\lambda(k)$ -chromatic number of G (which is x).
- Critical x -chromatic** Said of a graph G with the property that for every proper subgraph H of G the chromatic number of H is strictly less than the chromatic number of G (which is x).
- Cut-vertex** A vertex v of a graph G with the property that the graph $G - v$ has more components than G .
- Cycle** A walk of length at least 3 with the property that the first and last vertices are the same and no other (internal) vertices are repeated. A graph consisting of a single cycle of length n is so called and denoted C_n .
- Decision Problem** A problem that may be formulated as a binary question, which may be answered either **true** or **false**.
- Degree** The number of vertices adjacent to a vertex in a graph.
- Density** The ratio between the size of a graph and the size of a complete graph of the same order.
- Deletion** The subgraph $G - S$ [spanning subgraph $G - J$] of a graph G with vertex set $V(G) \setminus S$ [edge set $E(G) \setminus J$] for a nonempty vertex [edge] subset $S \subseteq V(G)$ [$J \subseteq E(G)$].
- D_x^Δ -colouring** An optimal maximum degree colouring of a graph G in which the colour class induced maximum degree for a given number of colours x , is minimized.
- Δ -chromatic Sequence** The sequence of the number of colours required in a $\chi_d^\Delta(G)$ -colouring of a graph G as the parameter d increases.
- $\Delta(d)$ -chromatic Number** The smallest possible number of colours used in a colouring of a graph in such a manner that each colour class induced subgraph has a maximum degree of at most d .
- $\Delta(d, x)$ -chromatic** Said of a graph with $\Delta(d)$ -chromatic number equal to x .
- $\Delta(d, x)$ -colourable** Said of a graph G if there exists a $\Delta(d, x)$ -colouring of G .
- $\Delta(d, x)$ -colouring** A maximum degree colouring of a graph in which x colours are used (and where each colour class induced subgraph has a maximum degree of at most d).
- Disjoint** Said of two sets if their intersection is empty.
- Distance** The length of a shortest path between two vertices in a graph if such a path exists, or infinity if such a path does not exist.
- Edge** A 2-element subset of the vertex set of a graph.
- Edge Set** A finite (possibly empty) set, $E(G)$, comprising all the edges of a graph G .
- Edge Union** A graph whose edge set is the union of the disjoint edge sets of two graphs G and H where $V(G) = V(H)$.

- (Time) Efficient** Said of a *polynomial (time) algorithm*.
- Elementary** Said of a *circulant* with *connection set* of *cardinality* 1.
- Embedded** Said of a *graph* that is drawn in the plane in such a way that no two *edges* intersect, except possibly at a *vertex*.
- Empty Graph** A combinatorial object, denoted by K_0 , used to indicate that during *graph* operations one has arrived at the inadmissible situation where the *vertex set* of the resulting *graph* becomes empty.
- Equal** Said of two *graphs* if they have the same *vertex set* and *edge set*.
- End-vertex** A *vertex* of a *graph* that has a *degree* of 1.
- Even [Odd] Cycle** A *cycle* of even [odd] *length*.
- Even [Odd] Vertex** A *vertex* of a *graph* that has an even [odd] *degree*.
- Exact Method** An *algorithm* which gives an optimal solution to a specific optimization problem.
- Exponential Time** Said of an *algorithm* whose *worst-case (time) complexity* is *asymptotically dominated* by an exponential function.
- F-free** Said of a *graph* if it does not contain an *induced subgraph isomorphic* to the graph F .
- Girth** The *length* of a shortest *cycle* in a *graph*. If no *cycles* exist in the *graph*, the *girth* is taken as infinite, by convention.
- (Simple) Graph** A combinatorial object $G = (V, E)$ consisting of a nonempty, finite set $V(G)$ of elements called *vertices*, together with a (possibly empty) set $E(G)$ of 2-element subsets of V called *edges*.
- Greedy Algorithm** An unsophisticated *algorithm* which progressively builds up a feasible (not necessarily optimal) solution to an optimization problem by making the best possible choice at each iteration, regardless of the subsequent effect of that choice.
- Height** The *length* of a longest *path* from the *root* in a *rooted tree*.
- Hereditary** Said of a family of *graphs* if every *induced subgraph* of any *graph* in the family is also a *graph* in the family.
- Heuristic Method** An *algorithm* which provides a feasible solution to a specific optimization problem that is hopefully close to optimal, within a reasonable amount of computational time for all *problem instances*, but which does not guarantee optimality.
- Incident** Said of a *vertex* v and an *edge* e of a *graph* G if e joins v to another *vertex* in G .
- Independence Number** The maximum *cardinality* of an *independent set* of a *graph*.
- (Vertex) Independent Set** A subset of the *vertex set* of a *graph* containing no *adjacent vertex* pairs.
- (Vertex-)Induced Subgraph** A *subgraph* H of a *graph* G with the property that $uv \in E(H)$ if $uv \in E(G)$ for all *vertex* pairs $u, v \in V(H)$.
- (Algorithm) Input** The information required for the implementation of an *algorithm* on a given *problem instance*.
- Internal Vertex** A *vertex* of a *graph*, or in a *walk* within a *graph*, which is not an *end-vertex* of the *graph* or of the *subgraph induced* by the *walk*.
- Isolated Vertex** A *vertex* of a *graph* which has no *adjacent vertices*.
- Isomorphic** Said of two *graphs* between which there exists an *isomorphism*.

- Isomorphism** A one-to-one and onto function between the *vertex sets* of two *graphs* that preserves *adjacency*.
- Iterative Algorithm** An *algorithm* that iterates through a number of steps during execution, without the possibility of calling itself.
- Join** Said of the *edge* between two *adjacent vertices* of a *graph*. Also said of the *graph* obtained from the *union* of two *graphs* G and H together with the edges $\{uv : u \in V(G), v \in V(H)\}$.
- k -admissible Colouring** A *colouring* of a *graph* G with respect to a *graph* parameter λ in such a manner that each *colour class* \mathcal{C} satisfies $\lambda(\langle \mathcal{C} \rangle) \leq k$.
- k -connected** Said of a *graph* from which at least k *vertices* must be removed before the *graph* is *disconnected*.
- k -partite** Said of a *graph* G if the *vertex set* may be partitioned into k subsets, such that no *edge* of G joins *vertices* in the same subset.
- λ -chromatic Sequence** The sequence of the number of colours required in a $\chi_k^\lambda(G)$ -*colouring* of a *graph* G as the parameter k increases.
- $\lambda(k)$ -chromatic Number** The smallest possible number of colours used in a *colouring* of a *graph* G with respect to a *graph* parameter λ in such a manner that each *colour class* \mathcal{C} satisfies $\lambda(\langle \mathcal{C} \rangle) \leq k$.
- $\lambda(k, x)$ -chromatic** Said of a *graph* with $\lambda(k)$ -*chromatic number* equal to x .
- $\lambda(k, x)$ -colourable** Said of a *graph* G if there exists a k -*admissible colouring* of G with respect to a *graph* parameter λ in x colours.
- $\lambda(k, x)$ -colouring** A k -*admissible colouring* of a *graph* with respect to a *graph* parameter λ in which x colours are used.
- Leaf** An *end-vertex* of a *tree*.
- Length** The number of *edges* contained in a *cycle*, *path* or *walk*.
- Level** A subset of the *vertex set* of a *rooted tree* containing all the *vertices* at the same *distance* from the *root*.
- Matching** A 1-*regular subgraph* of a *graph*.
- Matching Number** The *size* of a *maximum matching* of a *graph*.
- Maximum [Minimum] Degree** The maximum [minimum] of all *vertex degrees* of a *graph*.
- Maximum Degree Colouring** A *colouring* of a *graph* in such a manner that each *colour class induced subgraph* has a *maximum degree* of at most some specified value.
- Maximum Matching** A *matching* for which the *edge set* has maximum *cardinality*.
- Multipartite** Said of a k -*partite graph* if $k > 2$.
- (Open) [(Closed)] Neighbourhood Set** The set of all *vertices adjacent* to a given *vertex* v in a *graph* [including v itself].
- Non-decreasing** Said of a function $f(x)$ if $f(x) \geq f(y)$ whenever $x \geq y$.
- NP-complete** Said of a *decision problem* \mathcal{D} in the class **NP** if all *decision problems* in **NP** are *polynomial time reducible* to \mathcal{D} .
- Of the Order** Said of a function $f(n)$ if there exists a constant $c \in \mathbb{R}^+$ such that $c \cdot g(n)$ is larger than $f(n)$ as $n \rightarrow \infty$, where the function $g(n)$ is this order of $f(n)$.
- $\omega(k)$ -chromatic Number** The smallest possible number of colours used in a *colouring* of a *graph* in such a manner that each *colour class induced subgraph* has no *clique* of order $k + 1$.

- $\omega(k, x)$ -colouring** A *clique colouring* of a *graph* in which x colours are used (and where each *colour class induced subgraph* has no *clique* of order $k + 1$).
- Order** The *cardinality* of the *vertex set* of a *graph*.
- Path** A *walk* in a *graph* with the property that no *vertex* is repeated. A *graph* of order n consisting of only a *path* is so called and is denoted by P_n .
- Path Colouring** A *colouring* of a *graph* in such a manner that the *order* of a longest *path* in each *colour class induced subgraph* is at most some specified value.
- Path Number** *Order* of a longest *path* in a *graph*.
- Perfect Matching** A *matching* of a *graph* G , if it exists, containing all the *vertices* of G .
- Planar Graph** A *graph* that may be *embedded* in the plane. Otherwise, the *graph* is said to be nonplanar.
- Plane Graph** A *planar graph* already *embedded* in the plane.
- Polynomial Time** Said of an *algorithm* whose *worst-case (time) complexity* is *asymptotically dominated* by a polynomial function.
- Polynomial Time Reducible** Said of a *decision problem* \mathcal{D}_1 with respect to *decision problem* \mathcal{D}_2 if (1) there exists a function f transforming any instance I_1 of \mathcal{D}_1 to an instance $f(I_1)$ of \mathcal{D}_2 such that the answer to I_1 with respect to \mathcal{D}_1 is **true** if and only if the answer to $f(I_1)$ with respect to \mathcal{D}_2 is **true**, and (2) if there exists an *efficient algorithm* to implement the function f .
- Problem Instance** A particular set of values or quantities that completely describes the problem parameters for a *decision problem* or a *computational problem*.
- Proper Colouring** A *colouring* of a *graph* in such a manner that no two *adjacent vertices* of the *graph* are assigned the same colour.
- Proper Subgraph** A *subgraph* H of a *graph* G with the property that at least one of the *vertex set* or the *edge set* of H is a proper subset of the corresponding set of G .
- Pseudo-code** A method of expressing an *algorithm* using a natural language rather than a specific computer programming language.
- Recursive Algorithm** An *algorithm* that may perform calls to itself.
- Recursive Calls** Calls made during execution of a *recursive algorithm* to itself.
- Regular** Said of a *graph* in which each *vertex* has the same *degree*, say r for some $r \in \mathbb{N}_0$, in which case the *graph* is said to be r -regular.
- Root** Any distinguished *internal vertex* of a *tree*.
- Rooted Tree** A *tree* that contains a *root*.
- Singular** Said of a *circulant* of even *order* n for which one of the elements in the *connection set* is $n/2$. Otherwise, the *circulant* is said to be non-singular.
- Size** The *cardinality* of the *edge set* of a *graph*.
- Space Complexity** The amount of (computer) memory expended by an *algorithm* as a function of its input size.
- Spanning Subgraph** A *subgraph* H of a *graph* G with the property that $V(H) = V(G)$.
- Spanning Tree** A *spanning subgraph* of a given *graph* that is also a *tree*.
- Star** The *complete bipartite graph* $K_{1,n-1}$ of order n .

Subgraph A graph H associated with a graph G with the properties that $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$.

$\tau(k)$ -**chromatic Number** The smallest possible number of colours used in a *colouring* of a graph in such a manner that the *order* of a longest *path* in each *colour class induced subgraph* is at most k .

$\tau(k, x)$ -**colouring** A *path colouring* of a graph in which x colours are used (and where the *order* of a longest *path* in each *colour class induced subgraph* is at most k).

Time Complexity The number of *basic operations* performed by an *algorithm* as a function of its input size.

Tractable Said of a *decision problem* that may be solved by a *polynomial time algorithm*. If no such *polynomial time algorithm* is known, the *decision problem* is said to be intractable.

Tree A *connected, acyclic graph*.

Trivial Said of the *graph* of *order* 1 (there is only one up to isomorphism). Otherwise, a graph is said to be nontrivial.

Union A *graph* whose *vertex set* is the union of the *disjoint vertex sets* of two *graphs* G and H and whose *edge set* is the union of the *edge sets* of G and H .

Vertex A combinatorial object in terms of which the *vertex set* and *edge set* of a *graph* is defined.

Vertex set A nonempty, finite set, $V(G)$, of all the *vertices* of a *graph* G .

Vertex-transitive Said of a *graph* G if, for all *vertex pairs* $(u, v) \in V(G)$, there exists an *automorphism* that maps u to v .

Walk A finite alternating sequence of *incident vertices* and *edges* in a *graph*, both beginning and ending in a *vertex*.

Wheel A *graph* of *order* n obtained by the *join* of a *cycle* of *order* $n - 1$ and one additional *vertex*.

Worst-case Complexity The largest possible number of *basic operations* performed, or memory expended by an *algorithm* for a specific input size.

x -**chromatic** Said of a *graph* with *chromatic number* equal to x .

x -**colourable** Said of a *graph* G if there exists a *proper colouring* of G in x colours.

x -**colouring** A *colouring* of a *graph* in which x colours are used.

Chapter 1

Introduction

“The purest and most thoughtful minds are those which love colour the most.”

The Stones of Venice (1853), by John Ruskin (1819–1900)

1.1 Map Colouring: The Origin of Graph Colouring

When maps of countries are drawn for an atlas, one attempts to colour countries that share a common boundary with different colours. This principle has led to a mathematical statement, known for a long time as the four-colour conjecture:

Any map on a plane surface (or a sphere) may be coloured with at most four colours so that no two adjacent regions have the same colour [42].

Until 1976, proving the four-colour conjecture was one of the most famous unsolved problems in mathematics, ranking in stature with tantalising unresolved problems such as Fermat’s last theorem¹ [2, 76], the Riemann–hypothesis² [35, 76] and the Goldbach conjecture³ [35].

The four-colour conjecture became known as the four-colour disease, since many famous mathematicians spent a great deal of time working on this problem, without complete success. This problem has generated a strange history filled with attempts at proofs, publication of incorrect proofs and, in general, a significant number of unrewarded efforts [52].

The four-colour problem seems to have been formulated for the first time by Francis Guthrie while he was a student at University College, London. He attempted to prove his conjecture, but was not satisfied with his proof [91], so he mentioned the problem to his brother Frederick, also a student at University College, London [42]. Frederick Guthrie in turn asked his mathematics professor, Augustus De Morgan (for whom De Morgan’s Laws of set theory are named, [29]), to verify the “fact” that any map drawn in the plane could be coloured with at most four colours, so that adjacent countries received different colours. De Morgan responded by saying he did not know that this was indeed a “fact,” and in a letter dated October 23, 1852, [91] De Morgan mentioned the problem to Sir William Rowan Hamilton (for

¹Fermat’s last theorem, which states that $x^n + y^n = z^n$ has no non-zero integer solutions for x, y and z when $n > 2$, was finally proved by the British mathematician Andrew Wiles in his paper *Modular elliptic curves and Fermat’s Last Theorem* which appeared in the May 1995 issue of the *Annals of Mathematics*.

²The Riemann zeta-function is given by $\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s}$, where s is a complex number. The function $\zeta(s)$ has zeros at the negative even numbers and one refers to these as the trivial zeros. The Riemann–hypothesis, which states that all nontrivial zeros of $\zeta(s)$ have real part equal to $\frac{1}{2}$, remains open to this day. However, it is known that the first 1.5 billion nontrivial zeros do indeed have real part equal to $\frac{1}{2}$. This hypothesis is also known as Hilbert’s Eighth Problem [66].

³Although the Goldbach conjecture, namely that every even integer may be expressed as the sum of two prime numbers, holds for every even number from 4 to 4×10^{14} , the conjecture in general still remains unproven to this day.

whom hamiltonian graphs are named [29]) [52, 69]. Hamilton replied on 26 October 1852: “*I am not likely to attempt your quaternion of colour very soon*” [90].

Although Hamilton took no interest in the conjecture, De Morgan spoke often of this problem with other mathematicians. De Morgan is credited with writing an anonymous article in the April 14, 1860, issue of the journal *Athenaeum* in which he discusses the four-colour problem. This is the first known published reference to the problem [29, 91]. De Morgan probably also communicated the problem to the English mathematician Arthur Cayley and to a London lawyer Alfred Bray Kempe [42] who had studied mathematics under Cayley at Cambridge and devoted some of his time to mathematics throughout his life [90]. Almost twenty years after it first appeared in print, the four-colour problem was raised as an open problem by Cayley at a meeting of the London Mathematical Society on June 13, 1878 [26, 29, 91] as well as in a paper by Cayley in 1879 in which he postulated why this problem appeared to be so difficult [25, 29, 69]. According to Cayley, a general method for extending a given colouring to include an extra country, will be hard to find, and therein lies the immense difficulty of the four-colour problem [113]. During that same year, Kempe published what seems to have been the first attempted proof of the four-colour conjecture [29, 48, 75]. Kempe made use of so-called Kempe chains to recolour parts of a map so as to be able to colour some uncoloured country. Kempe’s proof of the four-colour theorem may be found in Appendix A of this dissertation.

Kempe received great acclaim for his proof. Based on this proof as well as his work on linkages [113], he was elected a Fellow of the Royal Society and served as its treasurer for many years. He was knighted in 1912 [90]. However, the four-colour problem continued to capture the imagination of many professional⁴ and amateur⁵ mathematicians.

Unfortunately, more than a decade after Kempe’s original proof was published, the four-colour theorem returned to being the four-colour conjecture, when Kempe’s proof was refuted in 1890 by Percy John Heawood in his first paper [60, 69]. Heawood, a lecturer at Durham in England [90], stated almost apologetically that he had discovered an error in Kempe’s proof that is so serious that he was unable to repair it [29]. Kempe reported the error to the London Mathematical Society himself and said he could not correct the mistake in his proof [90]. Heawood’s demolition of Kempe’s proof centres on the generalization of Kempe chains to an uncoloured country X surrounded by five regions coloured in all four colours. Kempe used two simultaneous interchanges of colour to recolour the countries on either side of X so that X itself could then be coloured. Either interchange of colour is perfectly valid, but to perform both at once is not permissible [113]. In his paper, Heawood gave an example of a map which, although it could easily be four-coloured, showed that Kempe’s proof technique did not work in general [29, 113]. The reader is referred to Appendix A of this dissertation for a more in-depth discussion of Heawood’s refutation of Kempe’s proof attempt.

Although Kempe’s work contained a flaw, it also contained a valuable contribution, which formed the basis of many later attempts to solve the four-colour problem, including the successful attempt of Appel and Haken in 1976 [48]. Furthermore, Heawood was able to use Kempe’s technique to prove the five-colour theorem, *i.e.* that every map may be five-coloured [29, 91]. This proof of the five-colour theorem also embodies a polynomial time algorithm to five-colour the vertices of a planar graph⁶ [48]. Heawood was to work throughout his life on map colouring, work which spanned nearly 60 years. He successfully investigated the number of colours needed for maps on surfaces other than the plane and in 1898 he proved that if the number of edges around each region of a map on the plane is divisible by 3 then the regions are four-colourable. He also produced many papers generalising the latter result [90]. Heawood’s proof of the five-colour theorem may be found in Appendix B of this dissertation.

⁴For example, Peter Guthrie Tait, Professor of Natural Philosophy of the University of Edinburgh described yet another “proof” [29, 90, 91]. It contained some clever ideas, but unfortunately also a number of basic errors [90]. Lewis Carroll, author of the famous children’s story “Alice in Wonderland,” created a game for two players in which one player designed a map for his or her opponent to four-colour.

⁵In 1889, the Bishop of London (Frederick Temple), later Archbishop of Canterbury, published his own solution of the four-colour problem in the *Journal of Education* [29, 91]. Temple, like Richard Baltzer among others, had considered it sufficient to prove the four-colour theorem by proving that it is impossible to draw five mutually neighbouring regions in the plane, *i.e.* each region is bordering the other four. If there is a map with five neighbouring regions, then the four-colour theorem is false. From this last fact, Temple, thus made the incorrect logical deduction that if there is no a map with five neighbouring regions, then the four-colour theorem is true [113].

⁶A **planar graph** is a graph that may be drawn in the plane without any of its edges intersecting.

Despite the fact that it is not particularly difficult to prove the five-colour theorem, it is extraordinarily difficult to prove the four-colour theorem. It remained a conjecture for several decades until 1976, when the four-colour conjecture became the four-colour theorem for the second, and last, time. On June 21, 1976 Kenneth Appel and Wolfgang Haken of the University of Illinois, announced that, with the computer aid of John Koch, they had found a complete computer proof of the four-colour conjecture [5, 29, 52, 69]. Their proof made use of two fundamental ideas developed during the first half of the twentieth century, namely so-called **unavoidable sets**⁷ and **reducible configurations**⁸. The approach in Appel and Haken's proof of the four-colour theorem was to find an unavoidable set of reducible configurations [113]. To determine unavoidability they made use of a refinement of Heinrich Heesch's method of **discharging**⁹ [9, 61]. Since the set is unavoidable, every map must contain at least one of the configurations, but each configuration is reducible and thus cannot be contained in a minimal non-four-colourable map. Thus, no minimal non-four-colourable map can exist [113]. The unavoidable set in the Appel and Haken proof has a cardinality of 1 476 [104].

The four-colour theorem was the first major theorem to be proved using a computer, having a proof that could not be verified directly by other mathematicians [90]. For this reason, as well as the fact that their proof is rather lengthy, the proof of Appel and Haken was met with skepticism, especially since the proposed solution had required hundreds of hours of computer calculations to test all 1 476 configurations for reducibility. However, their proof withstood scrutiny and the test of time [29, 91].

Although the four-colour theorem is now known to be true, it still captures the interest of various mathematicians. In 1996 Neil Robertson, Daniel Sanders, Paul Seymour and Robin Thomas published a new proof of the four-colour theorem in the *Electronic Research Announcements of the American Mathematical Society* [97], as well as a hard copy article in 1997 in the *Journal of Combinatorial Theory* [96]. The basic idea of the proof is the same as Appel and Haken's, but among a few other differences, they considered only 633 basic structures as opposed to the 1 476 of Appel and Haken in their unavoidable set, and their discharging procedure for proving unavoidability required only 32 discharging rules, in contrast to the 487 rules used by Appel and Haken. They also obtained a quadratic algorithm to four-colour planar graphs, an improvement over the quartic algorithm of Appel and Haken [104, 113]. In 2000 yet another proof of the four-colour theorem was produced by Ashay Dharwadkar [36]. In his proof he made use of **Steiner systems** (design theory)¹⁰.

From a mathematical point of view, graph colouring theory continually surprises by producing unexpected new results. For example, the century old five-colour theorem for planar graphs (due to Heawood) has this past decade been furnished with a new proof by Carsten Thomassen [105], avoiding the recolouring technique invented by Kempe [69]. Furthermore, even if many deep and interesting results have been obtained during the hundred and fifty years of graph colouring since its inception in 1852, there are many easily formulated, interesting open problems, as stated in the words of William Tutte:

“The four-colour theorem is the tip of the iceberg, the thin end of the wedge and the first cuckoo of spring” [69, 107].

It is in this context that many mathematicians study graph colouring — purely for the (often unexpected) beauty of the mathematics underlying the subject. Although the possibility of practical application is not excluded, this dissertation on maximum degree colouring falls within the latter category.

⁷It may be shown that every map has at least one country with five or fewer neighbours. This means that in every map there must be either a country with one, a country with two, a country with three, a country with four or a country with five neighbours. So in every map at least one country from this collection cannot be avoided and such a collection is called an **unavoidable set** [9, 112].

⁸A **reducible configuration** is any arrangement of countries that cannot occur in a minimal non-four-colourable map, *i.e.* a map that cannot be coloured with four colours and it has as few countries as possible — any map with fewer countries can be coloured with four colours. If a map contains a reducible configuration, then any colouring of the remainder of the map with four colours may be extended, perhaps after necessary local recolouring, to a colouring of the entire map [9, 112].

⁹Heesch formulated a **discharging** rule that assigns a charge of $6 - i$ to each vertex, where i is the degree of the vertex. One can then prove that a set of configurations is unavoidable if one can distribute the charges so that the vertices of a **triangulation** (a planar graph in which every face is a triangle) all have negative charges [12, 112].

¹⁰A **Steiner system** $S(t, k, v)$ is a set X of v points, and a collection of subsets of X of size k (called blocks), such that any t points of X are in exactly one of the blocks. For more information on Steiner systems in particular and design theory in general, see [99].

1.2 From Maps to Graphs: Other Types of Graph Colouring

Hitherto the four-colour theorem and related colouring problems were described in cartographic terms for historical reasons. However, these problems may be formulated elegantly in graph theoretic terms, as will be done in the remainder of this dissertation. Informally, the geographic regions of a map may be represented by points (which are called **vertices**), as in Figure 1.1, in which pairs of vertices representing regions containing common boundaries of positive length are joined by means of lines (called **edges**) to form a graph representing the map topology. Then the problem is to colour the vertices of the resulting graph so that no two adjacent vertices (*i.e.* vertices that are joined by an edge), have the same colour [42]. The vertices in the graph G_1 in Figure 1.1(c) have the same colours as the counties of the original map of Australia in Figure 1.1(a). Since there are four vertices present in the graph G_1 that are all pairwise joined by means of edges, these four vertices have to be coloured with four different colours. However, these four colours are sufficient to colour the graph G_1 completely, as indicated by the colouring of G_1 in Figure 1.1(c). The minimum number of colours with which a graph G may be coloured in this way (*i.e.* such that no two adjacent vertices have the same colour) is usually denoted by $\chi(G)$, and called the **chromatic number** of G . The four-colour theorem states that, for all graphs, G , which represent map topologies on the plane (or sphere)¹¹, $\chi(G) \leq 4$. This method of graph colouring is often referred to as colouring the vertices of a graph *in the classical sense*; hence $\chi(G)$ will be referred to as the *classical chromatic number* of G .

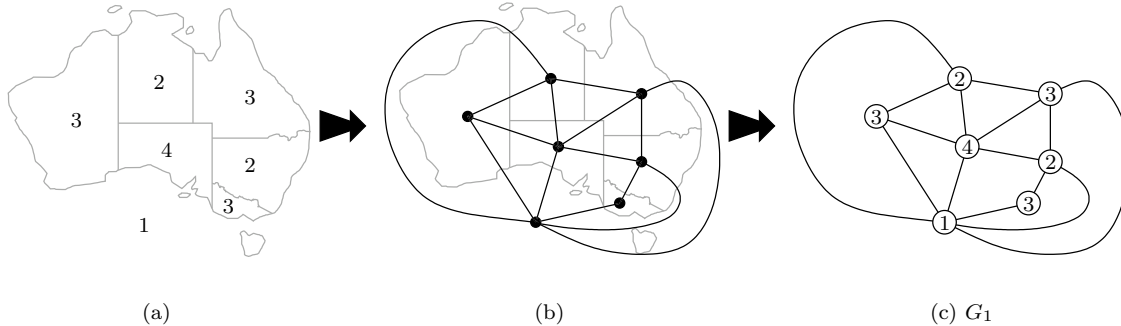


FIGURE 1.1: The counties of Australia are represented in the graph G_1 in (c) by vertices and these vertices have the same colours as the counties in the map in (a). Two vertices in G_1 are connected by means of an edge if the two counties they represent have a common boundary in the original map as indicated in (b).

There are several ways in which the above (classical) notion of graph vertex colouring may be generalized. One way is to colour the vertices of a graph in such a way that none of the colour induced subgraphs have $k + 1$ (or more) vertices for which there exists edges between all $\binom{k+1}{2}$ pairs of these $k + 1$ vertices, for some given integer k . The minimum number of colours with which a graph G may be coloured in this way is usually denoted by $\chi_k^\omega(G)$, and called the **k -th clique chromatic number** of G . Note that the classical chromatic number is therefore a special case of the clique chromatic number, in the sense that $\chi_1^\omega(G) = \chi(G)$, for any graph G . In Figure 1.2(a) the graph G_1 of Figure 1.1 is coloured in this way for $k = 3$. Since the graph contains four vertices that are all pairwise joined by means of edges, $\chi_3^\omega(G_1) > 1$. However, it is possible to colour the graph completely with only two colours, as indicated by the colouring of G_1 in Figure 1.2(a), and it is concluded that $\chi_3^\omega(G_1) = 2$.

A longest path in a graph is the longest traversable sequence of alternating non-repeating vertices and edges, and the length of the path is the number of edges in the path. For any given integer k , another way of colouring the vertices of a graph is to ensure that none of the colour induced subgraphs have a longest path whose length exceeds $k - 1$. The minimum number of colours with which a graph G may be coloured in this way is usually denoted by $\chi_k^\tau(G)$, and called the **k -th path chromatic number** of G . Note that the classical chromatic number is therefore also a special case of the path chromatic number, in

¹¹Such graphs are planar graphs.

the sense that $\chi_1^{\overline{}}(G) = \chi(G)$, for any graph G . In Figure 1.2(b) the path chromatic number is illustrated for $k = 3$. Since the graph contains several triangles, $\chi_3^{\overline{}}(G_1) > 1$. By trial and error it becomes clear that it is not possible to colour the graph completely with only two colours, but three colours are sufficient, as indicated by the colouring of G_1 in Figure 1.2(b), and it is concluded that $\chi_3^{\overline{}}(G_1) = 3$.

Yet another method of graph vertex colouring is so-called **multicolouring**. In multicolouring the objective is to assign sets of colours to the vertices of a graph, such that adjacent vertices receive disjoint sets of colours. Weights on the vertices prescribe upper bounds on the cardinality of the colour sets [68]. The minimum number of colours with which a graph G may be coloured in this way is usually denoted by $\chi(G, w)$, and called the **weighted chromatic number** of G [89]. Note that the classical chromatic number is therefore also a special case of the weighted chromatic number, in the sense that if all the weights are equal to one, $\chi(G, w) = \chi(G)$, for any graph G . The weighted chromatic number is illustrated in Figure 1.2(c) for the weights indicated on the graph.

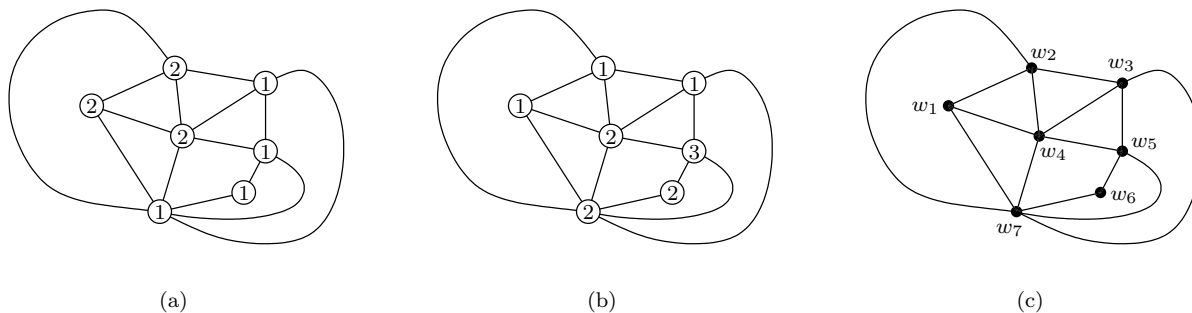


FIGURE 1.2: Let $k = 3$. Then the clique chromatic number for the graph G_1 in Figure 1.1 is $\chi_3^{\omega}(G_1) = 2$ and a colouring for these parameters is illustrated in (a). The path chromatic number for this graph is $\chi_3^{\overline{}}(G_1) = 3$ and a colouring is shown in (b). The values of the weights $w = \{w_1, w_2, w_3, w_4, w_5, w_6, w_7\}$ as indicated in (c), are $w = \{2, 3, 2, 2, 2, 3, 2\}$. A multicolouring of G_1 for these weights may be given by $\{\text{colour 1, colour 2}\}$, $\{\text{colour 3, colour 4, colour 5}\}$, $\{\text{colour 1, colour 2}\}$, $\{\text{colour 6, colour 7}\}$, $\{\text{colour 3, colour 4}\}$, $\{\text{colour 1, colour 2, colour 5}\}$ and $\{\text{colour 8, colour 9}\}$ for the vertices with weights $w_1, w_2, w_3, w_4, w_5, w_6$ and w_7 respectively. The corresponding weighted chromatic number for these indicated weights is $\chi(G_1, w) = 9$.

Before continuing with possible applications of graph vertex colouring and the specific colouring problem to be considered in this dissertation, a totally different way of graph colouring, namely to colour the edges in stead of the vertices of a graph, should be mentioned. This idea was presented for the first time by Peter Guthrie Tait in 1880 [113] and led to an interesting area of graph colouring known as **edge colouring** that is still active today and with its own set of results and applications. However, only vertex colouring problems will be considered in this dissertation. Therefore the prefix “vertex” will be omitted throughout this dissertation; hence only referring to chromatic numbers instead of vertex chromatic numbers or to graph colouring problems instead of graph vertex colouring problems.

1.3 Graph Colouring Applications

Although the four-colour problem became famous more as a mathematical challenge, rather than because of its application to map colouring, graph colouring theory is, in fact, also of interest due to many applications. Graph colouring deals with the fundamental problem of partitioning the vertices of a graph into equivalence classes, according to certain rules [69] as was demonstrated in §1.2. For some of the colouring rules in §1.2 there are interesting application possibilities.

The graph being coloured may, for example, have applications in time tabling and scheduling problems — graph colouring is therefore of practical importance in operations research [42]. Many scheduling problems involve allowing for a number of pairwise restrictions under which jobs may be performed

simultaneously. For instance, in attempting to schedule examinations at a university, two courses are in conflict if these two courses may be taken by the same group of students, and should therefore not be scheduled in the same examination time slot. The problem of determining the minimum number of time slots needed to accommodate all examinations, without scheduling conflicting courses into the same time slot, may be modelled as a graph colouring problem. The courses are represented by the vertices of a graph, with edges between two vertices if students may take the corresponding two courses simultaneously. After colouring the graph in the classical sense, the colours are associated with examination time slots [27, 31, 42, 53, 106].

In another scheduling application flights have to be assigned to a given number of airplanes according to a given flight timetable, under the additional condition that a fixed schedule of maintenances (prescribed for each airplane) may not be changed. In this model the colouring procedure starts from a feasible colouring of a part of the graph in question, and the minimum total number of colours admitting a colouring in the classical sense of the entire graph is sought, where the vertices represent time intervals (maintenances and flights), the colours correspond to the airplanes, the precoloured part is the set of maintenances, and where adjacency means that two represented time intervals overlap [108].

Consider a factory manufacturing a number of chemicals which it then stores in a warehouse. Certain pairs of chemicals may typically not be stored in the same warehouse compartment, because they react with each other when brought into close proximity. The question, “What is the smallest number of compartments into which the warehouse may be partitioned for safe storage?” may be answered using classical graph colouring. Let the vertices and edges of a graph represent chemicals and incompatibilities due to chemical reactions respectively. The number of storage compartments is then equal to the number of colours with which the vertices are coloured. This model also has applications in bin packing and resource allocation problems [29, 31, 42, 53].

One very active resource allocation application of graph colouring, is register allocation. In the register allocation problem the objective is to assign variables to a limited number of computer hardware registers during program execution. Variables in registers may be accessed much quicker than those not in registers. Typically, however, there are far more variables than registers — it is therefore necessary to assign multiple variables to registers. Variables conflict with each other if one is used both before and after the other within a short period of time during program execution (for instance, within a subroutine). The goal is to assign variables that do not conflict so as to minimize the use of non-register memory. A simple approach to achieve this objective is to use a graph model, where the vertices represent variables, and variables that are in conflict (*i.e.* that cannot be stored in the same register) are joined by means of an edge and the vertices representing these variables may not be coloured using the same colour. A classical colouring is then a conflict-free assignment. If the number of colours used is smaller than the number of registers, a conflict-free register assignment is possible [53, 67, 106, 112].

Another application of graph colouring within the computer industry is the problem of testing printed electronic circuit boards for unintended short circuits (caused by stray lines of solder). This gives rise to a graph colouring problem in which the vertices correspond to the nets on the board and there is an edge between two vertices if there is a potential for a short circuit between the corresponding nets. Colouring the graph corresponds to partitioning the nets into “supernets” such that the nets in each supernet may be tested simultaneously for short circuits against all other nets, thereby speeding up the testing process [106].

Practical problems in communication networks also ultimately derive from graph colouring problems [113]. An example is the problem of assigning frequencies to mobile radios and other users of the electromagnetic spectrum. In the simplest case, two customers who are sufficiently close to each other should be assigned different frequencies (so as to minimize disturbances), while those that are far apart may share the same frequencies. This may be modelled as a multicolouring graph problem in which one assigns sets of colours, representing radio frequency channels to the vertices which, in turn, represent the transmitters, so as to avoid excessive interference [53, 68, 106]. Significant cost savings may be affected by minimizing the number of different frequencies utilised.

Graph colouring theory also has a central position in discrete mathematics. It appears in many places with seemingly no or little connection to colouring. A good example is the Erdős–Stone–Simonovits theorem in extremal graph theory [69], showing that, for a family of graphs L , the behaviour of the

maximum number, $f(n, L)$, of edges in a graph G on n vertices not containing any subgraph $G_i \in L$, depends on the minimum number of colours used in a classical colouring of the graph, $\chi(G_k)$, where G_k is the graph in L with the smallest chromatic number of all the graphs in L , in which case [39]

$$\lim_{n \rightarrow \infty} \frac{f(n, L)}{n^2} = \frac{\chi(G_k) - 2}{2\chi(G_k) - 2}.$$

1.4 The Problem to be Considered in this Dissertation

In this dissertation yet another generalization of the classical graph colouring problem will be considered. The degree of a vertex is the number of adjacent vertices to it and the maximum degree of the graph is the maximum of the degrees of all the vertices in the graph. The vertices of a graph will be coloured in such a way that in no colour-induced subgraph, the maximum degree exceeds some specified number $d \in \mathbb{N}_0$. The minimum number of colours with which a graph G may be coloured in this way, shall be denoted by $\chi_d^\Delta(G)$, and called the **d-th maximum degree chromatic number** of G . Note that the classical chromatic number, $\chi(G)$, is therefore also a special case of the maximum degree chromatic number, in the sense that $\chi_0^\Delta(G) = \chi(G)$, for any graph G .

This way of colouring may also be applied within the computer industry. In a scheduling problem as demonstrated in Figure 1.3(a), where users of a computer system are in conflict if they require access to one or more of the same files simultaneously, some threshold of conflict may be tolerated. For example,

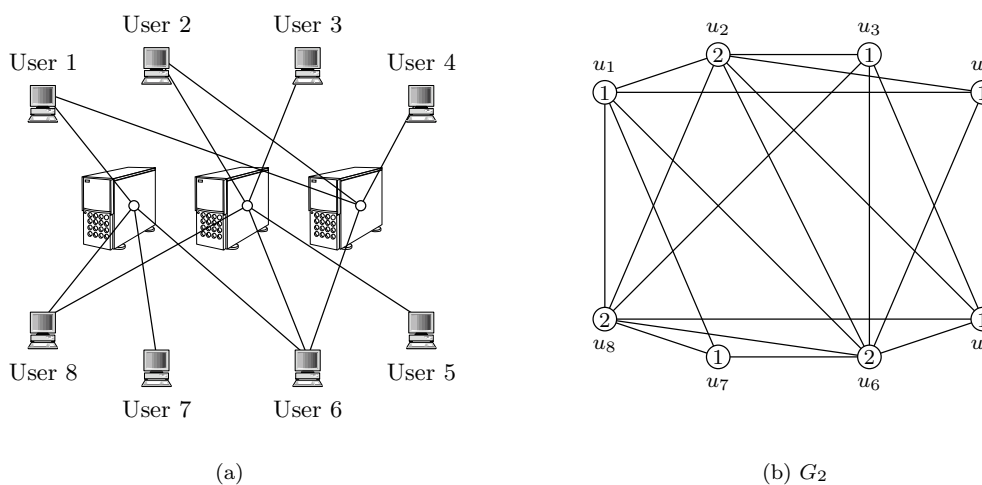


FIGURE 1.3: A schematic representation of users on a computer system is shown in (a) where a line between a user and a file storage computer indicates that the particular user requires access to that specific group of computer files. The file access graph for this computer system is given in (b), where user 1 is represented by vertex u_1 , user 2 is represented by vertex u_2 , and so forth. If two conflicting other users on the system may be tolerated, then the maximum degree chromatic number of G_2 is $\chi_2^\Delta(G_2) = 2$ and vertices u_1, u_3, u_4, u_5 and u_7 are all coloured with colour 1, while vertices u_2, u_6 and u_8 are coloured with colour 2.

each user may find the slowdown incurred for retrieval of data with, say two, conflicting other users on the system acceptable, but with more than two other conflicting users such slowdown becomes unacceptable. A file access graph, G , may be constructed by representing the users on a computer system by vertices, and conflicts between users by edges. A file access graph is illustrated in Figure 1.3(b). An efficient access schedule is the assignment of users to a minimum number of colour classes. If different colours are associated with different time periods in the schedule, an optimal file access schedule for the above set of hypothetical users may be found by means of a colouring of the file access graph with $\chi_2^\Delta(G)$ colours [34]. A colouring satisfying the restriction of no more than two other conflicting users for the file access graph G_2 in Figure 1.3(b) is shown in Figure 1.3(b), indicating that $\chi_2^\Delta(G_2) \leq 2$ — hence no more than

two time periods are required for the file access schedule. It is clear that $\chi_2^\Delta(G_2) > 1$, since the graph G_2 contains four vertices that are all pairwise joined by means of edges. Hence, $\chi_2^\Delta(G_2) = 2$. In the file access schedule of the users in Figure 1.3 users 1, 3, 4, 5 and 7 will have access to the files during one time period and users 2, 6 and 8 will have access during the other time period.

1.5 Dissertation Objectives

In this dissertation the objectives are:

- I To document the relevant results that have appeared in the literature on
 - (a) unifying theories of generalizations of the classical graph colouring problem, and
 - (b) the topic of this dissertation, namely the parameter χ_d^Δ for general graphs.
- II To study properties of the parameter χ_d^Δ for general graphs.
- III To adopt analytic approaches towards determining the value of the parameter χ_d^Δ for graphs from various structure classes, including bipartite graphs, cycles, wheels, products of paths and cycles, complete graphs and certain classes of circulants.
- IV To establish as tight as possible upper bounds for the parameter χ_d^Δ for graph classes for which the parameter cannot be obtained exactly, including certain classes of circulants, products of complete graphs and complete balanced multipartite graphs.
- V To develop algorithms that either
 - (a) determine the parameter χ_d^Δ exactly for a relatively small general graph, or
 - (b) determine an upper bound for the parameter χ_d^Δ for a larger general graph.
- VI To investigate whether known necessary conditions from the literature for the sequence $\{\chi_d^\Delta(G) : d \in \mathbb{N}_0\}$ for a general graph G , are, in fact, also sufficient conditions. This will be done in order to set a platform for the characterization of the sequence $\{\chi_d^\Delta(G) : d \in \mathbb{N}_0\}$ for a general graph G , *i.e.* to determine which integral sequences are, in fact, maximum degree chromatic sequences for graphs — this seems to be a very hard problem.

1.6 Dissertation Layout

Chapter 2 opens, in §2.1 – 2.3 with the basic graph theoretic, complexity theoretic and graph colouring terminologies respectively that are required and used in the remainder of this dissertation. This is followed, in §2.4, by a review of the properties of generalizations of the classical process of graph colouring, as well as two specific generalizations of the notion of graph colouring, namely path and clique colourings mentioned in §1.2. Hence, objective I(a) in §1.5 is addressed in §2.4.

The notion of maximum degree colouring, the generalized graph colouring process that is considered in this dissertation, is introduced formally in §3.1. A literature survey on maximum degree colouring is performed in §3.2 and thus addresses objective I(b) in §1.5. An inverted strategy, employed throughout this dissertation, to determine $\chi_d^\Delta(G)$ is explained in §3.3. This strategy is inverted in the sense that the number of colours, x say, is fixed and an attempt is then made to minimize the maximum degree of the colour-induced subgraphs, *i.e.* one wants to minimize the maximum degree over all the colour-induced subgraphs. This smallest maximum degree for a graph G is denoted by $D_x^\Delta(G)$. In the remainder of §3.3 basic properties of the parameters $\chi_d^\Delta(G)$ and $D_x^\Delta(G)$ are derived. Objective II in §1.5 is therefore addressed in §3.1 & §3.3.

In Chapter 4 the results mentioned in objectives III and IV in §1.5 (*i.e.* either the exact value of the maximum degree chromatic number, $\chi_d^\Delta(G)$, or an upper bound on the maximum degree chromatic number for various graph classes, including bipartite graphs (§4.1), cycles and wheels (§4.2), complete graphs (§4.3), products of paths, cycles and complete graphs (§4.4), some circulants (§4.5) and complete balanced multipartite graphs (§4.6)) are presented.

The topic of Chapter 5 is the development of algorithms for producing maximum degree colouring. In this regard, two heuristics that determine an upper bound on the parameter $\chi_d^\Delta(G)$ relatively fast are presented in §5.1, thus addressing objective V(b) in §1.5. Two exact algorithms for determining the exact value of the parameter $\chi_d^\Delta(G)$ are presented in §5.2, thus addressing objective V(a) in §1.5. Results obtained by these four algorithms for 196 benchmark instances are also presented and compared to one another.

The main focus of Chapter 6 is attempting to identify sufficient conditions for an integral sequence satisfying known necessary conditions from the literature for the sequence $\{\chi_d^\Delta(G) : d \in \mathbb{N}_0\}$ of a general graph G , to be the maximum degree chromatic sequence of some graph. This is done by first determining in §6.1 a representative subset of integral sequences, called basic sequences and satisfying the aforementioned necessary conditions, which suffice in the study of maximum degree chromatic sequences. Basic sequences that may easily be classified as maximum degree chromatic sequences are discussed in §6.2, while the remainder of the chapter is devoted to searching for graph constructions with maximum degree chromatic sequences coinciding with unclassified basic sequences. Hence objective VI in §1.5 is addressed in Chapter 6.

The dissertation concludes in Chapter 7 with a summary of achievements in §7.1 and a discussion of related further work and open problems in §7.2.

Chapter 2

Fundamental Concepts and Notation

“Every contrivance of man, every tool, every instrument, every utensil, every article designed for use, of each and every kind, evolved from a very simple beginning.”

Robert Collier (1885–1950)

This chapter provides the reader with the necessary graph theoretic definitions in order to understand the concepts and ideas employed throughout the remainder of this dissertation. In §2.1 certain fundamentals from graph theory are discussed, followed by an overview of certain basic complexity theoretic concepts in §2.2, as well as a description of a number of basic graph colouring definitions in §2.3. The chapter closes with a short literature survey of some generalized colourings in §2.4.

2.1 Basic Definitions in Graph Theory

A **simple graph** $G = (V, E)$ is a finite, nonempty set $V(G)$ of elements called **vertices**, together with a (possibly empty) set $E(G)$ of 2–element subsets of $V(G)$, called **edges**, where uv (or vu) denotes the edge between a vertex $u \in V(G)$ and a vertex $v \in V(G)$ [109, p 5]. The cardinality of the **vertex set** $V(G)$ of a graph G is called the **order** of G and is denoted $p(G)$, while the cardinality of the **edge set** $E(G)$ is called the **size** of G and is denoted $q(G)$. There is only one graph of order 1, which is called the **trivial graph** [53, p 2]. Thus, for any **nontrivial graph**, G , $p(G) \geq 2$ [29, p 10]. If a graph has order p and size $q \leq \binom{p}{2}$, it is often referred to as a (p, q) –graph [52, p 4]. The **density** of a graph G is the ratio between the size of the graph G of specific order and the size of a same–order graph containing every possible edge. This means that the density of a graph G is $q(G)/\binom{p(G)}{2}$ [9, p 123]. If $e = uv$ is an edge of the graph G , the vertices u and v are said to be **adjacent** in G and the edge e **joins** u and v . Otherwise, two vertices that are not joined by an edge are said to be **nonadjacent** [109, p 6]. A graph may be represented graphically by means of points and lines, where the points represent the vertices, and a line between two points (vertices) indicates adjacency [29, p 2]. A graphical representation of a graph G_1 with vertex set $V(G_1) = \{v_1, \dots, v_7\}$, edge set $E(G_1) = \{v_1v_3, v_1v_4, v_1v_6, v_1v_7, v_2v_3, v_2v_7, v_3v_6, v_3v_7\}$ and a density of $\frac{8}{21} \approx 0.3810$ is shown in Figure 2.1(a) as an example of a $(7, 8)$ –graph. If $e = uv$ is an edge of the graph G , the vertex u and the edge e (as well as v and e) are said to be **incident** with each other. The edges e and f are said to be **adjacent edges** if e and f are two different edges that are incident with the same vertex [31, p 13]. In the graph G_1 in Figure 2.1(a) the edge v_1v_3 is incident with the vertex v_1 , while the edges v_1v_3 and v_1v_4 are adjacent edges.

An **open neighbourhood set**, $N_G(v) \subseteq V(G)$, of a vertex v in a graph G is defined as the set

$$N_G(v) = \{u \in V(G) | uv \in E(G)\}$$

while the **closed neighbourhood set**, $N_G[v] \subseteq V(G)$, of a vertex v in a graph G is defined as the set

$$N_G[v] = N_G(v) \cup \{v\}.$$

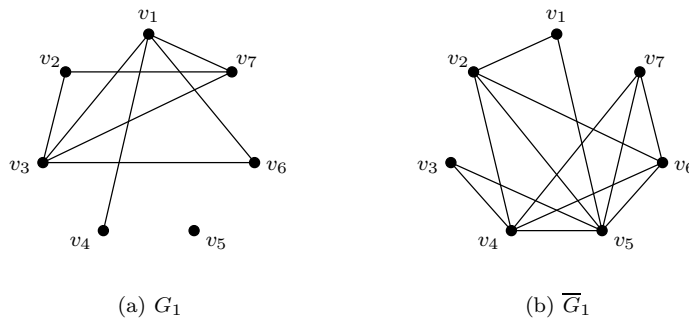


FIGURE 2.1: (a) A graphical representation of a $(7,8)$ -graph, G_1 , with vertex set $V(G_1) = \{v_1, \dots, v_7\}$, edge set $E(G_1) = \{v_1v_3, v_1v_4, v_1v_6, v_1v_7, v_2v_3, v_2v_7, v_3v_6, v_3v_7\}$ and a density of $\frac{8}{21}$. The vertices v_1 and v_2 are nonadjacent, while v_1 and v_3 are adjacent vertices. (b) A graphical representation of the complement, \overline{G}_1 , of the graph G_1 .

For any vertex v in a graph G , the number of vertices adjacent to v , i.e. $|N_G(v)|$, is called the **degree** of v in G and is denoted by $\deg_G(v)$. A vertex v with $\deg_G(v) = 0$ is called an **isolated vertex** of G , while a vertex with $\deg_G(v) = 1$ is called an **end-vertex** of G [9, p 3 & 4]. A vertex is called **odd** or **even** depending on whether its degree is odd or even [31, p 14]. The **minimum degree** of an order p graph G is defined as $\delta(G) = \min_{1 \leq i \leq p} \{\deg_G(v_i)\}$, while the **maximum degree** is defined as $\Delta(G) = \max_{1 \leq i \leq p} \{\deg_G(v_i)\}$, where $V(G) = \{v_1, \dots, v_p\}$ in both cases [52, p 4]. Referring to the graph G_1 in Figure 2.1(a), the open neighbourhood of the vertex v_7 is $N_{G_1}(v_7) = \{v_1, v_2, v_3\}$, while its closed neighbourhood is $N_{G_1}[v_7] = \{v_1, v_2, v_3, v_7\}$. Vertex v_5 is an isolated vertex, while vertex v_4 is an end-vertex. Furthermore, $\delta(G_1) = 0$ since $\deg_{G_1}(v_5) = 0 \leq \deg_{G_1}(v_i)$ for all $i = 1, \dots, 7$ and $\Delta(G_1) = 4$ since $\deg_{G_1}(v_3) = 4 \geq \deg_{G_1}(v_i)$ for all $i = 1, \dots, 7$. The following result, often referred to as the fundamental theorem of graph theory, is probably one of the best known results in the discipline and relates the sum total of the degrees and the number of edges in any graph [29, p 6].

Theorem 2.1 (Fundamental Theorem of Graph Theory) *Let G be a (p, q) -graph and let $V(G) = \{v_1, \dots, v_p\}$ be the set of vertices. Then*

$$\sum_{i=1}^p \deg_G(v_i) = 2q.$$

Proof: When the degrees of the vertices of G are summed, each edge is counted twice, once for each of the vertices that it joins [109, p 11]. ■

An intuitive consequence of Theorem 2.1 is given in the following corollary for which the proof may be found in [53, p 7].

Corollary 2.1 *Every graph contains an even number of odd vertices.* ■

The **complement** \overline{G} of a graph $G(V, E)$ is a graph with $V(\overline{G}) = V(G)$ and for which $e \in E(\overline{G})$ if and only if $e \notin E(G)$ [52, p 11]. Therefore, the complement of the graph G_1 in Figure 2.1(a) has the vertex set $V(\overline{G}_1) = V(G_1)$ and the edge set $E(\overline{G}_1) = \{v_1v_2, v_1v_5, v_2v_4, v_2v_5, v_2v_6, v_3v_4, v_3v_5, v_4v_5, v_4v_6, v_4v_7, v_5v_6, v_5v_7, v_6v_7\}$. \overline{G}_1 is shown in Figure 2.1(b).

Two graphical representations of the same graph may look completely different. Their similarities may only become noticeable by re-labelling the vertices in one of the representations. This leads to the concept of an isomorphism. Two graphs G and H are called **isomorphic**, denoted by $G \cong H$, if there exists a one-to-one and onto function $\phi : V(G) \rightarrow V(H)$ such that $uv \in E(G)$ if and only if $\phi(u)\phi(v) \in E(H)$, that is, the **isomorphism** ϕ preserves adjacency [9, p 3]. An isomorphism from a graph onto itself is called an **automorphism** of the graph [109, p 5]. Two graphs G and H are said to be **equal** if $V(G) = V(H)$ and

$E(G) = E(H)$. Therefore, graphs that are equal are also isomorphic, but the converse is not necessarily true [29, p 10]. An illustration of the notions of an isomorphism between and of equality of graphs may be found in Figure 2.2.

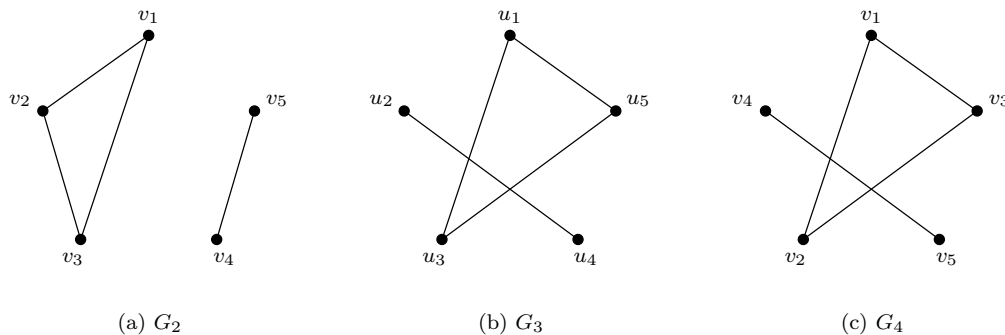


FIGURE 2.2: The graph G_3 in (b) is isomorphic (but not equal) to G_2 in (a), an isomorphism being $\phi(v_1) = u_1$, $\phi(v_2) = u_5$, $\phi(v_3) = u_3$, $\phi(v_4) = u_4$ and $\phi(v_5) = u_2$. The graph G_4 in (c) is both equal and isomorphic to G_2 .

Any graph H for which $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$ is called a **subgraph** of the graph G , denoted $H \subseteq G$, while a subgraph H is called a **spanning subgraph** of G if $V(H) = V(G)$ and $E(H) \subseteq E(G)$ [9, p 2]. Clearly, a graph G is a subgraph of itself, and a graph H is called a **proper subgraph** of a graph G , denoted $H \subset G$, if $H \neq G$ [57, p 25]. For a nonempty vertex subset $S \subseteq V(G)$ of a graph G the so-called **(vertex-) induced subgraph** of S in G , denoted by $\langle S \rangle_G$, is the subgraph of G with vertex set S and edge set $E(\langle S \rangle_G) = \{uv \in E(G) \mid u, v \in S\}$ [52, p 6]. Let $S \subset V(G)$ be a nonempty proper subset of the vertex set $V(G)$ of a graph G . Then the **degeneracy** of G is defined as $\rho(G) = \max_{S \subset V(G)} \delta(\langle S \rangle_G)$ [78]. The notions of a subgraph and of a spanning subgraph are illustrated in Figure 2.3. For a given graph F , a graph G is called **F -free** if G does not contain an induced subgraph isomorphic to F [29, p 290]. For example, the graph in Figure 2.3(c) does not contain any triangles, and is therefore a triangle-free graph, where a **triangle** in a graph consists of three vertices u, v and w , joined by means of the edges uv, uv and vw . A family of graphs \mathcal{G} is **hereditary** if every induced subgraph of a graph G in \mathcal{G} is also a graph in \mathcal{G} [112, p 226].

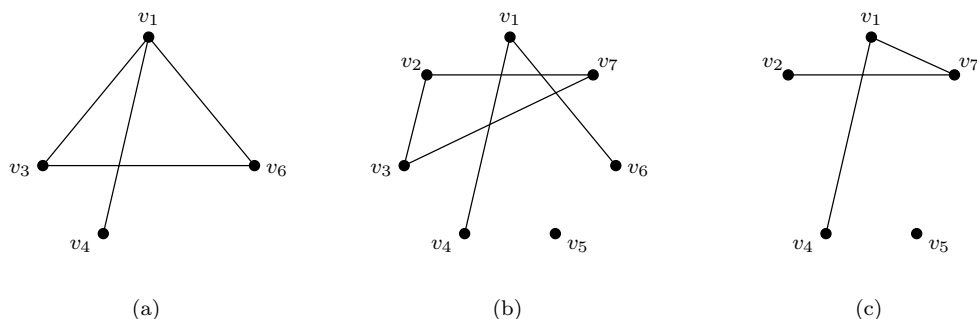


FIGURE 2.3: The graph shown in (a) is an example of a subgraph of G_1 (Figure 2.1(a)), while the graph in (b) is a spanning subgraph of G_1 . The induced subgraph $\langle \{v_1, v_2, v_4, v_5, v_7\} \rangle_{G_1}$ is shown in (c).

A $v_1 - v_n$ **walk** W of **length** $n - 1$ in a graph G is a finite alternating sequence

$$W : v_1, e_1, v_2, e_2, \dots, v_{n-1}, e_{n-1}, v_n \quad (n \geq 0)$$

of vertices and edges that begins with the vertex v_1 and ends with the vertex v_n , such that $e_i = v_i v_{i+1}$ is an edge of G for all $i = 1, 2, 3, \dots, n - 1$ (note that vertices and edges may appear more than once in a walk). For the sake of brevity the edges are omitted from the sequence when denoting a walk [31, p 25].

An example of a walk in the graph G_1 in Figure 2.1(a) is $v_1, v_3, v_2, v_7, v_1, v_4$. A $v_1 - v_n$ **path** in a graph G is a $v_1 - v_n$ walk in G in which no vertex is repeated. For a given graph G , the **path number**, $\tau(G)$, is the order of a longest path in G . The **distance** $d_G(u, v)$ between two vertices u and v in a graph G , is the length of a shortest $u - v$ path in G if such a path exists, otherwise $d_G(u, v) = \infty$ [57, p 28]. A graph of order n that consists of only a path is called the **path of order n** and is denoted by P_n . If $v_1 = v_n$ in the walk W and no other vertices in W are repeated, the walk W is called a **cycle** and has a length of $n - 1 \geq 3$ [9, p 5]. If a graph contains no cycles it is called **acyclic** [31, p 47]. A graph of order n that consists of only a cycle is called the **cycle of order n** and is denoted by C_n . A cycle of odd length [even length] is referred to as an **odd cycle** [**even cycle**] [52, p 48]. The length of a shortest cycle (if any) in a graph G is referred to as the **girth** of G , denoted by $g(G)$. An acyclic graph has infinite girth [112, p 13]. In the graph G_1 of Figure 2.1(a), $W : v_6, v_3, v_7, v_1, v_4$ is an example of a path of length 4, while $W : v_2, v_3, v_1, v_7, v_2$ is an example of a cycle of length 4 in G_1 , but $g(G_1) = 3$.

If there exists a $u - v$ path for every vertex pair u, v of a graph, the graph is said to be **connected**. Conversely, a graph that contains at least one vertex pair u, v for which there exists no $u - v$ path, is called **disconnected** [29, p 21]. A graph is referred to as **k -connected** if at least k vertices must be removed from the graph before the graph is disconnected [42, p 18]. A **tree** of order n is defined as a connected, acyclic graph. A tree of order n has size $n - 1$ [109, p 44]. A tree is minimally connected in the sense that a graph G with $q(G) < p(G) - 1$ cannot be connected [31, p 50]. Also, a **spanning tree** of a graph G is a spanning subgraph of G that is a tree [109, p 46]. Any nontrivial tree has at least two end-vertices and all end-vertices of a tree are called **leaves**, while the remaining vertices of the tree are called **internal vertices** [53, p 94]. If exactly one internal vertex of a tree is distinguished, the tree is called a **rooted tree** and the distinguished vertex is referred to as the **root** of the tree [42, p 32]. Furthermore, a rooted tree will be considered to be **leveled**, *i.e.* level 0 contains the root, r , of the tree, level 1 consists of all vertices in $N_G(r)$, level 2 consists of all vertices in $N_G(v) \setminus \{r\}$ for all vertices v on level 1, etc. If vertex v is on level k , then all vertices in $N_G(v)$ which are on level $(k + 1)$, are called the **children** of the vertex v . Finally, the length of a longest $r - v$ path from the root, r , to a leaf, v , in a rooted tree, is called the **height** of the tree [52, pp 85–86]. The notions of the root, the leaves, the internal vertices, the children, the levels and the height of a rooted tree are illustrated in Figure 2.4.

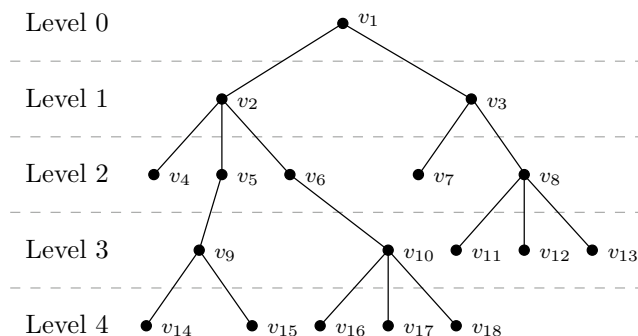


FIGURE 2.4: The vertices $v_1, v_2, v_3, v_5, v_6, v_8, v_9, v_{10}$ are the internal vertices of the rooted tree in the figure, while the vertices $v_4, v_7, v_{11}, \dots, v_{18}$ are the leaves of the tree. The vertices v_4, v_5 and v_6 are the children of the vertex v_2 and vertex v_1 is the root of the tree. The height of the tree is 4.

A **maximal connected subgraph** of a graph G is a subgraph of G that is connected and is not a subgraph of any other connected subgraph of G . A subgraph H of a graph G is also called a **component** of G if H is a maximal connected subgraph of G . The number of components of a graph G is denoted by $k(G)$ [112, p 22]. The graph G_2 in Figure 2.2(a) is disconnected, because there exists no path between, for example, the vertices v_1 and v_5 . Furthermore, $k(G_2) = 2$ and the graphs $\langle \{v_1, v_2, v_3\} \rangle_{G_2}$ and $\langle \{v_4, v_5\} \rangle_{G_2}$ are the two components of G_2 . The **deletion** of a nonempty vertex subset $S \subseteq V(G)$ from a graph G is the subgraph with vertex set $V(G) \setminus S$ and edge set $\{uv \in E(G) \mid u, v \notin S\}$. Such a subgraph is denoted by $G - S$. Similarly, for any edge subset $J \subseteq E(G)$ the **deletion** of the edge set J is the spanning subgraph of G with edge set $E(G) \setminus J$ and is denoted by $G - J$. If $S = \{v\}$ [$J = \{e\}$] for some $v \in V(G)$ [$e \in E(G)$], the graph $G - S$ [$G - J$] is simply denoted by $G - v$ [$G - e$]. The **addition** of a nonempty

edge subset $J \subseteq E(\overline{G})$ to a graph G is the graph obtained from G with vertex set $V(G)$ and edge set $E(G) \cup J$. Such a graph is denoted by $G + J$. If $J = \{e\}$ for some $e \in E(G)$, the graph $G + J$ is simply denoted by $G + e$ [9, p 2]. An edge e in a graph G is called a **bridge** of G if $k(G - e) > k(G)$. A vertex v in a graph G is called a **cut-vertex** of G if $k(G - v) > k(G)$ [31, p 53 & 78]. An illustration of the notions of a bridge and a cut-vertex may be found in Figure 2.5. The following theorem characterizes those edges of a graph that are bridges and a proof of the theorem may be found in [29, p 23].

Theorem 2.2 (Characterization of bridges in connected graphs) *An edge e of a connected graph G is a bridge of G if and only if e does not lie on a cycle of G .* ■

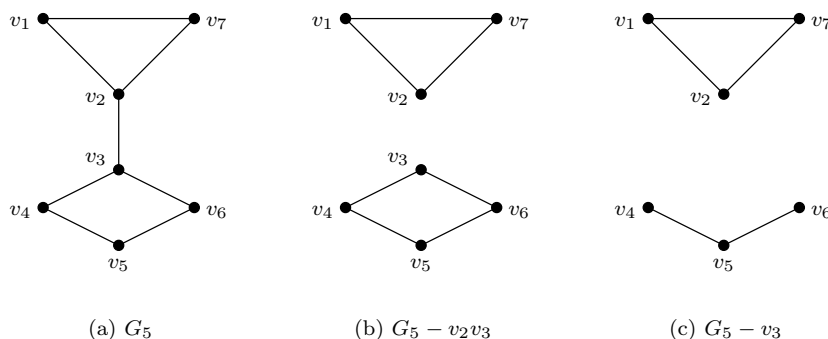


FIGURE 2.5: The graph G_5 in (a) has $k(G_5) = 1$, i.e. G_5 is a connected graph. (b) The edge v_2v_3 in G_5 is a bridge, since $k(G_5 - v_2v_3) = 2 > k(G_5)$. (c) The vertex v_3 is a cut-vertex, since $k(G_5 - v_3) = 2 > k(G_5)$.

A graph may be constructed from other graphs in several ways. The **union** of two graphs G and H with disjoint vertex sets, denoted by $G \cup H$, is the graph with vertex set $V(G \cup H) = V(G) \cup V(H)$ and edge set $E(G \cup H) = E(G) \cup E(H)$, while the union of n isomorphic copies of a graph G is denoted by nG [109, p 9]. The **join** of two graphs G and H with disjoint vertex sets, denoted by $G + H$, is the union of G and H together with all edges uv , where $u \in V(G)$ and $v \in V(H)$ [52, p 11]. The **cartesian product** of two graphs G and H with disjoint vertex sets, denoted by $G \times H$, is the graph with vertex set $V(G \times H) = V(G) \times V(H)$, and two vertices (u_1, u_2) and (v_1, v_2) being adjacent in $G \times H$ if and only if either $u_1 = v_1$ and $u_2v_2 \in E(H)$, or $u_2 = v_2$ and $u_1v_1 \in E(G)$. The cartesian product of n isomorphic copies of a graph G is denoted by G^n . From the symmetry in the definitions it follows that $G \cup H \cong H \cup G$, $G + H \cong H + G$ and $G \times H \cong H \times G$ [29, p 29]. The notions of a graph union, a graph join and a graph cartesian product are illustrated in Figure 2.6.

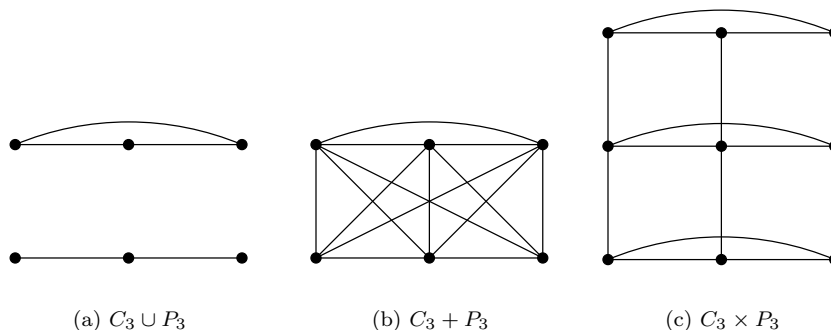


FIGURE 2.6: (a) The union of the graphs C_3 and P_3 , (b) the join of the graphs C_3 and P_3 and (c) the cartesian product of the graphs C_3 and P_3 .

If $\deg_G(v) = r$ for all vertices $v \in V(G)$ and some integer $0 \leq r \leq p(G) - 1$, then the graph G is said to be **r -regular**, or **regular of degree r** . In general, a graph is said to be **regular** if it is r -regular for

some value of $r \in \mathbb{N}_0$ [31, p 15]. A **matching** of a graph G is a 1-regular subgraph of G . A matching M of a graph G for which $E(M)$ is of maximum cardinality, is called a **maximum matching** of G , while the **matching number** $\nu(G)$ denotes the size of a maximum matching of G . A **perfect matching** of G , if it exists, is a matching of G in which every vertex of G is incident with an edge in the matching, *i.e.* $\nu(G) = p(G)/2$ [29, p 162].

A graph G is said to be **vertex-transitive** if, for all vertex pairs $u, v \in V(G)$, there is an automorphism of G that maps u to v [53, p 492]. Simplistically this means that “every vertex is like every other vertex” in a vertex-transitive graph (at the very least, vertex-transitive graphs have to be regular). Vertex-transitivity of a graph may be advantageous in certain applications. For example, given a certain (possibly greedy) algorithm to perform some task involving vertices of the graph, any vertex may be used for initialisation of the algorithm without loss of generality or efficiency of the algorithm in the case of a vertex-transitive graph. The notions of a regular graph, vertex-transitivity of a graph and a matching of a graph are illustrated in Figure 2.7.

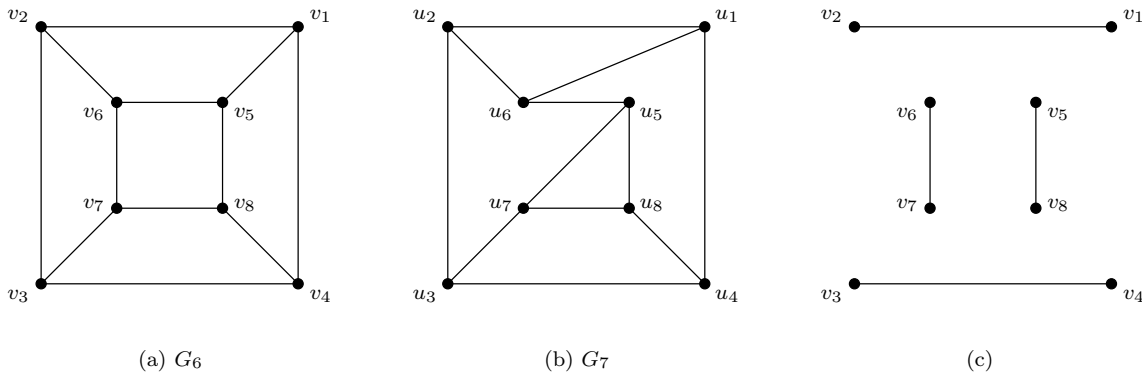


FIGURE 2.7: The graph G_6 , shown in (a), is a vertex-transitive, 3-regular graph, while the graph G_7 in (b) is also a 3-regular graph, but not vertex-transitive (since, for example, vertex u_3 forms part of no 3-cycle, while 3-cycles exist in G_7). (c) A perfect matching of G_6 and therefore $\nu(G_6) = 4$.

In a **complete graph** of order n , denoted by K_n , every distinct pair of vertices are adjacent. The complete graph K_n is therefore $(n-1)$ -regular [31, p 15]. Also, it is clear that K_n has $\binom{n}{2}$ edges [57, p 14]. The complete graph K_6 is shown in Figure 2.8(a). Contrasting a complete graph of order n , the graph \overline{K}_n of order n satisfies $E(\overline{K}_n) = \emptyset$. \overline{K}_n is called the **edgeless graph** of order n . The **empty graph**, denoted by K_0 , is the graph such that $V(K_0) = \emptyset$ and $E(K_0) = \emptyset$ [112, p 3].

A graph G is said to be k -**partite**, $k \geq 2$, if it is possible to partition the vertex set $V(G)$ into k nonempty subsets V_1, \dots, V_k , called **partite sets**, such that no two vertices of V_i are adjacent for $i = 1, \dots, k$. If $k = 2$, then the graph G is called a **bipartite graph**; otherwise it is also often called a **multipartite graph** [9, p 6]. The following theorem relates bipartiteness and the occurrence of cycles in a graph, and a proof of the theorem may be found in [29, p 27] or [109, p 16].

Theorem 2.3 (Bipartiteness and the occurrence of cycles in a graph) *A nontrivial graph G is bipartite if and only if G has no odd cycles.* ■

If every vertex in a partite set V_i , for all $i = 1, \dots, k$, of a k -partite graph is adjacent to every vertex $v \notin V_i$, then the graph is called a **complete k -partite/multipartite graph** (**complete bipartite graph** if $k = 2$) and is denoted by K_{n_1, \dots, n_k} , where $|V_i| = n_i$ for $i = 1, \dots, k$ [9, p 7]. The complete k -partite graph K_{n_1, \dots, n_k} may therefore also be viewed as the join of k edgeless graphs, *i.e.* $\overline{K}_{n_1} + \dots + \overline{K}_{n_k}$ [109, p 9]. If $n_1, \dots, n_k = n$ (say), then the graph is called a **complete, balanced k -partite/multipartite graph** (**complete, balanced bipartite graph** if $k = 2$) and is denoted $K_{k \times n}$ [52, p 12]. The complete bipartite graph $K_{2,3}$ and the complete balanced multipartite graph $K_{3 \times 2}$ are shown in Figure 2.8(b) and (c) respectively. Furthermore, the complete bipartite graph $K_{1, n-1}$ is a popular graph, called an n -**star** [112, p 67] and $K_{1,5}$ is shown in Figure 2.9(a).

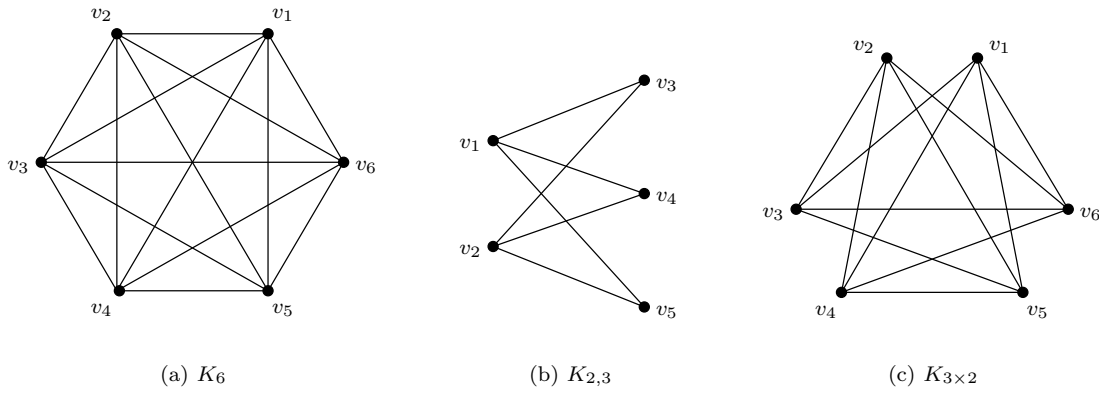


FIGURE 2.8: Examples of complete graphs. (a) The complete graph K_6 , (b) the complete bipartite graph $K_{2,3}$ and (c) the complete, balanced multipartite graph $K_{3 \times 2}$.

Consider the cycle of order $n - 1 \geq 3$, C_{n-1} , and let $V(C_{n-1}) = \{v_1, \dots, v_{n-1}\}$. Let v_n be another vertex. Then the **wheel** W_n of order n may be defined as the graph join $C_{n-1} + \langle v_n \rangle$ [112, p 174]. An example of W_6 is shown in Figure 2.9(b).

Let G_1, \dots, G_t be t graphs with $V(G_1) = \dots = V(G_t) = V$ and if $e \in E(G_i)$ for some $i \in \{1, \dots, t\}$, then $e \notin E(G_j)$ for all $j \in \{1, \dots, t\}$, $j \neq i$. The **edge union** of these graphs, denoted by

$$G = \bigoplus_{i=1}^t G_i = G_1 \oplus \dots \oplus G_t, \quad (2.1.1)$$

is the graph G with vertex set $V(G) = V$ and edge set $E(G) = \cup_{i=1}^t E(G_i)$. Each graph G_i in (2.1.1) is called a **factor** of G , while the expression (2.1.1) is called a **factorisation** of G [62, Chapter 1]. For example, the graph $C_3 \cup P_3$ in Figure 2.6(a) may be seen as a factor of the graph $C_3 + P_3$ in Figure 2.6(b), while a factorisation of the graph $C_3 + P_3$ is $(C_3 \cup P_3) \oplus K_{2 \times 3}$.

Suppose $n, z \in \mathbb{N}$, where $z < n$. Also, let $1 \leq i_1, \dots, i_z \leq n - 1$ be z distinct integers. Then the **circulant** $C_n \langle i_1, \dots, i_z \rangle$ of order n is defined as a graph with vertex set $V(C_n \langle i_1, \dots, i_z \rangle) = \{v_0, \dots, v_{n-1}\}$ and edge set $E(C_n \langle i_1, \dots, i_z \rangle) = \{v_\alpha, v_{\alpha+\beta(\text{mod } n)} \mid \alpha = 0, \dots, n - 1 \text{ and } \beta = i_1, \dots, i_z\}$. The inherent symmetry of the circulant graph $C_n \langle i_1, \dots, i_z \rangle$ may be visualised by arranging the vertex set $\{v_0, \dots, v_{n-1}\}$ on the edge of an imaginary circle and then joining every i_1 -th vertex, \dots , every i_z -th vertex on the circle by means of edges. The set $\{i_1, \dots, i_z\}$ is called the **connection set** of the circulant $C_n \langle i_1, \dots, i_z \rangle$ [62, Chapter 1]. One may view the edgeless graph of order n , \overline{K}_n , as the circulant of order n with an empty connection set, while the complete graph, K_n , is the circulant $C_n \langle 1, \dots, \lfloor n/2 \rfloor \rangle$ [110]. If $z = 1$ the circulant $C_n \langle i \rangle$ is said to be **elementary**, else it is called **composite**. From the symmetry of a circulant graph it follows that $C_n \langle i \rangle = C_n \langle n - i \rangle$. Elementary circulants may be categorised as follows:

(i) If $n \in \mathbb{N}$ and $i \in \mathbb{N}$ are relatively prime, then

$$C_n \langle i \rangle \cong C_n. \quad (2.1.2)$$

(ii) If n is even and $i = n/2$, then $C_n \langle i \rangle$ is called a **singular (elementary) circulant** and

$$C_n \langle i \rangle \cong \frac{n}{2} P_2. \quad (2.1.3)$$

(iii) Let m be the **greatest common divisor** of n and i , denoted $\text{gcd}(n, i)$, then

$$C_n \langle i \rangle \cong m C_r, \quad (2.1.4)$$

where $r = n/m$ [58, p 7 & 9]. A composite circulant may be viewed as a construction from two or more elementary circulants via an edge union, *i.e.*

$$C_n \langle i_1, \dots, i_z \rangle = \bigoplus_{k=1}^z C_n \langle i_k \rangle.$$

A circulant $C_n\langle i_1, \dots, i_z \rangle$ is **singular** if $i_j = n/2$ for some $j \in \{1, \dots, z\}$, and **non-singular** otherwise. A singular circulant is $(2z - 1)$ -regular and has size $(z - 1)n + n/2$, while a non-singular circulant is $2z$ -regular and has size zn [62, Chapter 1]. Furthermore, $k(C_n\langle i_1, \dots, i_z \rangle) = m$, where $m = \gcd(n, i_1, \dots, i_z)$. Hence, the circulant $C_n\langle i_1, \dots, i_z \rangle$ is connected if and only if $\gcd(n, i_1, \dots, i_z) = 1$ [58, p 5 & 6]. The 4-regular non-singular composite circulant $C_8\langle 1, 2 \rangle$ is given in Figure 2.9(c) and is connected, since $\gcd(8, 1, 2) = 1$.

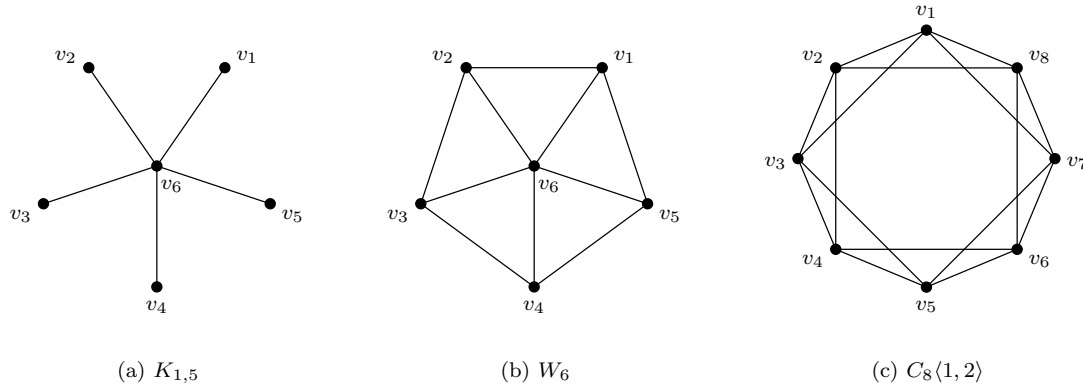


FIGURE 2.9: Examples of some special graphs. (a) The 6-star $K_{1,5}$, (b) the wheel of order 6, W_6 , and (c) the non-singular, composite circulant $C_8\langle 1, 2 \rangle$.

A **(vertex) independent set** $I \subseteq V(G)$ of a graph G is a vertex subset with the property that, for any pair of vertices $u, v \in I$, u and v are not adjacent in G , *i.e.* the vertex induced subgraph $\langle I \rangle_G$ is edgeless [109, p 7]. An independent set I is said to be **maximal independent** if no proper superset $I' \subseteq V(G)$ of I , is itself an independent set of G . The maximum cardinality of a maximal independent set of vertices of a graph G is called the **(upper) independence number** of G and is denoted by $\beta(G)$ [42, p 127]. The set $I_1 = \{u_1, u_7\}$ is an independent set of the graph G_7 in Figure 2.7(b), although it is not maximal independent. The set $I_2 = \{u_1, u_3, u_5\}$, however, is a maximal independent set (of maximum cardinality) for G_7 , because no proper superset of I_2 yields an independent set for G_7 and no independent set of cardinality 4 exists for G_7 , implying that $\beta(G_7) = 3$. Contrasting the notion of independence is the notion of a **clique**, which is a complete subgraph of a graph G . The maximum order of a clique in a graph G is the so-called **clique number** of G and is denoted by $\omega(G)$ [31, p 208]. Note that, for any graph G , $\omega(G) = \beta(\overline{G})$ [29, p 284]. The minimum number of cliques into which a graph G may be partitioned is known as the **clique partition number** of G and is denoted by $\mathfrak{c}(G)$. The induced graph $\langle v_4, v_5, v_6, v_7 \rangle_{\overline{G}_1} \cong K_4$ is the largest clique in the graph \overline{G}_1 in Figure 2.1(b), so that $\omega(\overline{G}_1) = 4$, while $\mathfrak{c}(\overline{G}_1) = 3$ and $\langle v_4, v_5, v_6, v_7 \rangle_{\overline{G}_1} \cong K_4$, $\langle v_1, v_2 \rangle_{\overline{G}_1} \cong K_2$ and $\langle v_3 \rangle_{\overline{G}_1} \cong K_1$ is an example of a partition of \overline{G}_1 into three cliques.

A graph that *may* be drawn in the plane without any of its edges intersecting (except possibly at vertices) is called a **planar graph** [9, p 20]. A planar graph that is already drawn in the plane without any of its edges intersecting, is said to be **embedded** in the plane and is called a **plane graph** [52, p 55]. Of course a given planar graph can give rise to several different plane graph representations [112, p 235]. A graph that cannot be embedded in the plane is said to be **nonplanar** [42, p 55]. The graph G_4 in Figure 2.2(c) is a planar graph, but not a plane graph. The graph G_2 in Figure 2.2(a), which is a different graphical representation of G_4 in Figure 2.2(c), is a plane graph.

Until now, a graph G was represented as either a set pair (V, E) , or graphically by means of points and lines, but to conclude this section, an alternative way of representing a graph is given, namely the so-called **adjacency matrix** $A(G)$ of a graph G . The adjacency matrix is usually used to represent graphs in computer memory (although other more efficient memory representations exist, see for example [40, p 4] and [53, p 74]), since algorithms involving graph manipulations may easily be performed with the aid of computers. The adjacency matrix for an order p graph G is a $p \times p$ symmetric, binary matrix with zeros down the main diagonal, where $a_{ij} = 1$ if and only if vertex v_i is adjacent to vertex v_j in G ($v_i \neq v_j$) and $a_{ij} = 0$ otherwise [112, p 6]. In the case of a circulant graph $C_n\langle i_1, \dots, i_z \rangle$ the adjacency matrix

is a circulant matrix (hence the name circulant graph), *i.e.* the adjacency matrix has the additional characteristic that each row vector is rotated one element to the right relative to the preceding row vector, *i.e.* $a_{ij} = a_{i-1, j-1}$ if $j > 1$ and $a_{i-1, n}$ if $j = 1$ [65]. The adjacency matrix of the graph G_6 in Figure 2.7(a) is given by

$$A(G_6) = \begin{matrix} & v_1 & v_2 & v_3 & v_4 & v_5 & v_6 & v_7 & v_8 \\ \begin{matrix} v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \\ v_7 \\ v_8 \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} & \end{matrix}.$$

2.2 Complexity of Graph Algorithms

A large number of graph parameters may be determined by algorithms and therefore practical applicability (efficiency) of a specific algorithm in terms of the computational speed and the amount of computer memory required to execute the algorithm to solve the problem at hand, is of vital importance. **Algorithmic complexity** is the process of analysing and comparing algorithms, ignoring implementation details that are often outside a given programmer's control. These factors include the computer on which the algorithm is run, the operating system, and the machine code translation of the programming language in which the algorithm is written. The objective of algorithmic complexity is rather to determine the functional dependence of the running time and memory space used during execution of the algorithm on the size n of the input to the algorithm [53, p 554]. In the case of graph algorithms, the input size of the algorithm is often the order of the graph, the size of the graph or the sum of the two [109, p 185].

Algorithmic complexity is usually measured by two functional variables: the **time complexity** $T(n)$ and **space complexity** $S(n)$ of the algorithm, where n refers to the size of the input to the algorithm. The functions $T(n)$ and $S(n)$ measure respectively the number of basic operations performed by the algorithm and the amount of memory required by the algorithm in terms of n [62, Chapter 3]. The **worst-case complexity** of an algorithm (meaning that the largest possible values of $T(n)$ and $S(n)$ for any problem instance of input size n is considered during complexity analyses, as opposed to an expected (average) complexity of an algorithm [42, p 149]) is considered in this section. As an example, consider Algorithm 1 which computes the maximum degree of a graph.

Algorithm 1 Maximum degree algorithm

Input: The $n \times n$ adjacency matrix $A(G)$ of a graph G of order n .

Output: A single integer value Δ , namely the maximum degree of G .

```

1:  $\Delta \leftarrow 0$ 
2: for all  $i = 1, \dots, n$  do
3:    $d \leftarrow 0$ 
4:   for all  $j = 1, \dots, n$  do
5:      $d \leftarrow d + a_{ij}$ 
6:   end for
7:   if  $d > \Delta$  then
8:      $\Delta \leftarrow d$ 
9:   end if
10: end for
11: return  $\Delta$ 

```

The order n of the input graph G may be taken as the size of the input to Algorithm 1. Let a space unit be the amount of memory required to store the value of an integer [62, Chapter 3]. Then, Algorithm 1 requires n^2 space units to store the adjacency matrix $A(G)$ on input, as well as two additional space units to store the intermediate variable d and the output variable Δ respectively. Hence, the amount of

memory required to implement Algorithm 1 is $S(n) = n^2 + 2$ space units. In Algorithm 1 the addition of two integers, variable assignment and the comparison of two integers may all be defined as basic operations [62, Chapter 3]. One variable assignment occurs in each of Steps 1, 3, 5 and 8. Step 1 is executed once only, Step 3 is repeated n times, Step 5 is repeated n^2 times and Step 8 is executed *at most* n times. Thus, the total number of variable assignments is at most $n^2 + 2n + 1$. Furthermore, one integer addition occurs in Step 5 which is repeated n^2 times, resulting in a total of n^2 integer additions. Finally, one comparison between two integers occurs in Step 7 and since Step 7 is repeated n times, an additional n basic operations are performed. Therefore, the time complexity of Algorithm 1 in terms of the number of basic operations required to execute the algorithm is $T(n) \leq 2n^2 + 3n + 1$.

Since Algorithm 1 is called only once (at the very start of the algorithm) during a single execution thereof and then iterates through a number of steps during execution, the algorithm is an example of a so-called **iterative algorithm** as opposed to a so-called **recursive algorithm**. A recursive algorithm also iterates through a number of steps, but the algorithm may perform calls to itself, called **recursive calls**, during execution of the algorithm. A smaller problem that is similar in structure to the original problem being solved by the first call to a recursive algorithm, is typically solved during each recursive call. Although recursive algorithms may often solve complex computation problems elegantly, space and time complexity analyses of such algorithms may be quite difficult and rather tedious [62, Chapter 3].

Instead of seeking the exact upper bounds on the functions $S(n)$ and $T(n)$ in algorithmic complexity analyses, it is usually sufficient to have asymptotic upper bounds on these two parameters, *i.e.* bounds that describe the worst-case growth behaviour of these two functions as $n \rightarrow \infty$. In this regard, let $f(n)$ and $g(n)$ be two functions from the set of positive integers to the set of real numbers. Then the function $f(n)$ is said to be of the **order** of the function $g(n)$, denoted by $f(n) = \mathcal{O}(g(n))$ (read $f(n)$ is “big O” of $g(n)$), if there are constants $c \in \mathbb{R}^+$ and $n_0 \in \mathbb{N}$ such that $0 \leq f(n) \leq c \cdot g(n)$ for all $n \geq n_0$ [109, p 186]. The function $f(n)$ is said to be **asymptotically dominated** by $g(n)$. In other words, $c \cdot g(n)$ is larger than $f(n)$ as $n \rightarrow \infty$ [53, p 554]. Consider, for example, the space complexity $S(n) = n^2 + 2$ associated with Algorithm 1, and let $g(n) = n^2$. Then, it is clear that $S(n)$ and $3/2 \cdot g(n)$ intersect where $n^2 - 4 = 0$, *i.e.* at the points $n = -2$ and $n = 2$, and that $0 \leq S(n) \leq 3/2 \cdot g(n)$ for all $n \geq 2$. Hence, $S(n) = \mathcal{O}(n^2)$. Similarly, for the time complexity $T(n) \leq 2n^2 + 3n + 1$ associated with Algorithm 1, it follows that $T(n)$ is also $\mathcal{O}(n^2)$, by taking $f(n) = T(n)$, $g(n) = n^2$, $c = 3$ and $n_0 = 4$ in the definition of the order notation.

It is easy to show that the space complexity $S(n)$ associated with Algorithm 1 is also $\mathcal{O}(n^k)$ for any $k \geq 2$, but $\mathcal{O}(n^2)$ is the sharpest among these asymptotic bounds and is therefore the desired choice for comparison purposes [53, p 554]. If an Algorithm A and an Algorithm B have time complexity functions $f(n)$ and $g(n)$ respectively, then Algorithm A is more desirable from an execution time point of view than Algorithm B if $f(n) = \mathcal{O}(g(n))$ but $g(n) \neq \mathcal{O}(f(n))$. On the other hand, if $f(n) = \mathcal{O}(g(n))$ and $g(n) = \mathcal{O}(f(n))$ then the functions $f(n)$ and $g(n)$ are said to be of the **same order** and either Algorithm A or Algorithm B may be implemented [29, p 40]. To test the order of two complexity functions $f(n)$ and $g(n)$ with respect to each other, it is often most convenient to examine $\lim_{n \rightarrow \infty} f(n)/g(n)$. If $\lim_{n \rightarrow \infty} f(n)/g(n) = c$ for some constant $c \in \mathbb{R} \setminus \{0\}$, then $f(n)$ and $g(n)$ are of the same order, and if $\lim_{n \rightarrow \infty} f(n)/g(n) = 0$, then $f(n) = \mathcal{O}(g(n))$, while $g(n) = \mathcal{O}(f(n))$ if $\lim_{n \rightarrow \infty} f(n)/g(n) = \infty$ [109, p 186].

An algorithm for which the time [space] complexity is asymptotically dominated by a linear function with respect to input size n , is referred to as a **linear time [space] algorithm** and is of order $\mathcal{O}(n)$, while the complexity is classified as **constant**, denoted $\mathcal{O}(1)$, if its complexity is independent of n . A **polynomial time [space] algorithm** is an algorithm whose execution time [memory space required] is of order $\mathcal{O}(p(n))$ for some polynomial function $p(n)$ [53, p 555]. Finally, an algorithm for which the complexity is of order $\mathcal{O}(a^{f(n)})$, where $a \in \mathbb{R}^+$ is constant, or $\mathcal{O}(n!)$ respectively, is referred to as an **exponential (time/space) algorithm** or **factorial (time/space) algorithm** respectively [42, p 150]. Polynomial time [space] algorithms (including constant and linear algorithms) are called **time- [space-] efficient** because, as the problem size increases, the number of basic operations [memory space] required by a polynomial algorithm, (as opposed to exponential or factorial algorithms) grow relatively slowly [29, p 39].

Trade-offs usually exist between the time and space complexities of an algorithm, in the sense that attempts at reducing the time complexity cause an increase in the memory required to execute the al-

gorithm (and hence the space complexity), and *vice versa*. All references to algorithmic complexity in this dissertation will be in terms of the time complexity of an algorithm (from now on merely called complexity for the sake of brevity). One should, however, not disregard the importance of space complexity — the space complexity of an extremely time-efficient algorithm often renders the algorithm impractical for large instances of the problem at hand [62, Chapter 3].

A problem formulated as a binary question that requires only a **true** or **false** answer is referred to as a **decision problem** [53, p 555]. A decision problem belongs to the class **P** (acronym for *Polynomial*) if there is a polynomial time algorithm to solve the problem and such a decision problem is called **tractable**. Conversely, if no such polynomial time algorithm is known, then the decision problem investigated is called an **intractable** or hard problem, since it seems unattainable to determine its solution within a reasonable amount of time as the problem input size increases [62, Chapter 3]. For example, given a graph G , the decision problem “Is G bipartite?” may be solved by a polynomial time algorithm and is therefore tractable and in the class **P** [53, p 555]. However, given two graphs G and H , no efficient algorithm is known to answer **true** or **false** to the question “Is G isomorphic to H ?” Hence, this decision problem is intractable [109, p 195].

The class **NP** (acronym for *Non-deterministic Polynomial*) comprises all decision problems that may be answered **true** by a polynomial time algorithm, given additional information (called the **certificate** to the decision problem instance at hand) [29, p 50]. These problems are termed non-deterministic, since no (deterministic) method is used to determine the certificate, one only claims that a certificate exists [109, p 195]. Note that although a certificate to the problem instance might exist, finding this certificate may be difficult for problems in the class **NP** [62, Chapter 3]. Clearly, every decision problem in the class **P** is also in the class **NP**, since for any decision problem in the class **P** an empty certificate would suffice and instead of verifying the correctness of a solution provided by a certificate, the correct answer may be obtained in polynomial time [29, p 54]. As an example, consider the following decision problem for a given connected graph G and an edge e of G : “Is e a bridge of G ?” It is possible to verify in polynomial time that e is indeed a bridge of G if the certificate of a pair of vertices $u, v \in V(G)$ such that no $u - v$ path exists in $G - e$, is given. Thus, the decision problem is in the class **NP**. However, the decision problem may, in fact, be answered in polynomial time, given no additional information (certificate) at all, by evaluating directly whether or not $k(G - e) = 1$. Hence the decision problem is also in the class **P** [62, Chapter 3].

Suppose \mathcal{D}_1 and \mathcal{D}_2 are two decision problems. Then \mathcal{D}_1 is said to be **polynomial time reducible** to \mathcal{D}_2 , denoted $\mathcal{D}_1 \rightsquigarrow \mathcal{D}_2$, if (1) there exists a function f transforming any instance I_1 of \mathcal{D}_1 to an instance $f(I_1)$ of \mathcal{D}_2 such that the answer to I_1 with respect to \mathcal{D}_1 is **true** if and only if the answer to $f(I_1)$ with respect to \mathcal{D}_2 is **true**, and (2) if there exists an efficient algorithm to implement the function f [40, p 198]. Polynomial time reducibility is transitive, as stated in the next theorem for which a proof may be found in [42, p 154].

Theorem 2.4 *Let \mathcal{D}_1 , \mathcal{D}_2 and \mathcal{D}_3 be decision problems. If $\mathcal{D}_1 \rightsquigarrow \mathcal{D}_2$ and $\mathcal{D}_2 \rightsquigarrow \mathcal{D}_3$, then $\mathcal{D}_1 \rightsquigarrow \mathcal{D}_3$. ■*

A decision problem \mathcal{D} is called **NP-complete** if $\mathcal{D} \in \mathbf{NP}$ and if all decision problems in **NP** is polynomial time reducible to \mathcal{D} [53, p 556]. **NP-complete** problems may be considered the hardest problems in **NP**, from a computational point of view, in the sense that a solution for a decision problem $\mathcal{D} \in \mathbf{NP}-**complete** provides a solution to all other decision problems in **NP** [52, p 18]. The non-emptiness of the class **NP-complete** was established in 1971 when Stephen Cook [32] proved that the so-called **satisfiability problem**, abbreviated as SAT, is **NP-complete**. Furthermore, if a decision problem \mathcal{D} is **NP-complete**, another decision problem \mathcal{D}' is in **NP** and $\mathcal{D} \rightsquigarrow \mathcal{D}'$, then \mathcal{D}' is also **NP-complete** by Theorem 2.4, so that SAT is often used to prove the **NP-completeness** of a decision problem [40, p 199–200].$

Let x_1, \dots, x_n be n boolean variables, where a **boolean variable** may take on one of only two values, say **true** or **false**. The **negation** \bar{x}_i of the boolean variable x_i is also a boolean variable which is **true** if and only if x_i is **false**. A **literal** is either one of the variables x_i or its negation \bar{x}_i [29, p 51]. A **clause** [s -**clause**] is a conjunction of literals [s literals], conjoined by means of the binary operation **or**, denoted by \vee , where typically $s \leq n$ and the resulting values for the binary operation **or** are given in Table 2.1. A boolean function of the boolean variables x_1, \dots, x_n , denoted $f(x_1, \dots, x_n)$, is said to be

in **conjunctive normal form** [*s*-**conjunctive normal form**] if the function consists of a number of clauses [*s*-clauses] conjoined by means of the binary operation **and**, denoted by \wedge , where the resulting values for the binary operation **and** are also given in Table 2.1 [62, Chapter 3]. For example, the function $f(x_1, x_2, x_3) = (x_1 \vee x_2) \wedge (x_1 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee \bar{x}_3) \wedge (\bar{x}_2 \vee x_3)$ is a boolean function in 2-conjunctive normal form comprising four clauses. The above definitions may now be used to define SAT. Given a boolean function $f(x_1, \dots, x_n)$ in conjunctive normal form, **SAT** is the decision problem “Does there exist an assignment of values to the boolean variables x_1, \dots, x_n for which the function f evaluates to **true**?” (*i.e.* “Is f **satisfiable**?”) Also, given a boolean function $f(x_1, \dots, x_n)$ in *s*-conjunctive normal form, **s-SAT** is the decision problem “Does there exist an assignment of values to the boolean variables x_1, \dots, x_n for which the function f evaluates to **true**?”

x_1	x_2	$x_1 \vee x_2$	$x_1 \wedge x_2$
false	false	false	false
false	true	true	false
true	false	true	false
true	true	true	true

TABLE 2.1: Truth table for the binary operations **or** (\vee) and **and** (\wedge).

A proof for the next theorem may be found in [40, p 200–204].

Theorem 2.5 *SAT is NP-complete.* ■

Since the existence of the **NP-complete** decision problem SAT was established, a large number of **NP** decision problems have been classified as **NP-complete**. The reader is referred to a landmark paper by Karp [74] in 1972 presenting twenty-one intractable combinatorial problems, including 3-SAT¹, that are all **NP-complete**, as well as a classic book by Garey and Johnson [47] on the theory of **NP-completeness** containing a large collection of **NP-complete** problems. A number of graph theoretic decision problems in the classes **P** and **NP-complete** related to topics within the scope of this dissertation are presented in Table 2.2.

In the class P	
Problem Name	Problem Formulation
Connectivity Problem	Is a given graph G connected? [62, Chapter 3]
Planarity Problem	Is a given graph G planar? [112, p 495]
Bridge Problem	For a given connected graph G and an edge e of G , is e a bridge of G ? [62, Chapter 3]
Girth Problem	For a given graph G and positive integer $k \leq p(G)$, is $g(G) \leq k$? [112, p 495]
Matching Number Problem	For a given graph G and positive integer $k \leq p(G)/2$, is $\nu(G) \geq k$? [42, p 180]
In the class NP-complete	
Problem Name	Problem Formulation
Longest Path Problem	For a given graph G and positive integer $k \leq p(G)$, is $\tau(G) \geq k$? [42, p 153]
Clique Number Problem	For a given graph G and positive integer $k \leq p(G)$, is $\omega(G) \geq k$? [109, p 196]
Independence Number Problem	For a given graph G and positive integer $k \leq p(G)$, is $\beta(G) \geq k$? [40, p 213]

TABLE 2.2: Graph theoretic decision problems in the classes **P** and **NP-complete**.

Although the discussions above were limited to decision problems, this is not an important restriction, since most computational problems may be recast as decision problems [109, p 195]. An optimization

¹Although 3-SAT is **NP-complete**, 2-SAT is in class *P* [112, p 500].

problem may, for example, be converted to a decision problem with the use of a bound (as was done in a number of decision problems discussed thus far) [42, p 174]. Furthermore, if an efficient algorithm to solve the decision problem exists, then the related computational problem may be solved efficiently too, and *vice versa* [40, p 195]. As an example, a proof of the **NP-completeness** of the Clique Number Problem which was achieved by polynomial time reducing an instance of 3-SAT to an instance of the Clique Number Problem, may be found in [62, Chapter 3]. Henning and Van Vuuren [62, Chapter 3] then demonstrate that the complexity of the decision problem and of the related computational problem are of the same order by efficiently determining the value of the clique number $\omega(G)$ of a graph G by means of a so-called interval halving scheme² in terms of the (decision) Clique Number Problem, where the execution of the procedure to solve the (decision) Clique Number Problem is also taken as a basic operation.

Knowing that a problem is **NP-complete** does not eliminate the need for an answer to the problem, but the line of approach towards the problem is usually adapted upon such knowledge. For example, one may attempt to find an efficient algorithm that solves certain special cases of the problem, or in the case of an optimization problem, one may give up the quest for optimality and try to develop an efficient algorithm that gives a nearly optimal solution [29, p 50]. An algorithm which gives the exact value for a specific optimization problem is called an **exact method**, while a **heuristic method** provides a solution that is hopefully close to optimal, within a reasonable amount of computational time for all problem instances, but which does not guarantee optimality or feasibility [42, p 226]. A simple heuristic method that is often used is a **greedy algorithm**, which is an unsophisticated strategy of progressively building up a solution by making the best possible choice at each iteration, regardless of the subsequent effect of that choice [29, p 54]. In the case of a greedy graph algorithm, the algorithm may result in an optimal solution for certain types of graph structure classes, or provide good solutions for other graph structure classes, but may also fail for a third group of graph structure classes [112, p 496]. Finally, heuristic methods may also be used to determine an upper or lower bound on an optimal solution to a specific **NP-complete** optimization problem.

A **metaheuristic method** is a high-level general strategy which guides other heuristics to search for feasible solutions [11]. In this regard, metaheuristic methods modify other heuristic methods to avoid getting trapped in local optima or repeatedly visiting the same set of solutions (called **cycling**), to produce solutions beyond those normally obtained by straightforward heuristic methods, and to provide reasonable assurance that promising regions of the **search space** (space of candidate solutions) were not overlooked [95, p 284].

Starting from an initial solution (which may be obtained via some other technique or may be generated randomly), a **local search technique** iteratively steps from one candidate solution to a neighbouring solution in an attempt at improving it. The modification that transforms a solution into one of its neighbours is called a **move**. The selection of the move to be performed at each step of the search is based on the **cost function**, which assesses the quality of the solution. The search continues to move from solution to solution in the search space until a solution is deemed optimal or a time bound has elapsed [101]. By allowing changes, as described above, to improve a candidate solution, local search techniques eventually often perform better than greedy algorithms [112, p 497].

2.3 Basic Concepts in Graph Colouring

The notion of graph colouring was informally introduced in §1.2, but may now be defined in formal graph theoretic terms. A **colouring** of a graph G is an assignment of colours to the vertices of G , one colour to each vertex, according to a certain rule \mathcal{R} [53, p 326]. Necessarily, every colouring of G utilising x colours, partitions the vertex set $V(G)$ into x sets, called **colour classes**, such that each vertex of a particular colour class receives the same colour [112, p 191]. If x colours are used for a colouring of G , the colouring is referred to as an x -**colouring** of G . In the *classical* graph colouring problem the rule \mathcal{R} is that the colour classes should be independent sets and this type of colouring is referred to as a **proper colouring**, that is, a proper colouring of G is an assignment of colours to the vertices in such a manner

²At each iteration of an algorithm implementing an interval halving scheme, an interval $[\ell, r)$ on the real line is halved, starting, in this case, with the interval $[1, p(G) + 1)$, until the answer $\omega(G)$ is found at ℓ .

that no two adjacent vertices of G are assigned the same colour [109, p 85]. If it is possible to find a proper colouring of a graph G , using x colours, G is said to be x -**colourable**. A graph is x -colourable if and only if the graph is x -partite. In particular, a 2-colourable graph is the same thing as a bipartite graph [31, p 193]. The (classical) **chromatic number**, $\chi(G)$, of a graph G is the smallest integer x for which G is x -colourable [112, p 191]. Equivalently, $\chi(G)$ is the smallest integer x for which G is an x -partite graph. A graph G with chromatic number x is referred to as x -**chromatic** and G is said to be $\chi(G)$ -**colourable**; such an optimal colouring is called an $\chi(G)$ -**colouring** of G [29, p 286]. As an example two colourings of the well-known Grötzsch graph are given in Figure 2.10.

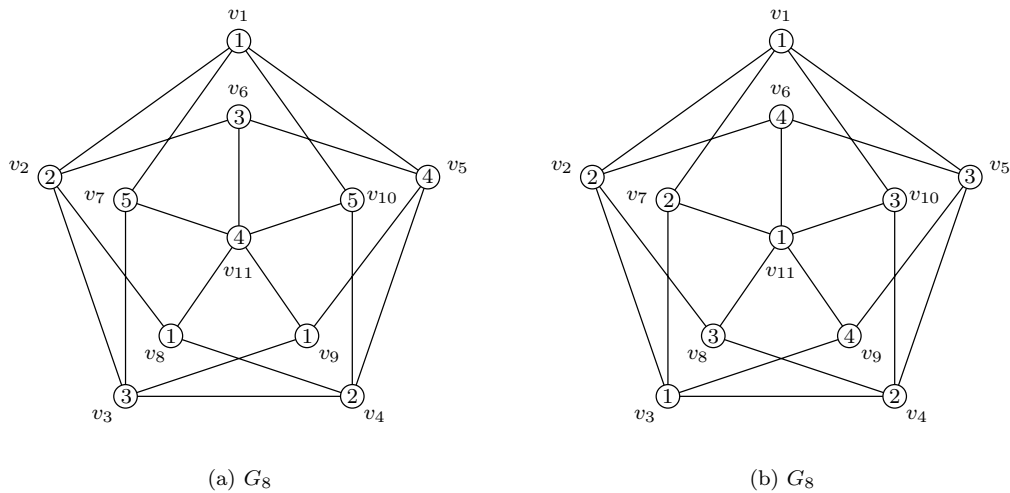


FIGURE 2.10: (a) A proper 5-colouring of the Grötzsch graph where vertices v_1, v_8, v_9 are coloured with colour 1, v_2, v_4 with colour 2, v_3, v_6 with colour 3, v_5, v_{11} with colour 4 and v_7, v_{10} with colour 5. However, the Grötzsch graph is 4-chromatic and a proper 4-colouring of this graph with colour classes $\mathcal{C}_1 = \{v_1, v_3, v_{11}\}$, $\mathcal{C}_2 = \{v_2, v_4, v_7\}$, $\mathcal{C}_3 = \{v_5, v_8, v_{10}\}$ and $\mathcal{C}_4 = \{v_6, v_9\}$ is given in (b).

Clearly, only edgeless graphs may be properly coloured with one colour only. For a proper colouring, any component H of a graph G with $q(H) \geq 1$ cannot be coloured using only one colour, thus $\chi(G) \geq 2$ if G contains an edge [9, p 146].

Two-colourability of a graph G may be tested by computing the distances from an arbitrary vertex $v \in V(G)$ (for each component if $k(G) > 1$). Let $\mathcal{Y} = \{u \in V(G) : d_G(u, v) \text{ is even}\}$ and let $\mathcal{Z} = \{w \in V(G) : d_G(w, v) \text{ is odd}\}$. Then the graph G is bipartite and thus 2-colourable, if \mathcal{Y} and \mathcal{Z} are independent sets [112, p 192]. This strategy may easily be converted to a polynomial time algorithm as illustrated in [103]. Therefore, given a graph G , the decision problem “Is G 2-colourable?” is in the class **P**. However, in general, it is not easy to determine the chromatic number of a graph. In fact, Karp [74] proved the following theorem on the decision problem “Given a graph G and an integer x , with $2 < x \leq p(G)$, is $\chi(G) \leq x$?” called the **chromatic number problem (CN)**.

Theorem 2.6 *CN is NP-complete.* ■

The proof of Theorem 2.6 was performed by first noticing that a proper x -colouring of a graph G is a certificate to the problem instance at hand. Testing the correctness of this x -colouring requires counting the number of different colours used on the vertices, which can be accomplished in linear time, and testing for each pair of vertices whether the two vertices have received the same colour and whether they are adjacent, which can be performed in quadratic time. Therefore CN is in the class **NP** [53, p 555]. Secondly, proving the **NP-completeness** of CN was achieved by polynomial time reducing an instance of 3-SAT to an instance of CN [29, p 287].

If CN is restricted to only three colours, the resulting decision problem “Given a graph G , is $\chi(G) \leq 3$?” (**3CN**) is still **NP-complete**. A proof of this result may be found in [112, p 500–501]. Even if 3CN is

further restricted to planar graphs, the decision problem is still **NP-complete**. A proof of this result may be found in [40, p 221–223].

Since no efficient algorithm to determine the chromatic number of a general graph is known, the establishment of analytic bounds on the chromatic number is desirable. First of all, an upper bound on the number of colours required for a proper colouring of the large class of planar graphs was already discussed informally in Chapter 1. This bound, namely The Four-colour Theorem, is perhaps one of the most famous results in graph theory, and may now be formulated in terms of the chromatic number.

Theorem 2.7 (Four-colour Theorem) *For every planar graph G , $\chi(G) \leq 4$.* ■

Furthermore, it follows by Theorem 2.3 that a graph is 2-colourable if and only if it has no odd cycles; therefore, if a graph G contains an odd cycle in any of its components, then $\chi(G) \geq 3$. Also, for the complete graph K_n , it holds that $\chi(K_n) = n$. Therefore, if a graph G contains K_n as a subgraph, then $\chi(G) \geq n$ [42, p 133]. The latter is included in the lower bound on the chromatic number of a graph G given in Theorem 2.8 below, together with various other upper and lower bounds on the chromatic number of a graph G . The proofs of the two lower bounds, as well as that of the first upper bound, are standard, and may be found in [112, p 194]. The second upper bound, due to Szekeres and Wilf [102], is also becoming well-known and a proof of this bound may be found in [9, p 148]. A proof of the third upper bound, due to Welsh and Powell [111], may be found in [31, p 207].

Theorem 2.8 (Chromatic Bounds) *Let G be a graph of order n . Let the vertices v_i , $1 \leq i \leq n$, of G be arranged in non-increasing order of degree and let $\deg_G(v_i) = \ell_i$. Then*

$$\max \left\{ \omega(G), \frac{n}{\beta(G)} \right\} \leq \chi(G) \leq \min \left\{ 1 + \Delta(G), 1 + \max\{\delta(H) : H \subset G\}, \max_{1 \leq i \leq n} \min\{i, \ell_i + 1\} \right\},$$

where $\omega(G)$, $\beta(G)$ and $\Delta(G)$ denote respectively the clique number, the independence number and the maximum degree of G . ■

Clearly, the upper bound $\chi(G) \leq 1 + \Delta(G)$ holds with equality for complete graphs and odd cycles [109, p 89]. Brooks [18] proved that these are the only connected graphs that require more than $\Delta(G)$ colours. This result is given, without loss of generality, for connected graphs only in Theorem 2.9. It may be extended to all graphs, since the chromatic number of a graph is the maximum chromatic number of its components [112, p 193]. There exist various proofs of Theorem 2.9. For example, a proof using induction on the order of the graph and then considering the cases of regular graphs and graphs that are not regular, may be found in [31, p 194–197]. Other proofs may also be found in [9, p 148], [53, p 334] and [112, p 197]. Although the proof preferred by the author is also standard, the proof is repeated here, because it is constructive in the sense that it suggests an efficient algorithm for finding a proper colouring for a graph G that is neither complete nor an odd cycle, using at most $\Delta(G)$ colours.

Theorem 2.9 (Brooks' Theorem) *Let G be a connected graph with maximum degree $\Delta(G)$. Then $\chi(G) \leq \Delta(G)$ if and only if G is neither a complete graph nor an odd cycle.*

Proof: If $\Delta(G) \leq 1$, then G is a complete graph. If $\Delta(G) = 2$, then G must be a cycle or a path. If G is an odd cycle, then $\chi(G) = 3$, otherwise if G is an even cycle or a path, $\chi(G) \leq 2$ [42, p 133]. Hence, the theorem holds for a connected graph G with maximum degree $\Delta(G) \leq 2$. Thus, assume that G is not a complete graph and $\Delta(G) \geq 3$. Let v be a vertex such that $\deg_G(v) = \Delta(G)$. Since G is connected and not complete, there exist two nonadjacent vertices u and w such that uv and vw are edges of G .

Let H_1 be the component of $G - \{u, w\}$ that contains v . Arrange the vertices of H_1 in non-increasing order of their distances from v in H_1 . Suppose this gives rise to the sequence $v_3, v_4, \dots, v_n (= v)$. Let $v_1 = u$ and $v_2 = w$. Observe that for $1 \leq i < n$, v_i is adjacent to some v_j where $j > i$. Assign colour 1 to v_1 and v_2 ; then successively colour v_3, v_4, \dots, v_n each with the first available colour in the list colour 1, colour 2, \dots , colour Δ . By the construction of the sequence v_1, v_2, \dots, v_n , each vertex v_i ($1 \leq i \leq n - 1$) is adjacent to some vertex v_j with $j > i$, and therefore is adjacent to at most $\Delta - 1$ vertices v_j with $j < i$. So, when its turn comes to be coloured, v_i is adjacent to at most $\Delta - 1$ already coloured vertices.

Therefore, one of the colours colour 1, colour 2, \dots , colour Δ will be available to colour v_i . Finally, since v_n is adjacent to two vertices of colour 1, namely v_1 and v_2 , it is adjacent to at most $\Delta - 1$ vertices that have received distinct colours and can be assigned one of the colours colour 2, colour 3, \dots , colour Δ to produce a proper x -colouring of $\langle V(H_1) \cup \{u, w\} \rangle$ with $x \leq \Delta(G)$ [52, p 228].

If vertices of G remain that have not yet been coloured, then $G - \{u, w\}$ is disconnected. Let H_2 be the union of the components of $G - \{u, w\}$ that do not contain v . Then $H_3 = \langle V(H_2) \cup \{u, v, w\} \rangle$ is connected. Order the vertices of H_2 in non-increasing order of their distances from v in H_3 . Let w_1, w_2, \dots, w_m be the resulting sequence and let $w_{m+1} = u$ and $w_{m+2} = w$. For each w_i ($1 \leq i \leq m$) there is a w_j with $j > i$ so that w_i is adjacent to w_j . As before, a y -colouring of H_2 with $y \leq \Delta$ may be described by successively colouring w_1, w_2, \dots, w_m , each with the first available colour in the list colour 1, colour 2, \dots , colour Δ .

In order to obtain a colouring of G that uses at most Δ colours, some vertices of G may have to be recoloured. Suppose that some colour i is present in the neighbourhood of u in $\langle V(H_2) \cup \{u, w\} \rangle$ as well as in the neighbourhood of w in $\langle V(H_2) \cup \{u, w\} \rangle$. Let colour j be assigned to v . If colour $i \neq$ colour j , then interchange the colours of the vertices that have been coloured with colour i and colour j in H_1 . Now both u and w are adjacent with at most $\Delta - 1$ colours. So u and w can each be recoloured with a colour that does not yet appear in its neighbourhood in G to produce a proper colouring of G that uses at most Δ colours.

Suppose now that there is no colour that is represented in the neighbourhood of both u and w in $\langle V(H_2) \cup \{u, w\} \rangle$. Let C_1 and C_2 be the sets of colours assigned to the vertices of H_2 that are adjacent to u and w , respectively. For this case, it follows that $C_1 \cap C_2 = \emptyset$. If colour 1 $\notin C_1 \cup C_2$, then u and w may retain colour 1 and a proper colouring of G that uses at most Δ colours is obtained. Thus, assume, without loss of generality, that colour 1 $\in C_1$. Let D_1 and D_2 be the sets of colours assigned to the vertices of H_1 that are adjacent to u and w , respectively. From the x -colouring of H_1 it follows that colour 1 $\notin D_1$. If $|C_1 \cup D_1| < \Delta$, retain colour 1 for w and recolour u with one of the colours in $\{\text{colour 1, colour 2, } \dots, \text{colour } \Delta\}$ to obtain a proper colouring of G that uses at most Δ colours.

Assume, therefore, that $|C_1 \cup D_1| = \Delta$. If $|C_1 \cup C_2| < \Delta$, let colour $i \in \{\text{colour 1, colour 2, } \dots, \text{colour } \Delta\} - (C_1 \cup C_2)$. Then colour $i \in D_1$. Interchange the colours of the vertices coloured with colour 1 and colour i in $\langle V(H_1) \cup \{u, w\} \rangle$ to obtain a colouring of G that uses at most Δ colours. Assume, thus, that $|C_1 \cup C_2| = \Delta$. Since $\deg_G(u) = \Delta$ in this case, $|C_2| = |D_1|$. If $|C_1| > 1$, let colour $i \in C_1 - \{\text{colour 1}\}$. Interchange the colours of the vertices coloured with colour j and colour i in H_1 . Retain colour 1 for w and recolour u with one of the colours in $\{\text{colour 1, colour 2, } \dots, \text{colour } \Delta\}$ not represented in its neighbourhood. Finally, suppose that $|C_1| = 1$. Then $|C_2| = |D_1| = \Delta - 1$. So v is the only neighbour of w in H_1 . Let colour $i \in D_1 - \{\text{colour } j\}$. It follows that colour $i \neq$ colour 1. Interchange the colours of the vertices coloured with colour 1 and colour i in H_1 . Then v retains colour j (\neq colour 1). Since at least two vertices adjacent to u are now coloured with colour 1, it follows that u may be coloured with one of the colours in $\{\text{colour 2, colour 3, } \dots, \text{colour } \Delta\}$ to produce a proper colouring of G that uses at most Δ colours [29, p 290]. ■

Although the upper bound for $\chi(G)$ given in Brooks' Theorem is exact for the class of graphs $K_n \times K_2$, $n \geq 2$, there exist graphs for which the difference $\Delta(G) - \chi(G)$ can be made arbitrarily large. For example, $\chi(K_{2 \times n}) = 2$, but $\Delta(K_{2 \times n}) = n$ [29, p 290]. Similarly, for each of the bounds in Theorem 2.8 there is a graph whose chromatic number differs significantly from the bound. For example, the lower bound of $\omega(G)$ is exact for complete graphs and even cycles, but there are also graphs for which $\chi(G) - \omega(G)$ is arbitrarily large: if G is a triangle-free graph, then $\omega(G) \leq 2$. However, Mycielski [87] established, by construction, the existence of an x -chromatic, triangle-free graph for every positive integer x [31, p 209]. The Grötzsch graph, shown in Figure 2.10, is an example of a 4-chromatic triangle-free graph (with clique number $\omega = 2$).

Since there exist graphs for which the bounds on the chromatic number in Theorem 2.8 are poor, and in view of the fact that CN is **NP-complete**, a practical consideration is that of developing efficient heuristic methods for colouring a graph which give approximations to minimum colourings. The most elementary heuristic, the **sequential colouring algorithm**, is a greedy algorithm which labels the vertices of the graph to be coloured in a *random* order and then assigns, in sequence, the first available colour that has not yet been assigned to any of its neighbours to each vertex [53, p 333]. For a graph of

order p and size q , the sequential colouring algorithm has time complexity $\mathcal{O}(pq)$ [29, p 293] and is given in pseudo-code as Algorithm 2 for later reference.

Algorithm 2 Sequential colouring algorithm

Input: A graph G of order n .

Output: A proper colouring of G , using at most $\Delta(G) + 1$ colours.

- 1: List the vertices of G as v_1, v_2, \dots, v_n
 - 2: List the colours available as $1, 2, \dots, n$
 - 3: **for all** $i = 1, \dots, n$ **do**
 - 4: Assign j to v_i with j the smallest number not yet assigned to any of the neighbours of v_i
 - 5: **end for**
 - 6: **return** the resulting colouring of G .
-

However, the number of colours used in this algorithm depends in a sensitive manner on the sequence in which the vertices are labelled — especially in large graphs, as illustrated in [109, p 86], but will always use at most $\Delta + 1$ colours [9, p 147]. As an illustration of the sequential colouring algorithm, suppose the vertices of the Grötzsch graph are labelled as in Figure 2.10. A proper colouring in four colours, using the sequential colouring algorithm on this labeling of the vertices, is given in Figure 2.11(a).

Generally speaking, if a vertex has large degree and many of its neighbours have already been coloured, then it is more likely that the vertex has to be coloured with a previously unused colour. Therefore, the **Welsh and Powell algorithm** [111], also known as the **largest-first algorithm**, labels the vertices in non-increasing order of their degrees. The algorithm then proceeds with the colouring process in exactly the same manner as the sequential colouring algorithm (Steps 2–6 of Algorithm 2). The Welsh and Powell algorithm may be found in [31, p 202] and the number of colours used in the algorithm is related to Welsh and Powell's upper bound (the third upper bound in Theorem 2.8). This algorithm too, does not always give a minimum colouring, since different sequences of non-increasing degree may still exist (for example, in regular graphs) but will also always use at most $\Delta + 1$ colours [31, p 202]. In the Grötzsch graph in Figure 2.10, vertex v_{11} is the only vertex with degree 5 and will thus be coloured first by the Welsh and Powell algorithm. Next, the vertices v_1, \dots, v_5 , all of degree 4, may be coloured in any sequence. Finally, the vertices v_6, \dots, v_{10} may be coloured in any sequence, since they all have degree 3. A proper colouring of the Grötzsch graph in four colours, using the largest-first algorithm and the vertex sequence $v_{11}, v_1, \dots, v_{10}$, is given in Figure 2.11(b).

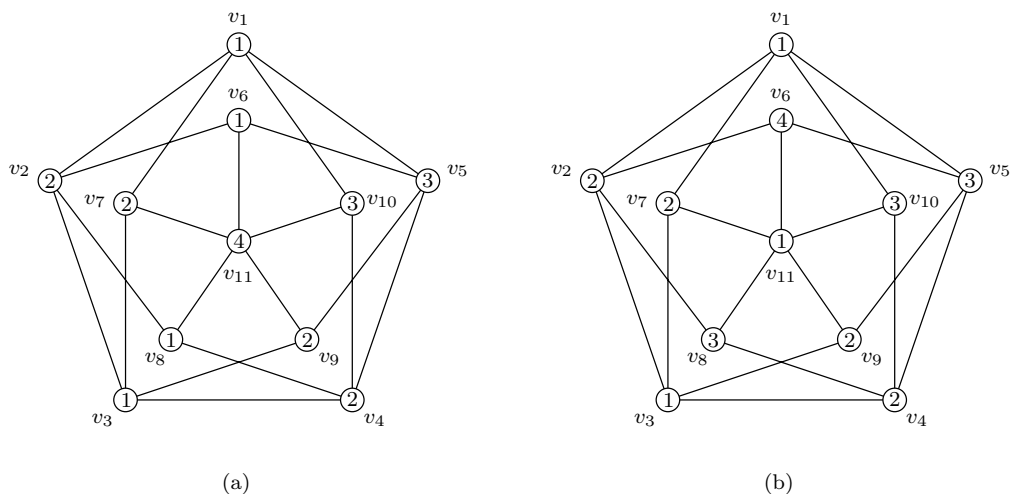


FIGURE 2.11: (a) A proper 4-colouring of the Grötzsch graph as obtained via the sequential colouring algorithm on the vertex labelling given in the graph. (b) A proper 4-colouring of the Grötzsch graph as obtained via the largest-first colouring algorithm using the vertex sequence $v_{11}, v_1, \dots, v_{10}$ indicating in which order the vertices are coloured.

The largest–first algorithm may further be refined based on the intuition that if two vertices have equal degree, then the one having the more densely coloured neighbourhood will be harder to colour later. Let the **colour degree** of a vertex v of a graph G be the number of different colours already assigned to vertices in $N_G(u)$. Then an adapted version of the largest–first algorithm, called **Brelaz’s heuristic**³, is obtained if, among the uncoloured vertices with maximum colour degree, the vertex with largest degree in the uncoloured subgraph is chosen to be coloured next. Therefore, in Brelaz’s heuristic the order in which the vertices should be coloured is not determined beforehand, but is determined as the algorithm proceeds [52, p 232]. Applying Brelaz’s heuristic to the Grötzsch graph in Figure 2.10, vertex v_{11} is still coloured first as in the largest–first algorithm. The vertices v_6, \dots, v_{10} all have colour degree 1 and a degree of 2 in the uncoloured subgraph, $G_8 - v_{11}$, given in Figure 2.12(a). Any one of these vertices may thus be coloured next. Choose v_6 to be coloured next. Then $v_2, v_5, v_7, \dots, v_{10}$ all have maximum colour degree 1, and v_2 and v_5 have a degree of 3 in the uncoloured subgraph, $G_8 - \{v_6, v_{11}\}$, given in Figure 2.12(b), while v_7, \dots, v_{10} have a degree of 2. From v_2 and v_5 , choose v_2 to be coloured next. At this point v_8 is the only uncoloured vertex with maximum colour degree and is coloured next. All uncoloured vertices now have the same colour degree of 1. In the uncoloured subgraph, $G_8 - \{v_2, v_6, v_8, v_{11}\}$ given in

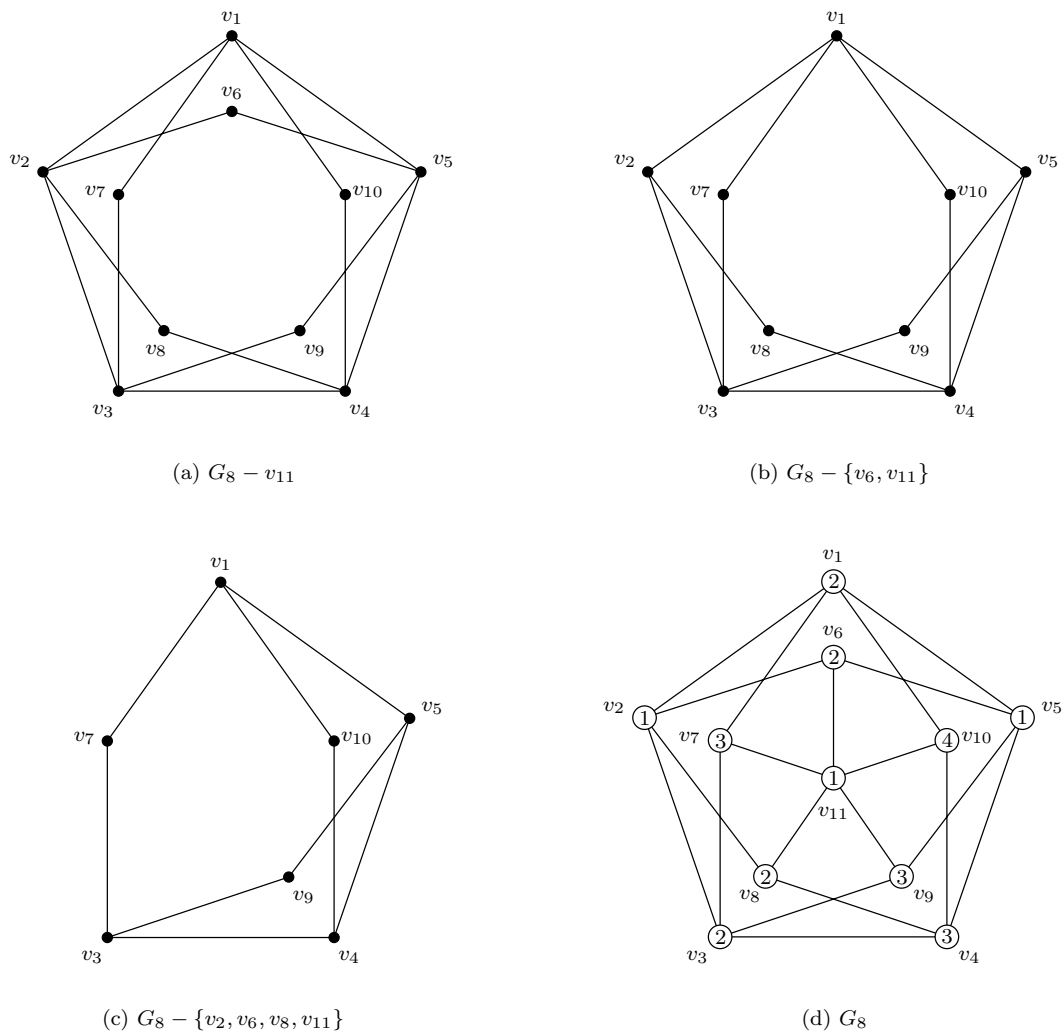


FIGURE 2.12: Uncoloured subgraphs of the graph G_8 in Figure 2.10 during the course of Brelaz’s heuristic, (a) after vertex v_{11} has been coloured, (b) after vertices v_6 and v_{11} have been coloured, and (c) after vertices v_2, v_6, v_8 and v_{11} have been coloured. (d) A proper 4–colouring of the Grötzsch graph as obtained via Brelaz’s heuristic.

³In Brelaz’s original paper [13] this algorithm was called DSATUR, but has become popularly known as Brelaz’s heuristic.

Figure 2.12(c), vertices v_1, v_3, v_4 and v_5 all have a degree of 3. Choose v_1 to be coloured next. Continuing in this manner v_5, v_4, v_{10} and v_3 are coloured in this sequence. Finally, v_7 and v_9 may be coloured in any order. This leads to the proper colouring of the Grötzsch graph in four colours as given in Figure 2.12(d).

A reverse strategy of the same basic idea that a vertex with large degree is more likely to force one to use a previously unused colour, gives rise to the **smallest–last algorithm**. In the smallest–last algorithm, due to Matula *et al.* [80], the ordering of the vertices is determined in a reverse order by repeatedly selecting one of the vertices with minimum degree and deleting the selected vertex from the graph. The vertex with minimum degree is thus selected first and deleted from the graph. This vertex is placed in the last position of the order in which the vertices should be coloured. In the remaining subgraph, one again selects the vertex with minimum degree, deletes it from the graph and places it in the second last position of the order in which the vertices should be coloured. This process is continued until all vertices have been positioned in the colouring order [31, p 202]. Thereafter, the colouring proceeds as in Steps 2-6 of Algorithm 2. In the Grötzsch graph, G_8 , in Figure 2.10, vertices v_6, \dots, v_{10} all have a minimum degree of 3 and any one may be selected. Using the convention of choosing the smallest subscripted vertex in case of a tie, vertex v_6 is selected to be coloured last. In the graph $G_8 - v_6$ all the vertices $v_2, v_5, v_7, \dots, v_{10}$ now have a minimum degree of 3. Choose v_2 next. After deleting v_2 from $G_8 - v_6$, the resulting graph has only one vertex of degree 2, namely v_8 , and v_8 should thus be coloured just before v_2 . For clarity, the current subgraph, $G_8 - \{v_2, v_6, v_8\}$ is given in Figure 2.13(a). As may be seen in Figure 2.13(a), the graph $G_8 - \{v_2, v_6, v_8\}$ is 3-regular. Thus, choose v_1 to be inserted in the reverse colouring order next. In $G_8 - \{v_1, v_2, v_6, v_8\}$ vertices v_7 and v_{10} both have minimum degree equal to 1. Thus, choose v_7 next. Continuing in the above manner, v_{10} and then v_{11} must be inserted in the reverse colouring order. Finally, in $G_8 - \{v_1, v_2, v_6, v_7, v_8, v_{10}, v_{11}\}$ all remaining vertices have minimum degree equal to 2. Thus v_3 is chosen next, resulting in the final selection of v_4, v_5 and v_9 in that order. Therefore the colouring sequence for the smallest–last algorithm is $v_9, v_5, v_4, v_3, v_{11}, v_{10}, v_7, v_1, v_8, v_2, v_6$ and the resulting proper colouring of the Grötzsch graph in four colours using the smallest–last algorithm is given in Figure 2.13(b).

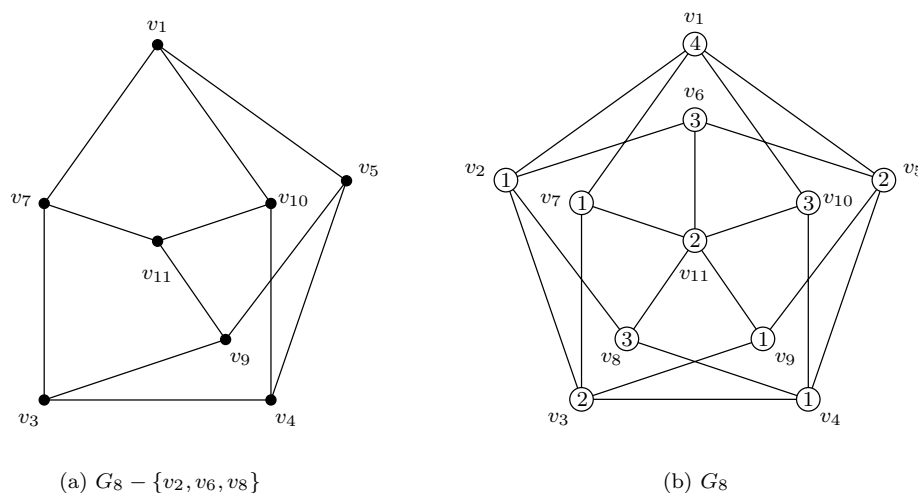


FIGURE 2.13: (a) The resulting subgraph of G_8 after 3 steps of determining the order in which the vertices should be coloured during the smallest–last colouring algorithm. (b) A proper 4-colouring of the Grötzsch graph using the vertex sequence $v_9, v_5, v_4, v_3, v_{11}, v_{10}, v_7, v_1, v_8, v_2, v_6$ as obtained via the smallest–last colouring algorithm.

The smallest–last algorithm generally achieves slightly better results than the largest–first algorithm, as demonstrated by Brezaz [13], while, in general, Brezaz’s heuristic out–performs both the largest–first and smallest–last algorithms [52, pp 231, 233]. Furthermore, all four of the above heuristics use a technique called **successive augmentation**, where the vertices are coloured in sequence, one at a time, without any attempt to improve the colouring by means of perturbation, until all vertices have been coloured and the final colouring is thus obtained [71]. However, any of the above algorithms may be improved

(as Brelaz's [13] testing showed) by incorporating a colour interchange approach. In such an approach all vertices from a specific subset of the vertex set of the graph that were previously coloured with colour i , are recoloured with colour j and all vertices in the subset previously coloured with colour j are recoloured with colour i in an $i - j$ **colour interchange**. The original algorithm would then be adapted to attempt performing a number of colour interchanges before using additional colours [52, p 233]. One more algorithm will be given here — the one suggested by the proof of Brooks' Theorem and which incorporates a colour interchange as described above. This algorithm, which is called **Brooks' algorithm** for the purposes of this dissertation, gives a proper colouring of the vertices of a graph G that is neither complete nor an odd cycle, using at most $\Delta(G)$ colours, and is given in pseudo-code as Algorithm 3.

The working of Brooks' algorithm is illustrated with the aid of the graph G_9 given in Figure 2.14. It is clear from Figure 2.14 that $\Delta(G_9) = 7$. In fact, there are eleven vertices with degree 7 and any one of these vertices may be chosen in Step 1 of Brooks' algorithm. The vertex v that was chosen in Step 1 and the subsequent choices of u and w in Step 2 are already indicated in Figure 2.14.

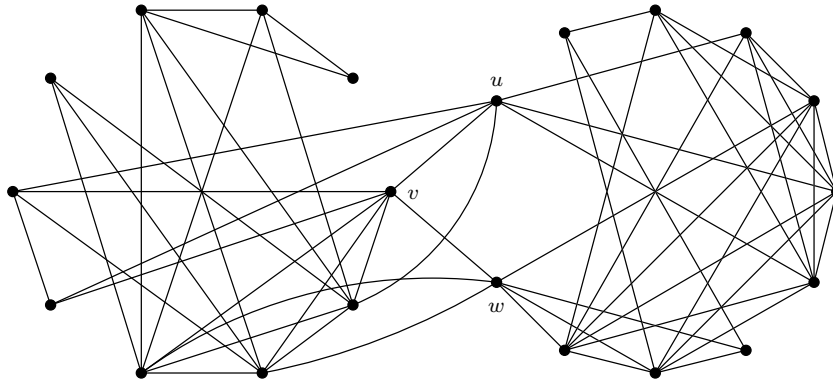


FIGURE 2.14: The graph G_9 used to illustrate the working of Brooks' algorithm. The vertices u, v and w that are chosen in Steps 1 and 2 of the algorithm are already indicated in the figure.

Brooks' algorithm proceeds with the colouring of $\langle V(H_1) \cup \{u, w\} \rangle$ as dictated by Steps 5–11, where H_1 is the component of $G_9 - \{u, w\}$ that contains v . The resulting proper colouring of $\langle V(H_1) \cup \{u, w\} \rangle$ as well as the listing of the vertices of $\langle V(H_1) \cup \{u, w\} \rangle$ performed in Steps 5 and 6 of the algorithm, are given in Figure 2.15(a). Since some of the vertices of G_9 have not yet been coloured, one has to proceed from Step 13 of the algorithm onwards. Let H_2 be the union of the components of $G - \{u, w\}$ that do not contain v as before. The proper colouring of H_2 as was performed in Steps 13–17 of Brooks' algorithm as well as the listing of the vertices of H_2 performed in Step 14, are given in Figure 2.15(b).

In Steps 18 and 19 $C_1 = \{1, 4, 5\}$ and $C_2 = \{2, 3, 6, 7\}$ respectively, so that $C_1 \cap C_2 = \emptyset$. Thus, Brooks' algorithm continues at Step 27, where $D_1 = \{2, 3, 6, 7\}$ and $D_2 = \{4, 5, 7\}$ in Steps 27 and 28, respectively. Since $1 \in C_1$, $|C_1 \cup D_1| = \Delta$, $|C_1 \cup C_2| = \Delta$ and $|C_1| > 1$ the colour interchange in Steps 38 and 39 of the algorithm has to be performed. In Step 38 of Brooks' algorithm $C_1 - \{1\} = \{4, 5\}$ and $i = 4$ is chosen. Therefore, in Step 39 of the algorithm vertex v_9 is recoloured with colour 7, and vertex $v_{12} = v$ with colour 4, respectively. The algorithm concludes by recolouring vertex u with colour 7 in Step 44, to obtain a proper colouring of the original graph G_9 . This proper colouring of G_9 is given in Figure 2.16.

To conclude the discussion on heuristic colouring algorithms, the existence of algorithms that attempt to colour independent sets, where the cardinality of each independent set is an expected value determined beforehand, are mentioned. According to Gould [52, p 234], Johri and Matula [73] have produced a variety of algorithms based on this approach. As opposed to algorithms using successive augmentation, there also exist local optimization colouring algorithms. For example, Hertz and De Werra [64] used the tabu search methodology to colour a graph and Johnson *et al.* [71] applied the technique of simulated annealing to graph colouring. Unfortunately, none of the above algorithms produces for *every* graph a colouring that is close to its chromatic number [29, p 294].

Although heuristic colouring methods are often capable of producing a $\chi(G)$ -colouring of a graph G

Algorithm 3 Brooks' algorithm**Input:** A graph G that is neither a complete graph nor an odd cycle.**Output:** A proper colouring of G in at most $\Delta(G)$ colours.

```

1: Choose  $v$  such that  $\deg_G(v) = \Delta(G)$ 
2: Choose  $u$  and  $w$  such that  $uv, vw \in E(G)$  and  $uw \notin E(G)$ 
3:  $H_1 \leftarrow$  component of  $G - \{u, w\}$  such that  $v \in V(H_1)$ 
4:  $H_2 \leftarrow$  union of components of  $G - \{u, w\}$  such that  $v \notin V(H_2)$ 
5: List the vertices of  $H_1$  as  $v_3, v_4, \dots, v_n (= v)$  such that  $d_{H_1}(v_3, v) \geq d_{H_1}(v_4, v) \geq \dots \geq d_{H_1}(v_n, v) = 0$ 
6:  $v_1 \leftarrow u, v_2 \leftarrow w$ 
7: List the colours available as  $1, 2, \dots, \Delta$ 
8: Assign 1 to  $v_1$  and  $v_2$ 
9: for all  $i = 3, \dots, n$  do
10:   Assign  $j$  to  $v_i$  with  $j$  the smallest number not yet used in  $N_{\langle V(H_1) \cup \{u, w\} \rangle}(v_i)$ 
11: end for
12: if  $V(H_2) \neq \emptyset$  then
13:    $H_3 \leftarrow \langle V(H_2) \cup \{u, v, w\} \rangle; H_4 \leftarrow \langle V(H_2) \cup \{u, w\} \rangle$ 
14:   List the vertices of  $H_2$  as  $u_1, u_2, \dots, u_m$  such that  $d_{H_3}(u_1, v) \geq d_{H_3}(u_2, v) \geq \dots \geq d_{H_3}(u_m, v)$ 
15:   for all  $i = 1, \dots, m$  do
16:     Assign  $j$  to  $u_i$  with  $j$  the smallest number not yet used in  $N_{H_2}(u_i)$ 
17:   end for
18:    $C_1 \leftarrow$  set of colours assigned to vertices in  $N_{H_4}(u)$ 
19:    $C_2 \leftarrow$  set of colours assigned to vertices in  $N_{H_4}(w)$ 
20:   if  $C_1 \cap C_2 \neq \emptyset$  then
21:      $i \leftarrow$  a colour in  $C_1 \cap C_2$ 
22:     if  $i \neq j$  then
23:       interchange  $i$  and  $j$  in  $H_1$ 
24:     end if
25:     recolour  $u, w$  with a colour not assigned to vertices in  $N_G(u), N_G(w)$  respectively
26:   else
27:      $D_1 \leftarrow$  set of colours assigned to vertices in  $N_{H_1}(u)$ 
28:      $D_2 \leftarrow$  set of colours assigned to vertices in  $N_{H_1}(w)$ 
29:     if  $1 \in C_1$  ( $1 \in C_2$ ) then
30:       if  $|C_1 \cup D_1| < \Delta$  ( $|C_2 \cup D_2| < \Delta$ ) then
31:         recolour  $u$  with a colour not in  $C_1 \cup D_1$  ( $C_2 \cup D_2$ )
32:       else  $[|C_1 \cup D_1| = \Delta$  ( $|C_2 \cup D_2| = \Delta$ )]
33:         if  $|C_1 \cup C_2| < \Delta$  then
34:            $i \leftarrow$  a colour in  $\{1, 2, \dots, \Delta\} - (C_1 \cup C_2)$ 
35:           interchange 1 and  $i$  in  $\langle V(H_1) \cup \{u, w\} \rangle$ 
36:         else  $[|C_1 \cup C_2| = \Delta]$ 
37:           if  $|C_1| > 1$  ( $|C_2| > 1$ ) then
38:              $i \leftarrow$  a colour in  $C_1 - \{1\}$  ( $C_2 - \{1\}$ )
39:             interchange  $i$  and  $j$  in  $H_1$ 
40:           else  $[|C_1| = 1$  ( $|C_2| = 1$ )]
41:              $i \leftarrow$  a colour in  $D_1 - \{j\}$  ( $D_2 - \{j\}$ )
42:             interchange 1 and  $i$  in  $H_1$ 
43:           end if
44:           recolour  $u$  with a colour not in  $N_G(u)$ 
45:         end if
46:       end if
47:     end if
48:   end if
49: end if
50: return the resulting colouring of  $G$ .

```

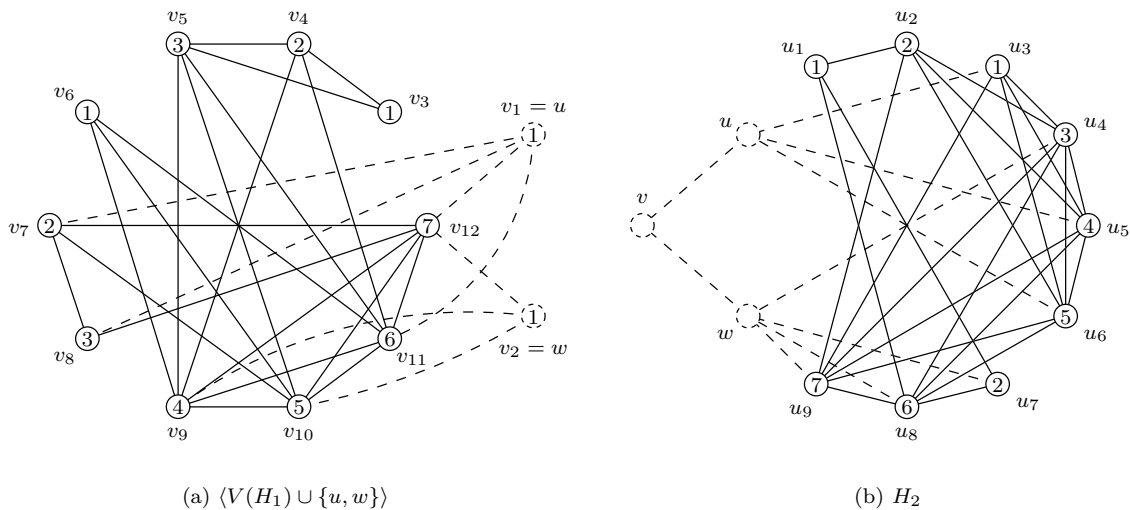


FIGURE 2.15: (a) The resulting proper colouring of the subgraph $\langle V(H_1) \cup \{u, w\} \rangle$ of G_9 after performing Steps 5–11 of Brooks’ algorithm. The solid lines and vertices form the graph H_1 determined in Step 3. (b) The resulting proper colouring of the subgraph H_2 of G_9 after Steps 13–17 of Brooks’ algorithm have been performed. The solid lines and vertices form the graph H_2 determined in Step 4.

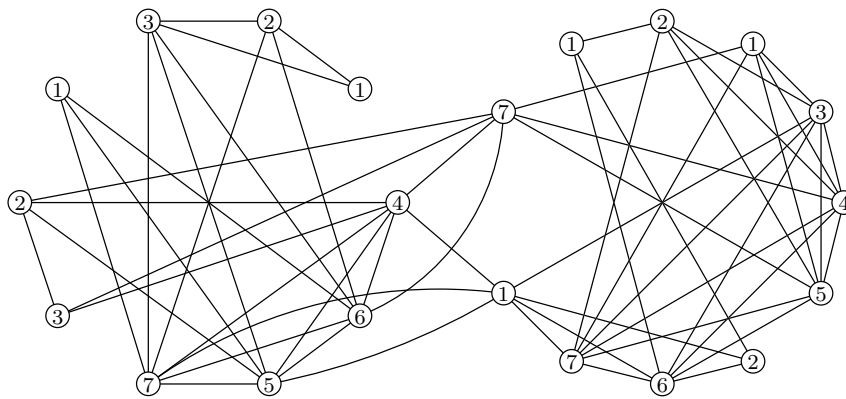


FIGURE 2.16: The proper colouring of the graph G_9 in Figure 2.14 obtained via Brooks’ algorithm.

easily, *proving* that no $(\chi(G) - 1)$ -colouring exists is a difficult task which dominates the complexity of any exact method [63]. In order to appreciate the computational complexity of computing the chromatic number of a graph G of order n exactly, it is worth pondering the computational expense of a natural (albeit naive) brute-force approach whereby every possible x -colouring of G is tested to determine whether it is a proper colouring of G . If no x -colouring of G is proper, then G is not x -colourable and the bound $\chi(G) > x$ has been established. Otherwise the bound $\chi(G) \leq x$ has been established. If this procedure is repeated for $x = 3, 4, 5, \dots$ until a proper x -colouring of G is obtained for the first time, then clearly $\chi(G) = x$ [53, p 332]. The worst-case time complexity of this naive algorithm may be estimated by noting that the number of partitions of the vertex set of G into x colour classes of sizes n_1, \dots, n_x is the well-known multinomial coefficient

$$\binom{n}{n_1, \dots, n_x} = \frac{n!}{n_1! \cdots n_x!}$$

where, of course, $n_1 + \dots + n_x = n$. Therefore the total number of partitions of the vertex set of G into

x colour classes is

$$\sum_{\substack{(n_1, \dots, n_x) : \\ n_1 + \dots + n_x = n}} \binom{n}{n_1, \dots, n_x} = \sum_{\substack{(n_1, \dots, n_x) : \\ n_1 + \dots + n_x = n}} \binom{n}{n_1, \dots, n_x} 1^{n_1} 1^{n_2} \dots 1^{n_x} = \underbrace{(1 + 1 + \dots + 1)}_{x \text{ terms}}^n = x^n$$

by the multinomial theorem [98, p 11]. It is concluded that the worst-case time complexity of the above naive exact colouring algorithm is $\mathcal{O}(\chi(G)^n)$, which is an exponential function of the order of G .

Of course, exact algorithms used in practice are not as naive in their approach as the exact procedure described above. One of the first exact graph colouring algorithms, due to Brown [21], uses his idea of avoiding redundancy. Applied to graph colouring problems, Brown's [21] general idea of redundant solutions to problems where a solution is obtained by partitioning a set of objects, states that it is useless to determine solutions that may be obtained from one another by simply interchanging some of the colours. For example, consider the problem of colouring a graph G with vertex set $V(G) = \{v_1, v_2, v_3\}$ using only two colours, ignoring the proper colouring rule for the time being. The three vertices may be coloured in eight different ways as listed below.

Vertices	Colourings							
v_1	Colour 1	Colour 1	Colour 1	Colour 1	Colour 2	Colour 2	Colour 2	Colour 2
v_2	Colour 1	Colour 1	Colour 2	Colour 2	Colour 1	Colour 1	Colour 2	Colour 2
v_3	Colour 1	Colour 2	Colour 1	Colour 2	Colour 1	Colour 2	Colour 1	Colour 2

The two colours, colour 1 and colour 2, may be seen as merely an assignment of labels to the vertices which are in the same colour classes and in themselves have no specific meaning. Therefore, the labellings colour 1 and colour 2 may be interchanged without obtaining a different solution. Thus, four of the colourings above are **redundant** because they may be obtained from one of the other four colourings by reassigning the vertices labelled colour 1 as colour 2 and the colour 2 vertices as colour 1, *i.e.* one out of each of the following solution pairs is redundant [21]:

Vertices	Colourings							
	Pair 1		Pair 2		Pair 3		Pair 4	
v_1	Colour 1	Colour 2	Colour 1	Colour 2	Colour 1	Colour 2	Colour 1	Colour 2
v_2	Colour 1	Colour 2	Colour 1	Colour 2	Colour 2	Colour 1	Colour 2	Colour 1
v_3	Colour 1	Colour 2	Colour 2	Colour 1	Colour 1	Colour 2	Colour 2	Colour 1

When a proper colouring of a graph G is sought, a **partial colouring**, p , of G is defined as an assignment of colours to a subset of vertices $S \subset V(G)$. Similarly, a **redundant partial colouring** is defined as any partial colouring that may be obtained from another partial colouring by interchanging two or more colours. During the search for a colouring of G in x colours, after a partial colouring p has been obtained, an uncoloured vertex v is chosen, and a maximum of x new partial colourings are generated from p by assigning each of the x colours to v . The following theorem by Brown [21], stated here in terms of graph colouring, is used to prevent the generation of redundant partial colourings.

Theorem 2.10 *Let \mathcal{A} be a set of colours. Let \mathcal{A}_u and \mathcal{A}_n be disjoint subsets of \mathcal{A} such that $\mathcal{A}_u \cup \mathcal{A}_n = \mathcal{A}$, where \mathcal{A}_u contains those colours in \mathcal{A} that are used in a partial colouring p and \mathcal{A}_n contains those colours in \mathcal{A} that are not used in p . Then, when new partial colourings are generated from p , redundant partial colourings are prevented by only using colours in $(\mathcal{A}_u \cup \{\alpha\}) \subseteq \mathcal{A}$ to form new partial colourings from p , where α is any colour from \mathcal{A}_n . ■*

Brown [21] uses the above ideas to construct an exact algorithm for colouring the vertices of a graph with a minimum number of colours that prevents the generation of a large number of unnecessary (redundant or non-optimal) colourings. This is achieved by what he called "screening techniques". Suppose v_k is the vertex chosen to generate new partial colourings from partial colouring p of the graph G of order n . Initially, v_k may be coloured with any of the colours in the set $\{\text{colour 1, colour 2, } \dots, \text{colour } n\}$, but then colours are removed from this list by the screening process. Let $\mathcal{X}(v_k)$ be the final set of colours that may be assigned to v_k after the screening process has been completed. In determining $\mathcal{X}(v_k)$, three

algorithm by Brown uses the fact included in the first lower bound in Theorem 2.8 that the vertices of a graph containing a clique (not necessarily a maximum clique) cannot be coloured with fewer colours than the order of the clique, and commences by colouring the vertices of an initial clique. After a colouring in x colours has been obtained, the algorithm also repeatedly attempts to find a colouring in $x - 1$ colours, $x - 2$ colours and so forth. However, as soon as one of the initial clique vertices needs to be recoloured after a backtracking in an attempt to obtain a colouring with fewer colours, the algorithm terminates with the previous proper colouring obtained as final colouring. Therefore, the tree traversal is more structured in the sense that unnecessary backtracking is limited [13].

To appreciate the possible improvement on the algorithm by Brown, the tree that might be constructed and traversed during the course of Brelaz's algorithm is given in Figure 2.18. Using the fact that the clique vertices cannot be coloured with fewer colours than the order of the clique, the root of the tree in Figure 2.17 may be replaced by the node $[v_4 \in C_1, v_5 \in C_2, v_6 \in C_3, v_7 \in C_4]$ in the tree construction during execution of Brelaz's algorithm. The tree constructed by Brelaz's algorithm, given in Figure 2.18, has three levels and seven leaves as opposed to the tree constructed by Brown's algorithm, given in Figure 2.17, which has six levels and eleven leaves.

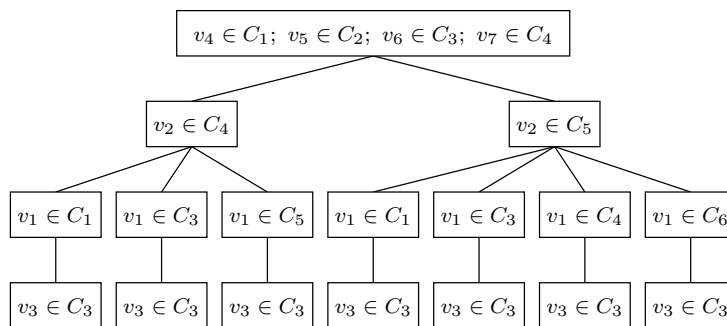


FIGURE 2.18: The tree that is implicitly constructed for the graph \overline{G}_1 in Figure 2.1(b) during the course of the exact colouring algorithm by Brelaz.

Unfortunately, Brelaz's improvement algorithm on the algorithm by Brown, contains two errors as pointed out by Peemöller [92]. Peemöller [92] presented two examples to illustrate where the algorithm by Brelaz fails, followed by a correct version of the modified version of Brown's algorithm. This algorithm, called **Brown's modified colouring algorithm** is given in pseudo-code as Algorithm 4.

Brown's modified colouring algorithm commences in Step 1 by testing whether or not the upper bound, x , obtained by some heuristic colouring algorithm, is equal to the order of a clique contained in the graph G for which a $\chi(G)$ -colouring is sought. If the two values are equal, then the optimum value of $\chi(G)$ has already been obtained and the algorithm terminates at Step 2. Otherwise, the search for an optimal colouring begins at Step 4 by determining an order in which the vertices should be coloured. Here the same ordering process as in Brown's [21] original algorithm may be used. The labelling in Steps 5 and 37 of Algorithm 4 is used to guide the backtracking during the (implicit) tree traversal. Initialization is performed in Step 6, where among others, the vertex counter, k , is set to $w + 1$, since v_{w+1} is the first vertex that needs to be coloured after the clique vertices have been coloured in Step 5. The final set of colours, $\mathcal{X}(v_k)$, that may be assigned to v_k after Brown's [21] screening process described above has been completed, is determined in Steps 9 and 10. If $\mathcal{X}(v_k)$ is not empty, then v_k is coloured in Steps 17 and 18, and the vertex counter is incremented in Step 18 so that the next vertex to be coloured is considered. In the second part of Step 17, v_k is removed from its previous colour class if v_k has to be recoloured during backtracking. Steps 9 and 10, and Steps 15–18 are repeated as long as the vertex counter, k , is smaller than the order of the graph n , and as long as the set of possible colours for each vertex is not empty.

As soon as all the vertices have been coloured ($k > n$ in Step 19 of Algorithm 4), a colouring in fewer colours than before has been obtained. Again, this new upper bound, x^* , in Step 20, is compared to the order of the initial clique in Step 21. As before, if the two values are equal, then the optimal value of $\chi(G)$ has already been obtained and the algorithm terminates at Step 22. Otherwise, this colouring

Algorithm 4 Brown's modified colouring algorithm

Input: A graph G of order n , an upper bound x on $\chi(G)$ as well as a proper colouring s^* of G determined by a heuristic colouring algorithm, and the order w of an initial clique.

Output: The value of $\chi(G)$ as well as the resulting colouring of G .

```

1: if  $x = w$  then
2:   STOP  $\leftarrow$  true,  $\chi(G) \leftarrow x$ ,  $s = s^*$ 
3: else [ $x > w$ ]
4:   Determine colouring order  $v_1, \dots, v_n$ , where  $v_1, \dots, v_w$  are the clique vertices
5:   Colour and label clique vertices  $(v_1, \dots, v_w)$ 
6:   STOP  $\leftarrow$  false, BACK  $\leftarrow$  false,  $k \leftarrow w + 1$ 
7:   while not STOP do
8:     if not BACK then
9:        $\tilde{x}_k \leftarrow$  number of colours used in the partial colouring of level  $k - 1$ 
10:       $\mathcal{X}(v_k) \leftarrow$  set of colours from  $\{1, \dots, \min(\tilde{x}_k + 1, x - 1)\}$  which may be used to colour  $v_k$ 
11:    else [BACK = true]
12:       $c \leftarrow$  colour of  $v_k$ 
13:       $\mathcal{X}(v_k) \leftarrow \mathcal{X}(v_k) - \{c\}$ ; remove label from  $v_k$  if there is one
14:    end if
15:    if  $\mathcal{X}(v_k) \neq \emptyset$  then
16:       $j \leftarrow$  minimal colour in  $\mathcal{X}(v_k)$ 
17:       $\mathcal{C}_j \leftarrow \mathcal{C}_j \cup \{v_k\}$ ; remove  $v_k$  from  $\mathcal{C}_\ell$  if  $v_k \in \mathcal{C}_\ell$  for some  $\ell \neq j$ 
18:       $k \leftarrow k + 1$ 
19:    if  $k > n$  then
20:      Let  $x^*$  be the number of colours used in this colouring  $s^*$ ;
21:      if  $x^* = w$  then
22:        STOP  $\leftarrow$  true,  $\chi(G) \leftarrow x^*$ ,  $s \leftarrow s^*$ 
23:      else [ $x^* > w$ ]
24:         $\mathcal{X}(v_i) \leftarrow \mathcal{X}(v_i) - \{x^*, \dots, x\}$  for all coloured vertices  $v_i$ ,  $w < i \leq n$ 
25:         $x \leftarrow x^*$ 
26:         $k \leftarrow$  smallest  $i$  for which  $v_i \in \mathcal{C}_x$ 
27:        Remove all labels from  $v_k, \dots, v_n$  if there are any
28:        BACK  $\leftarrow$  true
29:      end if
30:    else [ $k \leq n$ ]
31:      BACK  $\leftarrow$  false
32:    end if
33:    else [ $\mathcal{X}(v_k) = \emptyset$ ]
34:      BACK  $\leftarrow$  true
35:    end if
36:    if BACK then
37:      Label all unlabelled vertices  $v_i$  such that (i)  $i < k$ , (ii) smallest  $i = j$  such that  $v_j \in \mathcal{C}_\ell$  and
38:       $v_j \in N(v_k)$  for all  $\ell = 1, \dots, x$ 
39:       $k \leftarrow$  maximum  $i$  such that  $v_i$  is labelled
40:      if  $k \leq w$  then
41:        STOP  $\leftarrow$  true,  $\chi(G) \leftarrow x^*$ ,  $s = s^*$ 
42:      end if
43:    end if
44:  end while
45: return  $\chi(G)$ ,  $s = (\mathcal{C}_1, \dots, \mathcal{C}_x)$ , where  $x = \chi(G)$ 

```

is saved as a new possible optimal colouring, and all the relevant variables in Steps 24–28 are reset as preparation to search for a colouring with even fewer colours than the one just obtained. In this regard, the excessive colours are deducted from the sets of possible colours for each vertex in Step 24. In Step 26 the vertex counter is set to the first vertex that was coloured with colour x^* (since this vertex is the first

one that needs to be recoloured in order to obtain a colouring in fewer than x^* colours) and in Step 28 the boolean variable, BACK, is set to `true` to indicate that backtracking is necessary.

Whenever the set of colours $\mathcal{X}(v_k)$ is empty when v_k is the vertex to be coloured next, backtracking is also necessary. Backtracking in this case is also initiated by the boolean variable BACK. BACK is set to `true` at Step 34 in this case. When backtracking is necessary, vertices are labelled in Step 37. Vertices that may have to be recoloured in order to either recolour the current vertex if a new colouring in fewer colours than before is sought, or to be able to colour the current vertex for which the set of possible colours was empty, are all labelled to indicate that the algorithm may need to backtrack to these vertices. One would like to avoid backtracking too far back, therefore the vertex counter in Step 38 is set to the vertex closest to the current vertex in the colouring order that needs to be recoloured in order possibly to colour the current vertex. If the vertex counter, k , is set to one of the clique vertices, indicating that this vertex needs to be recoloured in order possibly to obtain a colouring in fewer colours, then there does not exist a colouring in fewer colours than the best colouring obtained thus far. In this case the algorithm terminates at Step 40 with the previously obtained best colouring that was saved as optimal colouring.

Of course, only graphs containing edges would be coloured using an algorithm (edgeless graphs may all be coloured with one colour). Therefore the smallest clique in a graph G for which a $\chi(G)$ -colouring is sought, is K_2 . Thus, the root of the tree constructed during execution of Brown's modified colouring algorithm will always contain the colouring of at least two vertices (see, for example, Figure 2.18) and the height of the tree will be at most $n - 2$. Let the first vertex, v_1 , be coloured with colour 1 and the second vertex, v_2 , be coloured with colour 2. At worst, the third vertex, v_3 , may be coloured with either colour 1 or colour 2. Level 1, therefore, contains at most two vertices. (If v_3 should be coloured with colour 3, then this vertex is adjacent to both v_1 and v_2 , but then a clique K_3 in G exists. These clique vertices may then all be coloured at level 0, in which case the maximum height of the tree is $n - 3$ with less branching and thus fewer leaves than before.) At each level from here onwards, a maximum of one additional colour may be used in order to avoid redundancy. For example, level 1 contains two colours and consequently only a third colour may be added at level 2, so that each of the two vertices on level 2 may contain a maximum of three children, resulting in a maximum of six vertices on level 2. Of these six vertices, four vertices may have a maximum of three children each, while the other two vertices may have a maximum of four children each. When all vertices, except one, have already been coloured, the last vertex will be coloured with the first available colour. Hence, the number of leaves in the tree constructed during the course of execution of Brown's modified colouring algorithm will be the same as the number of nodes in the second last level of the tree (at most level $n - 3$). If branching continues as described above, the highest term in the function for the number of leaves in the tree constructed while executing Brown's modified colouring algorithm, is $(n - 6)!$ However, it is very difficult to quantify the tree more accurately, since the number of nodes at each level depends on the adjacency of the vertex to be coloured on that level. Both the number of vertices to which the particular vertex to be coloured is adjacent as well as which vertices the vertex is adjacent, influence the construction of the tree in compliance with Brown's modified colouring algorithm.

For example, the tree constructed during the course of execution of Brown's modified colouring algorithm on a graph of order 7 without considering adjacency as described above may contain a maximum of 74 leaves, while the tree constructed for the sparse graph P_7 contains 15 leaves, the tree constructed for the $(7, 13)$ -graph \overline{G}_1 in Figure 2.1(b) with density 0.62 contains seven leaves (see Figure 2.18) and the tree constructed for the dense graph \overline{C}_7 contains only five leaves. Furthermore, Brown [21, Tables 1–3] reported average running times for random graphs of orders 20–40 with densities around 0.25, 0.4 and 0.75 (165 graphs in total) as ranging between 0.0046 and 5.3166 seconds on a UNIVAC 1108 where the algorithm was coded in Fortran. Finally, tests by Brelaz [13], shows an average improvement of 35% on the running times of Brown's modified colouring algorithm on the original algorithm by Brown.

The working of Brown's modified colouring algorithm is illustrated in the following example.

Example 2.1 *Considering the same graph that was used to illustrate the working of the heuristic colouring algorithms, namely the Grötzsch graph given in Figure 2.10, an upper bound x on $\chi(G)$ has already been obtained, viz. four. Any of the colourings in Figures 2.10(b), 2.11(a)ℰ(b), 2.12(d) or 2.13(b) may be given as input s^* to Algorithm 4. Furthermore, the largest clique in the Grötzsch graph is K_2 , so*

that $w = 2$. Since $x = 4 > 2 = w$ in Step 1 of Brown's modified colouring algorithm, the algorithm will attempt to find a proper colouring in three colours. Suppose the colouring order determined in Step 4 of the algorithm is v_1, \dots, v_{11} and let v_1 and v_2 be the clique vertices. The first part of the algorithm is summarised in Table 2.3.

After the steps listed in Table 2.3, Brown's modified colouring algorithm proceeds with steps similar to those from line 7 up to the end of Table 2.3, where v_4 is coloured with colour 3 instead of colour 2. This leads to backtracking to v_3 , and eventually, backtracking to v_2 . Thus, the algorithm terminates at Step 40 and the upper bound x and colouring determined by the heuristic colouring algorithm is given as the final output. ■

Before another exact graph colouring algorithm is discussed, the behaviour of the chromatic number under graph constructions is considered first. For the union of two graphs G and H it holds intuitively that $\chi(G \cup H) = \max\{\chi(G), \chi(H)\}$ [112, p 193], while a proof of the result $\chi(G + H) = \chi(G) + \chi(H)$ for the join of two graphs G and H may be found in [53, p 329]. A proof of the corresponding result for the cartesian product of two graphs G and H , namely $\chi(G \times H) = \max\{\chi(G), \chi(H)\}$, may be found in [112, p 194]. Using the cartesian product of two graphs, Berge [8] proved that a graph G is x -colourable if and only if $G \times K_x$ has an independent set of size $p(G)$. Finally, $\chi(H) \leq \chi(G)$ if H is a subgraph of a graph G . If, however, $\chi(H) < \chi(G)$ for every proper subgraph H of a graph G with $\chi(G) = x \geq 2$, then G is said to be **critical x -chromatic** [109, p 87]. For example, K_1 is the only critical 1-chromatic graph, K_2 is the only critical 2-chromatic graph, odd cycles are critical 3-chromatic, even order wheels are critical 4-chromatic and every complete graph of order n is critical n -chromatic [31, p 205], [53, p 330]. Furthermore, every graph with chromatic number $x \geq 2$ contains a critical x -chromatic subgraph and every critical x -chromatic graph is connected [57, p 25]. The next theorem, due to Dirac [37], gives a lower bound on the minimum degree of a critical x -chromatic graph and a proof of the theorem may be found in [29, p 294].

Theorem 2.11 *Let G be a critical x -chromatic graph, $x \geq 2$. Then, $\delta(G) \geq x - 1$.* ■

Using the fact that any x -chromatic graph contains a critical x -chromatic subgraph, the following theorem may be formulated (a proof may be found in [31, p 207]).

Theorem 2.12 *An x -chromatic graph G has at least x vertices v_i such that $\deg_G(v_i) \geq x - 1$.* ■

The last exact graph colouring algorithm considered in this section, employs the idea of critical x -chromatic subgraphs. This algorithm, due to Herrmann and Hertz [63], attempts to find the smallest possible critical chromatic subgraph H of the graph G for which the chromatic number needs to be determined. Hopefully, the order of H is much less than G , so that the time used to determine the chromatic number by an exact method is decreased when applied to H instead of G . The algorithm by Herrmann and Hertz [63] makes use of another exact colouring algorithm as well as a heuristic colouring algorithm which, for discussion purposes, will be called EXACT and HEURISTIC respectively. Combining EXACT and HEURISTIC with the notion of critical chromatic subgraphs, leads to the algorithm by Herrmann and Hertz [63], called the **Herrmann–Hertz algorithm**. This algorithm is given in pseudo-code as Algorithm 5.

Given a graph G as input, the Herrmann–Hertz algorithm commences with a few initializations in Steps 1–3, including the determination of an upper bound x on the chromatic number of G with the aid of a heuristic colouring algorithm in Step 2. Steps 4–14 are what Herrmann and Hertz [63] called the **descending phase** and attempt to find a critical chromatic subgraph H of G . This is done by repeatedly removing a vertex v from a subgraph H of G , originally equal to the graph G itself, in order to obtain a new subgraph H of lesser order, until the upper bound k on the chromatic number of H is less than x . To obtain the final subgraph H , all the vertices in $V(G)$ are tested to determine whether or not the vertex may be removed. The list P is used to keep track of which vertices have already been tested and which ones have not yet been tested — initially P contains all the vertices of G (Step 3) and as a vertex is tested, whether or not it is removed from subgraph H , the vertex is removed from P (Step 5). The order in which the vertices should be tested is determined as the algorithm proceeds by selecting

Step	Action	BACK	Vertices labelled	Vertices labelled thus far
5	$v_1 \in \mathcal{C}_1, v_2 \in \mathcal{C}_2$		v_1, v_2 (clique)	v_1, v_2 (clique)
6	$k = 3$	false		v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_3) = \{\text{colour 1, colour 3}\}$			v_1, v_2 (clique)
15–18	$v_3 \in \mathcal{C}_1, k = 4$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_4) = \{\text{colour 2, colour 3}\}$			v_1, v_2 (clique)
15–18	$v_4 \in \mathcal{C}_2, k = 5$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_5) = \{\text{colour 3}\}$			v_1, v_2 (clique)
15–18	$v_5 \in \mathcal{C}_3, k = 6$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_6) = \{\text{colour 1, colour 3}\}$			v_1, v_2 (clique)
15–18	$v_6 \in \mathcal{C}_1, k = 7$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_7) = \{\text{colour 2, colour 3}\}$			v_1, v_2 (clique)
15–18	$v_7 \in \mathcal{C}_2, k = 8$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_8) = \{\text{colour 1, colour 3}\}$			v_1, v_2 (clique)
15–18	$v_8 \in \mathcal{C}_1, k = 9$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_9) = \{\text{colour 2}\}$			v_1, v_2 (clique)
15–18	$v_9 \in \mathcal{C}_2, k = 10$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_{10}) = \{\text{colour 3}\}$			v_1, v_2 (clique)
15–18	$v_{10} \in \mathcal{C}_3, k = 11$			v_1, v_2 (clique)
9 & 10	$\mathcal{X}(v_{11}) = \emptyset$			v_1, v_2 (clique)
15 & 34		true		v_1, v_2 (clique)
37 & 38	$k = 10$		v_6, v_7, v_{10}	v_1, v_2 (clique); v_6, v_7, v_{10}
12 & 13	$c = \text{colour 3}, \mathcal{X}(v_{10}) = \emptyset$		remove v_{10} label	v_1, v_2 (clique); v_6, v_7
15 & 34		true		v_1, v_2 (clique); v_6, v_7
37 & 38	$k = 7$		v_4	v_1, v_2 (clique); v_4, v_6, v_7
12 & 13	$c = \text{colour 2}, \mathcal{X}(v_7) = \{\text{colour 3}\}$		remove v_7 label	v_1, v_2 (clique); v_4, v_6
15–17	$v_7 \in \mathcal{C}_3, \mathcal{C}_2 = \mathcal{C}_2 \setminus \{v_7\}$			v_1, v_2 (clique); v_4, v_6
18 & 31	$k = 8$	false		v_1, v_2 (clique); v_4, v_6
9 & 10	$\mathcal{X}(v_8) = \{\text{colour 1, colour 3}\}$			v_1, v_2 (clique); v_4, v_6
15–18	$v_8 \in \mathcal{C}_1, k = 9$			v_1, v_2 (clique); v_4, v_6
9 & 10	$\mathcal{X}(v_9) = \{\text{colour 2}\}$			v_1, v_2 (clique); v_4, v_6
15–18	$v_9 \in \mathcal{C}_2, k = 10$			v_1, v_2 (clique); v_4, v_6
9 & 10	$\mathcal{X}(v_{10}) = \{\text{colour 3}\}$			v_1, v_2 (clique); v_4, v_6
15–18	$v_{10} \in \mathcal{C}_3, k = 11$			v_1, v_2 (clique); v_4, v_6
9 & 10	$\mathcal{X}(v_{11}) = \emptyset$			v_1, v_2 (clique); v_4, v_6
15 & 34		true		v_1, v_2 (clique); v_4, v_6
37 & 38	$k = 10$		v_7, v_{10}	v_1, v_2 (clique); v_4, v_6, v_7, v_{10}
12 & 13	$c = \text{colour 3}, \mathcal{X}(v_{10}) = \emptyset$		remove v_{10} label	v_1, v_2 (clique); v_4, v_6, v_7
15 & 34		true		v_1, v_2 (clique); v_4, v_6, v_7
37 & 38	$k = 7$		–	v_1, v_2 (clique); v_4, v_6, v_7
12 & 13	$c = \text{colour 3}, \mathcal{X}(v_7) = \emptyset$		remove v_7 label	v_1, v_2 (clique); v_4, v_6
15 & 34		true		v_1, v_2 (clique); v_4, v_6
37 & 38	$k = 6$		v_3	v_1, v_2 (clique); v_3, v_4, v_6
12 & 13	$c = \text{colour 1}, \mathcal{X}(v_6) = \emptyset$		remove v_6 label	v_1, v_2 (clique); v_3, v_4
15 & 34		true		v_1, v_2 (clique); v_3, v_4
37 & 38	$k = 5$		v_5	v_1, v_2 (clique); v_3, v_4, v_5
12 & 13	$c = \text{colour 3}, \mathcal{X}(v_5) = \emptyset$		remove v_5 label	v_1, v_2 (clique); v_3, v_4
15 & 34		true		v_1, v_2 (clique); v_3, v_4
37 & 38	$k = 4$		–	v_1, v_2 (clique); v_3, v_4
12 & 13	$c = \text{colour 2}, \mathcal{X}(v_4) = \{\text{colour 3}\}$		remove v_4 label	v_1, v_2 (clique); v_3
15–17	$v_4 \in \mathcal{C}_3, \mathcal{C}_2 = \mathcal{C}_2 \setminus \{v_4\}$			v_1, v_2 (clique); v_3
18 & 31	$k = 5$	false		v_1, v_2 (clique); v_3

TABLE 2.3: The values of the parameters during the first part of Brown's modified colouring algorithm applied to the Grötzsch graph in Figure 2.10, considered in Example 2.1.

Algorithm 5 Herrmann–Hertz algorithm**Input:** A graph G of order n .**Output:** The value of $\chi(G)$.

```

1: STOP  $\leftarrow$  false
2: Determine an upper bound  $x$  on  $\chi(G)$  by means of HEURISTIC
3:  $H \leftarrow G$ ;  $P \leftarrow V(G)$ 
4: while  $P \neq \emptyset$  do
5:   Choose  $v \in P$  such that  $\deg_H(v) \leq \deg_H(u)$  for all  $u \in P$ ;  $P \leftarrow P \setminus \{v\}$ 
6:   Determine an upper bound  $k$  on  $\chi(H - v)$  by means of HEURISTIC
7:   if  $k = x$  then
8:      $H \leftarrow H - v$ 
9:   end if
10: end while
11: Determine  $k' = \chi(H)$  by means of EXACT
12: if  $k' = x$  then
13:   STOP  $\leftarrow$  true,  $\chi(G) \leftarrow x$ 
14: end if
15: while not STOP do
16:    $L \leftarrow \emptyset$ 
17:    $T \leftarrow V(G) \setminus V(H)$ 
18:   for all  $i = 1, \dots, |T|$  do
19:     Choose  $v \in T$ ;  $T \leftarrow T \setminus \{v\}$ 
20:     Determine an upper bound  $m$  on  $\chi(H + v)$  by means of HEURISTIC
21:     if  $m > k'$  then
22:       Determine  $m' = \chi(H + v)$  by means of EXACT
23:       if  $m' = k' + 1$  then
24:          $L \leftarrow L \cup \{v\}$ 
25:       end if
26:     end if
27:   end for
28:   if  $L \neq \emptyset$  then
29:     Choose  $v \in L$  such that  $|N_{H+v}(v)| \geq |N_{H+u}(u)|$  for all  $u \in L$ 
30:      $H \leftarrow H + v$ ;  $k' \leftarrow k' + 1$ 
31:   else [ $L = \emptyset$ ]
32:     Choose  $v \in V(G) \setminus V(H)$  such that  $|N_{H+v}(v)| \geq |N_{H+u}(u)|$  for all  $u \in V(G) \setminus V(H)$ 
33:      $H \leftarrow H + v$ 
34:   end if
35:   if  $H = G$  then
36:     STOP  $\leftarrow$  true,  $\chi(G) \leftarrow k'$ 
37:   else [ $H \subset G$ ]
38:     if  $k' = x$  then
39:       STOP  $\leftarrow$  true,  $\chi(G) \leftarrow k'$ 
40:     end if
41:   end if
42: end while
43: return  $\chi(G)$ 

```

from amongst the untested vertices in P , one with the smallest degree in the current subgraph H . After all the vertices in $V(G)$ have been tested and removed (if applicable), the exact chromatic number, the parameter k' in the algorithm is determined in Step 11 of the Herrmann–Hertz algorithm for the final subgraph H . Herrmann and Hertz [63] proved the following theorem, which is the motivation for why the algorithm terminates at Step 13 if $k' = x$.

Theorem 2.13 *Let G be a graph and let x be an upper bound on $\chi(G)$. Let H be any subgraph of G . If $\chi(H) = x$, then $\chi(G) = \chi(H)$. \blacksquare*

Therefore, if the chromatic number of H (k' in Algorithm 5) is less than the upper bound x on $\chi(G)$, then the parameter k' in the rest of the Herrmann–Hertz algorithm serves as a lower bound on $\chi(G)$ and is adapted as the algorithm proceeds from Step 15 onwards. Herrmann and Hertz [63] called this last part of the algorithm the **augmenting phase** and here vertices are added back to the subgraph H until either $\chi(H) = x$ and thus $\chi(G) = x$, or $H = G$. At the beginning of each iteration of the augmenting phase the list T contains all the vertices in $V(G)$ that are not contained in the vertex set of the current subgraph H , as assigned in Step 17 of the algorithm. For each iteration of the for-loop spanning Steps 18 and 27, a vertex v in T is selected and removed from T (Step 19). Using HEURISTIC, an upper bound m on $\chi(H + v)$ is determined in Step 20. If this upper bound m is strictly greater than the chromatic number of H , the exact value m' of the chromatic number of $H + v$ is determined in Step 22 of the algorithm with the aid of EXACT. If the chromatic number of $H + v$ is one more than the chromatic number of H (only one vertex was added, thus only one more colour could be required), this vertex v is stored in the list L in Step 24. Thus, the list L contains all vertices that increase the chromatic number if they were to be added to subgraph H . At the completion of the for-loop, if L is not empty, a vertex v is selected from L (Step 29) and added to subgraph H and the chromatic number of the current subgraph H (parameter k') is updated in Step 30 of the algorithm. If L is empty, a vertex in G that is not in H is selected and added to H in Step 33 of the algorithm. In both cases the vertex v with largest number of vertices adjacent to it in H , is selected. Vertices are added to the current subgraph H in this way as long as either all the vertices of G that were not originally in H are added to H , or the chromatic number of the current subgraph H is equal to the upper bound x on the chromatic number of G , in which case the algorithm terminates at Steps 36 and 39, respectively. In the first case the subgraph H is the original graph G and the chromatic number of H , k' , is thus the chromatic number of G . In the second case, if $k' = x$, then $\chi(G) = k'$ by Theorem 2.13.

According to Herrmann and Hertz [63] it is very important to make a good choice with respect to HEURISTIC. If the upper bound x obtained by the specific choice of HEURISTIC in Step 2 of the Herrmann–Hertz algorithm is indeed the exact value for the chromatic number of G , then only one execution of EXACT is required. Furthermore, in this case EXACT is executed on the subgraph H which hopefully has fewer vertices than the original graph G , so that the computation time is less than when EXACT would have been executed on G . On the other hand, if the upper bound x obtained by HEURISTIC in Step 2 of the Herrmann–Hertz algorithm is strictly greater than the exact value for the chromatic number of G , then several executions of EXACT are required, including perhaps an execution of EXACT on the original graph G itself, which would increase the computation time considerably. In the latter case nothing is gained by using the Herrmann–Hertz algorithm instead of Brown’s modified colouring algorithm.

To illustrate the importance of the choice of HEURISTIC, consider the Grötzsch graph, shown in Figure 2.10 as G_8 . First, consider Brelaz’s heuristic as choice for HEURISTIC. The upper bound x on $\chi(G_8)$ in Step 2 of the Herrmann–Hertz algorithm given by Brelaz’s heuristic is 4, as discussed earlier in the section. Any of the vertices in $\{v_6, v_7, v_8, v_9, v_{10}\}$ may be selected in Step 5 of the Herrmann–Hertz algorithm. Suppose v_6 is selected. If an upper bound k on $\chi(G_8 - v_6)$ in Step 6 of the Herrmann–Hertz algorithm is determined by Brelaz’s heuristic, the vertices are coloured in the order $v_1, v_2, v_3, v_4, v_5, v_9, v_{10}, v_{11}, v_8, v_7$ if the convention of selecting the vertex with smallest subscript in case of a tie is again followed. The resulting colouring of $G_8 - v_6$ is given in Figure 2.19(a). In this case the value of k is 3, so that v_6 is not removed from the graph $H = G_8$ in Step 8 of Algorithm 5. Similar 3-colourings are obtained by Brelaz’s heuristic when vertices v_7, v_8, v_9 and v_{10} are selected in Step 5 of Algorithm 5, so that none of these vertices are removed from H . At this stage of execution of the Herrmann–Hertz algorithm, the subgraph H is still equal to G_8 and $P = \{v_1, v_2, v_3, v_4, v_5, v_{11}\}$. Any of the vertices v_1, v_2, v_3, v_4 or v_5 may be considered next in Step 5 of the Herrmann–Hertz algorithm. Suppose v_1 is selected. If an upper bound k on $\chi(G_8 - v_1)$ in Step 6 is determined by Brelaz’s heuristic, the vertices are coloured in the order $v_{11}, v_8, v_4, v_3, v_9, v_5, v_6, v_2, v_7, v_{10}$. The resulting 3-colouring of $G_8 - v_1$ is given in Figure 2.19(b) and again, since $k = 3$, the vertex v_1 is not removed from the graph $H = G_8$ in Step 8 of Algorithm 5 either. Similar 3-colourings are obtained by Brelaz’s heuristic when the vertices v_2, v_3, v_4 and v_5 are selected in Step 5 of Algorithm 5, so that none of these vertices are removed from H .

At this stage of execution of the Herrmann–Hertz algorithm, the subgraph H is still equal to G_8 and v_{11} is the only vertex remaining in the list P . Finally, v_{11} is considered in Step 5 of the Herrmann–Hertz algorithm. Again a 3-colouring of $G_8 - v_{11}$, which is the same graph as the one in Figure 2.12(a), is

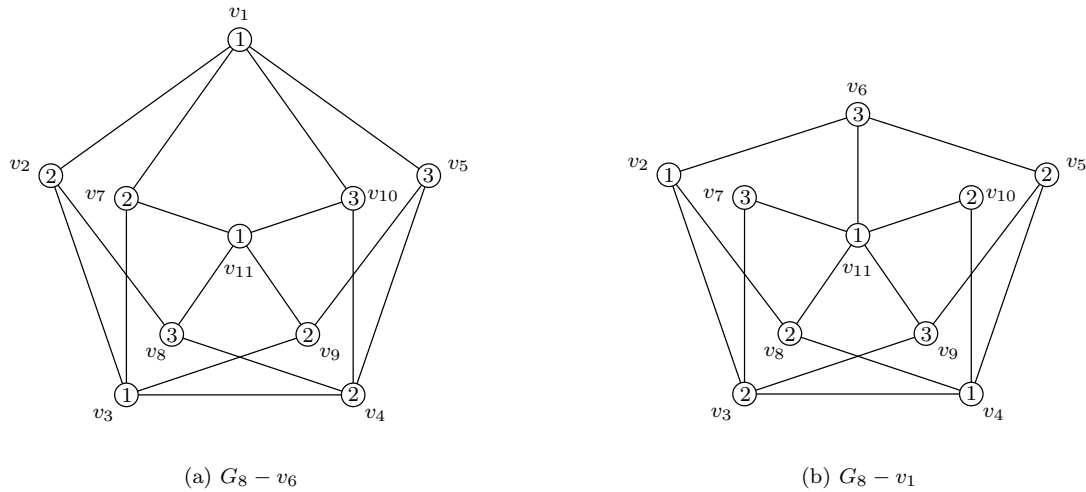


FIGURE 2.19: Proper 3-colourings of the graphs (a) $G_8 - v_6$ and (b) $G_8 - v_1$ as obtained via Brelaz's heuristic during the course of the Herrmann–Hertz algorithm.

obtained by Brelaz's heuristic with colour classes $\mathcal{C}_1 = \{v_1, v_3, v_6, v_8\}$, $\mathcal{C}_2 = \{v_2, v_4, v_7, v_9\}$ and $\mathcal{C}_3 = \{v_5, v_{10}\}$. Therefore, at the beginning of Step 11 of Algorithm 5, $H = G_8$ and EXACT is executed once, after which the Herrmann–Hertz algorithm terminates at Step 13.

If the elementary sequential colouring algorithm (Algorithm 2) were chosen for HEURISTIC, the course of the Herrmann–Hertz algorithm would be different. The sequential colouring algorithm also obtained an upper bound of 4 on $\chi(G_8)$, as discussed earlier in the section. Thus, the value of x in Step 2 of the Herrmann–Hertz algorithm is again 4. When Step 4 of Algorithm 5 commences, as before, any of the vertices in $\{v_6, v_7, v_8, v_9, v_{10}\}$ may be selected in Step 5 of the Herrmann–Hertz algorithm. Again following the convention of selecting the vertex with smallest subscript in case of a tie, v_6 is selected. The resulting 4-colouring of $G_8 - v_6$ obtained by the sequential colouring algorithm is given in Figure 2.20(a).

Since $k = x = 4$ in Step 7 of Algorithm 5, vertex v_6 is removed and the new subgraph is $H = G_8 - v_6$. In the new subgraph H any of the vertices in $\{v_2, v_5, v_7, v_8, v_9, v_{10}\}$ may be selected next in Step 5 of the Herrmann–Hertz algorithm. Suppose v_2 is selected. The resulting 4-colouring of $G_8 - \{v_2, v_6\}$ obtained by the sequential colouring algorithm is given in Figure 2.20(b) and v_2 is removed from H . The vertices v_8, v_1 and v_3 are selected in this sequence in Step 5 of the Herrmann–Hertz algorithm and are not removed from the subgraph $H = G_8 - \{v_2, v_6\}$. At this point $P = \{v_4, v_5, v_7, v_9, v_{10}, v_{11}\}$. Vertex v_5 is selected next and the resulting 4-colouring of $G_8 - \{v_2, v_5, v_6\}$ obtained by the sequential colouring algorithm is given in Figure 2.20(c). Thus, v_5 is removed from H so that now $H = G_8 - \{v_2, v_5, v_6\}$. At the next iteration of the while-loop between Steps 4 and 10 of the Herrmann–Hertz algorithm, v_9 is selected and the resulting 4-colouring of $G_8 - \{v_2, v_5, v_6, v_9\}$ obtained by the sequential colouring algorithm is given in Figure 2.20(d). Since $k = x = 4$ in Step 7 of Algorithm 5, vertex v_9 is removed and the subgraph H is now equal to $G_8 - \{v_2, v_5, v_6, v_9\}$.

The remaining vertices in P are selected in the order v_4, v_7, v_{10}, v_{11} , but none of these vertices are removed from H . At Step 11 the exact value k' of $\chi(G_8 - \{v_2, v_5, v_6, v_9\})$ is determined as 3 by EXACT and the augmenting phase of the Herrmann–Hertz algorithm commences. The first time that the while-loop spanning Steps 15 and 42 of the Herrmann–Hertz algorithm is executed, the list T in Step 17 is $\{v_2, v_5, v_6, v_9\}$. If v_2 is selected at Step 19, then the value of an upper bound m on $\chi(H + v_2)$ in Step 20 is 4 — the resulting 4-colouring of $H + v_2$ obtained by the sequential colouring algorithm is given in Figure 2.21(a).

Since $m = 4$ is greater than $k' = 3$ in Step 21, the exact value m' of $\chi(H + v_2)$ must be determined by EXACT in Step 22. The value of m' is 3 and v_2 is not inserted into the list L in Step 24. Similarly, the resulting 4-colourings of $H + v_5, H + v_6$ and $H + v_9$ obtained by the sequential colouring algorithm are given in Figure 2.21(b), (c) and (d) respectively. Since $m = 4$ in all three cases, the exact values of $\chi(H + v_5)$,

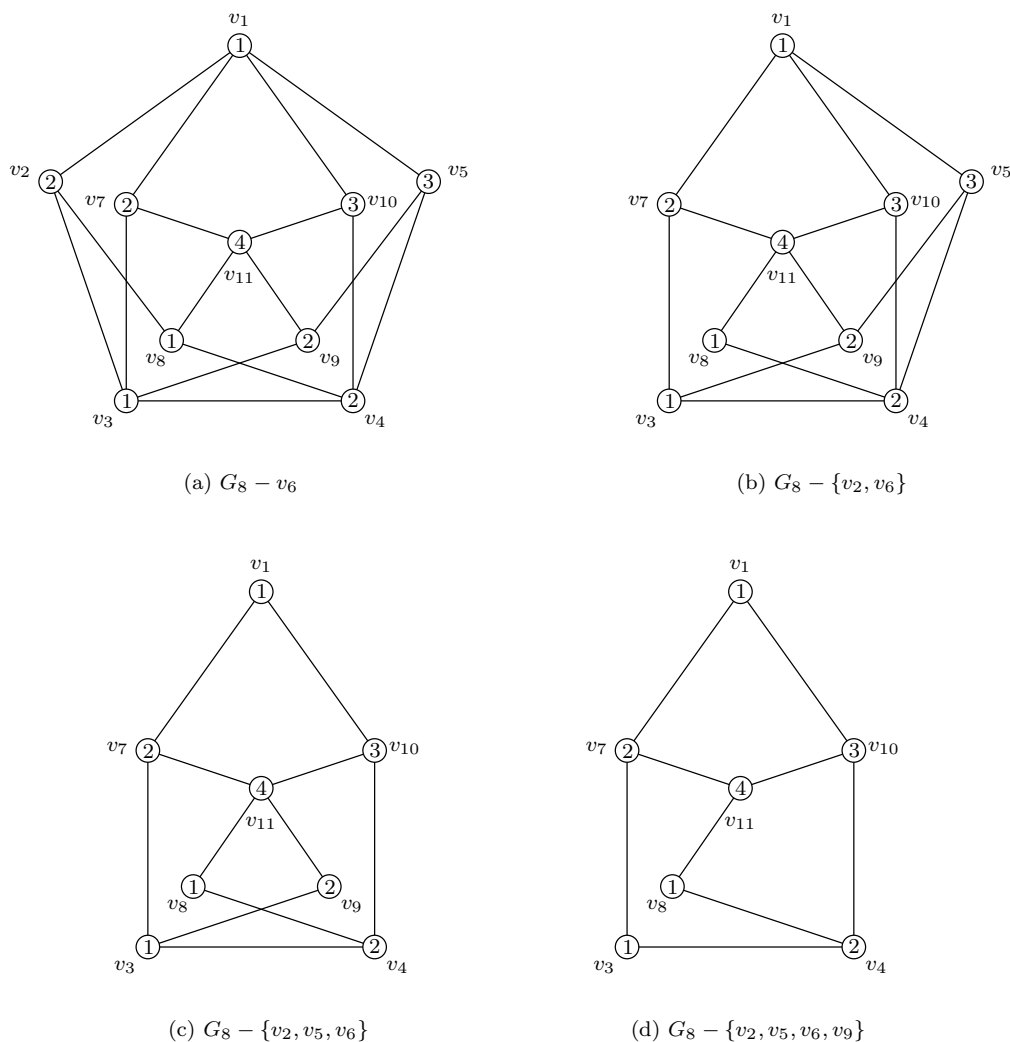


FIGURE 2.20: Proper 4-colourings of the graphs (a) $G_8 - v_6$, (b) $G_8 - \{v_2, v_6\}$, (c) $G_8 - \{v_2, v_5, v_6\}$ and (d) $G_8 - \{v_2, v_5, v_6, v_9\}$ as obtained via the sequential colouring algorithm during the course of the descending phase of the Herrmann–Hertz algorithm.

$\chi(H+v_6)$ and $\chi(H+v_9)$ must be determined by EXACT in Step 22. Again in all three cases the chromatic number is 3 and none of these vertices are inserted in the list L in Step 24. Therefore, the list L is empty at Step 28 and Step 32 must thus be executed. Here, $|N_{H+v_2}(v_2)| = 3$, $|N_{H+v_5}(v_5)| = |N_{H+v_9}(v_9)| = 2$ and $|N_{H+v_6}(v_6)| = 1$. Hence, v_2 is selected and the new subgraph in Step 33 is $H = G_8 - \{v_5, v_6, v_9\}$. Both termination conditions in Steps 35 and 38 are false and the while-loop spanning Steps 15 and 42 of the Herrmann–Hertz algorithm has to be executed again. For this iteration of the while-loop, $T = \{v_5, v_6, v_9\}$ in Step 17. Again both HEURISTIC in Step 20 and EXACT in Step 22 have to be executed for all three choices of v_5 , v_6 and v_9 in Step 19, but as in the previous iteration, none of the vertices are inserted into the list L in Step 24. At Step 32 $|N_{H+v_5}(v_5)| = |N_{H+v_6}(v_6)| = |N_{H+v_9}(v_9)| = 2$. Vertex v_5 is added to H in Step 33. The third and fourth iterations of the while-loop spanning Steps 15 and 42 of the Herrmann–Hertz algorithm also lead to the execution of both HEURISTIC and EXACT and the algorithm terminates at Step 36 with $\chi(G_8) = 4$. Therefore, when using the sequential colouring algorithm as HEURISTIC, the computationally expensive EXACT in this case had to be executed a total of eleven times, including an execution of the original graph G_8 .

The website by Marco Chiarandini [30] is a comprehensive resource on references on graph colouring algorithms. He divided graph colouring algorithms into seven main groups, namely exact methods,

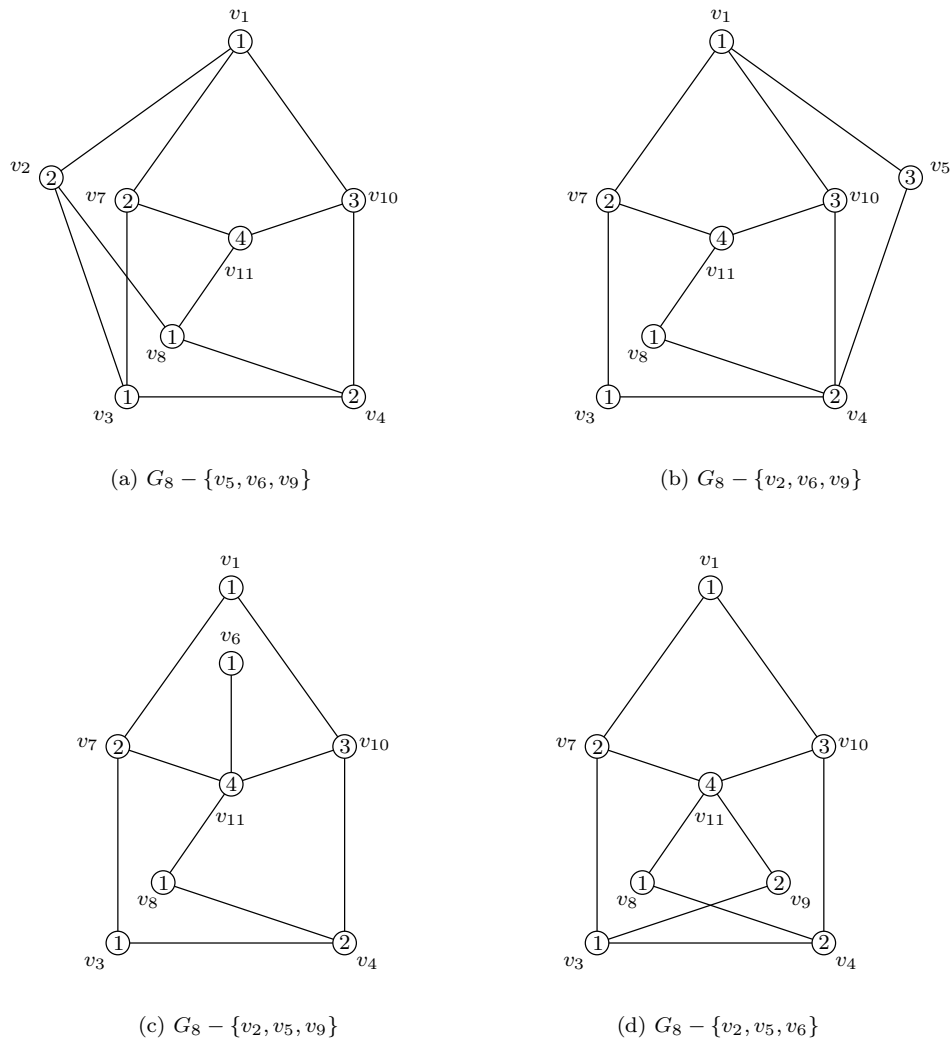


FIGURE 2.21: Proper 4-colourings of the graphs (a) $H + v_2 = G_8 - \{v_5, v_6, v_9\}$, (b) $H + v_5 = G_8 - \{v_2, v_6, v_9\}$, (c) $H + v_6 = G_8 - \{v_2, v_5, v_9\}$ and (d) $H + v_9 = G_8 - \{v_2, v_5, v_6\}$ as obtained via the sequential colouring algorithm during the course of the augmenting phase of the Herrmann–Hertz algorithm.

construction heuristics, simple metaheuristics, hybrid heuristics, evolutionary algorithms, population-based metaheuristics and other approximation methods. Under the heading for exact methods 15 papers are listed, including the paper by Brelaz [13] on Brelaz’s heuristic and the improvement on the exact algorithm by Brown [21] which were both discussed in this section, as well as the paper by Peemöller [92] which pointed out the two errors in Brelaz’s exact algorithm. The paper by Herrmann and Hertz [63] on the Herrmann–Hertz algorithm discussed above is also listed there, as well as an article by Mehorta and Trick [83] describing an approach based on the graph colouring problem being formulated as an integer programming problem.

Finally, in addition to the non-optimality of heuristic methods and the expensive time complexity of exact methods, another limitation of the majority of colouring algorithms (heuristic methods as well as exact methods) is that they only attempt to colour the vertices with as few colours as possible, irrespective of how unevenly the vertices are distributed in number among the colour classes. For example, let \mathcal{C}_i be the set of vertices coloured with colour i . Then, for the proper colouring of the Grötzsch graph using the sequential colouring algorithm given in Figure 2.11(a), $|\mathcal{C}_1| = 4$, $|\mathcal{C}_2| = 4$, $|\mathcal{C}_3| = 2$ and $|\mathcal{C}_4| = 1$. Colouring the Grötzsch graph using the largest–first algorithm, one notices from Figure 2.11(b) that $|\mathcal{C}_1| = 3$, $|\mathcal{C}_2| = 4$, $|\mathcal{C}_3| = 3$ and $|\mathcal{C}_4| = 1$. Using the smallest–last algorithm to colour the Grötzsch graph,

the resulting colouring, as given in Figure 2.13(b), also leads to an “unbalanced” colouring with $|\mathcal{C}_1| = 4$, $|\mathcal{C}_2| = 3$, $|\mathcal{C}_3| = 3$ and $|\mathcal{C}_4| = 1$. However, a “balanced” colouring of the Grötzsch graph was given in Figure 2.10(b) where $|\mathcal{C}_1| = 3$, $|\mathcal{C}_2| = 3$, $|\mathcal{C}_3| = 3$ and $|\mathcal{C}_4| = 2$. This notion of “balancedness” may not be important when seeking the chromatic number of a graph, but in some applications a “balanced” colouring is desirable. For example, in time tabling and scheduling problems (as discussed in §1.3) it would be inconvenient to require large numbers of supervisors and examination rooms one day, and only a few the next day.

2.4 Generalized Colourings

As informally alluded to §1.2 the notion of a proper graph colouring introduced in §2.3 may be generalized in several different ways according to the rule \mathcal{R} for that particular colouring. Some graph theoretic definitions of generalized colourings are introduced in §2.4.1. The section also contains requirements to be satisfied by these colourings as well as known bounds on the minimum number of colours in these particular types of generalized colourings. This is followed by a description of two specific generalized colourings that have attracted significant attention in the literature, namely path colourings in §2.4.2 and clique colourings in §2.4.3. The section closes with a discussion on chromatic sequences in §2.4.4.

2.4.1 Requirements and Bounds for Generalized Colourings

According to Frick [43] and Harary [54] the chromatic number is probably one of the most intensively studied graph parameters. It therefore comes as no surprise that the chromatic number has been generalized in several ways, as far back as 1968 when, to the best knowledge of the author, the first paper on a generalization of the chromatic number appeared. This paper by Chartrand, *et al.* [28] covers the topic of the path–chromatic colourings as informally introduced in §1.2. Since then a significant number of papers have appeared on some form of generalization of the chromatic number. Some of the early generalizations were collected in a paper by Harary [54] in which he proposed that a general formulation be applied to certain generalizations. In this formulation, let P be a property of a hereditary family of graphs \mathcal{G} . Then the specific generalized chromatic number of a graph $G \in \mathcal{G}$ with respect to P is the smallest integer such that each colour class induced subgraph of G satisfies the property P . Harary [54] then listed generalized chromatic numbers of a graph with respect to nine different graph properties, where the property in seven of these cases are such that the colour classes are F –free for the specific choice of the graph F . In the other two cases the property is actually a conjunction of two properties. Unfortunately, not all types of generalizations of the chromatic number can be formulated according to Harary’s scheme. For example, the weighted chromatic number informally introduced in §1.2 cannot be formulated with respect to some properties of a hereditary family of graphs [54].

Brown [19] studied the complexity of generalized graph colourings where the property in Harary’s scheme described above, is that each colour class induced subgraph is F –free, where F is a graph of order at least 3. Let –FxGenCN be the decision problem “Given two graphs G and F and an integer $x \in \mathbb{N}$, can the graph G be coloured with at most x colours, such that each colour class induced subgraph is F –free?” Note that if $F = K_2$ then –FxGenCN is **CN**. Brown [19] conjectured that for any graph F of order at least 3, –FxGenCN is **NP–complete** for any $x \geq 2$, but was unable to prove this. However, he was able to prove the following theorem by polynomial time reducing **CN** to –FxGenCN .

Theorem 2.14 –FxGenCN is **NP–complete**

- (a) for $x \geq 3$ if $F = G + H$, where $G \neq K_0$ and $H \neq K_0$, and
- (b) for $x \geq 2$ if F or \overline{F} is 2–connected. ■

The only family of graphs excluded in Theorem 2.14 is the class of graphs for which the graph F and its complement is 1–connected [19].

Even if one restricts oneself to generalized chromatic numbers that may be formulated according to Harary’s scheme described above, there are still many variations. Therefore, generalized colourings considered in this dissertation are assumed to satisfy a further restriction that the property in Harary’s

scheme is that some graph parameter must be less than a given integer number, as suggested by Frick [43]. With this in mind, the following definitions on generalized colourings are applicable. Given a graph parameter λ and $k \in \mathbb{N}_0$, a k -**admissible colouring** of a graph G (with respect to λ) is defined as a colouring of G such that each colour class \mathcal{C} satisfies $\lambda(\langle \mathcal{C} \rangle) \leq k$, that is, the colouring rule \mathcal{R} is determined by the parameter λ . If $x \in \mathbb{N}$ colours are used for a k -admissible colouring of a graph G with respect to a graph parameter λ , the colouring is referred to as a $\lambda(k, x)$ -**colouring** of G . A graph is said to be $\lambda(k, x)$ -**colourable** if a $\lambda(k, x)$ -colouring of the graph exists. The smallest integer x for which a $\lambda(k, x)$ -colouring of G exists (for some fixed value of k), is called the $\lambda(k)$ -**chromatic number** of G , denoted by $\chi_k^\lambda(G)$, and such an optimal colouring is called a $\chi_k^\lambda(G)$ -**colouring** of G . Finally, a graph G for which $\chi_k^\lambda(G) = x$, is referred to as $\lambda(k, x)$ -**chromatic**, and such a graph is said to be $\chi_k^\lambda(G)$ -**colourable** [22, p 2].

The maximum degree of a graph falls within the class of graph parameters for generalized colourings described above which satisfy the following requirements formulated by Brown and Corneil [20] and reformulated by Frick [43] in order to resemble results on a proper colouring more closely:

- I. There must be an integer $k_0 \in \mathbb{N}_0$ such that, for each colour class \mathcal{C} in a k -admissible colouring of a graph G with respect to λ , $\langle \mathcal{C} \rangle$ is an edgeless graph if and only if $\lambda(\langle \mathcal{C} \rangle) = k_0$, while $\langle \mathcal{C} \rangle$ is the empty graph, K_0 , if and only if $\lambda(\langle \mathcal{C} \rangle) < k_0$.
- II. For every integer $k \geq k_0$, there exists a graph G for which $\lambda(G) = k$.
- III. If H is a subgraph of a graph G , then $\lambda(H) \leq \lambda(G)$.
- IV. If G and H are graphs satisfying $\lambda(G) \leq k$ and $\lambda(H) \leq k$, then $\lambda(G \cup H) \leq k$.

Requirement I ensures that there exists an integer k_0 such that $\chi_{k_0}^\lambda(G) = \chi(G)$ for any graph G . Brown and Corneil [20, Theorem 1.2] proved by induction that Requirements II and III ensure that for every $x \in \mathbb{N}$, $k \in \mathbb{N}_0$ and $k \geq k_0$, there exists a graph G for which $\chi_k^\lambda(G) = x$. Requirement III also ensures that $\chi_k^\lambda(H) \leq \chi_k^\lambda(G)$ for any subgraph H of a graph G . Finally, Requirement IV ensures that if $\chi_k^\lambda(G) = x$ for any graph G , then in any $\lambda(k, x)$ -colouring of G , any two distinct colour classes are joined by an edge [43].

Using Requirements I and III above, the following lemma on $\lambda(k, x)$ -chromatic graphs, which originated as a note by Brown and Corneil [20], is proved in [22, p 3].

Lemma 2.1 *If a graph G is $\lambda(k, x)$ -chromatic, then $\chi_k^\lambda(G - v) = x$ or $\chi_k^\lambda(G - v) = x - 1$ for any $v \in V(G)$. ■*

If, however, for a graph G , $\chi_k^\lambda(H) < \chi_k^\lambda(G)$ for some fixed value of k and *every* proper subgraph H of G , then G is called **critical $\lambda(k, x)$ -chromatic**. When $\lambda(k, x)$ -chromatic graphs are studied, it often suffices to consider only those graphs that are critical $\lambda(k, x)$ -chromatic [43]. The following two lemmas concerning critical $\lambda(k, x)$ -chromatic subgraphs have been proved by Brown and Corneil [20, Lemma 2.2 & Theorem 2.3].

Lemma 2.2 *Any $\lambda(k, x)$ -chromatic graph contains a critical $\lambda(k, x)$ -chromatic subgraph. ■*

Lemma 2.3 *Let G be a $\lambda(k, x)$ -chromatic graph. Then G contains a critical $\lambda(k, \ell)$ -chromatic subgraph for all $\ell \leq x$. ■*

Besides the papers cited so far ([19, 20, 22, 43, 54]), the reader is also referred to a few more papers on the theory of generalized colourings without considering only one specific generalization of the chromatic number. For example, Mynhardt and Broere [88] studied different restrictions on the graph F for which the generalized chromatic number of a graph, where the generalized graph colourings have the property that each colour class induced subgraph is F -free, can be made arbitrarily large. Also, Bollobás and West [10] extended the theorem by Erdős that graphs with a large chromatic number and large girth do exist, to generalized F -free colourings. Finally, Albertson, *et al.* [1, Table 1] provides a list of properties for which the generalized chromatic number has been studied.

Instead of concentrating on generalized graph colourings having the property that each colour class induced subgraph should be F -free, the specific generalized colourings considered in the remainder of this section, *i.e.* those satisfying the requirements I–IV above, are considered next. Examples of graph parameters satisfying requirements I–IV include:

1. the path number $\tau(G)$ of a graph G , with $k_0 = 1$,
2. the clique number $\omega(G)$ of a graph G , with $k_0 = 1$,
3. the maximum degree $\Delta(G)$ of a graph G , with $k_0 = 0$, and
4. the degeneracy $\rho(G)$ of a graph G , with $k_0 = 0$.

Thus, $\chi_1^\tau(G) = \chi_1^\omega(G) = \chi_0^\Delta(G) = \chi_0^\rho(G) = \chi(G)$ [43]. The $\tau(k)$ -**chromatic number** of a graph G , $\chi_k^\tau(G)$, and the $\omega(k)$ -**chromatic number** of a graph G , $\chi_k^\omega(G)$, were informally introduced in §1.2. In-depth discussions of all four of these graph parameters are beyond the scope of this dissertation. Instead, as an introduction to the results obtained for generalized chromatic numbers, short literature surveys on two of these graph parameters, namely $\chi_k^\tau(G)$ and $\chi_k^\omega(G)$ are presented in §2.4.2 and §2.4.3 respectively. These two parameters were chosen because $\chi_k^\tau(G)$ is the first generalization of the chromatic number that the author is aware of and a significant number of results on $\chi_k^\omega(G)$ is already known. The k -**th degenerate chromatic number** of a graph G , $\chi_k^\rho(G)$, is also known as the **Lick–White (vertex partition) number** named after its originators, and includes the so-called **point-arboricity** of a graph ($\chi_1^\rho(G)$). The notion of degenerate chromatic numbers of a graph was defined in 1970 by Lick and White [78]. Since then a few more papers on this topic have been published and the reader is referred to [85, 86] in this regard. Results that have appeared in the literature on the topic of this dissertation, namely the d -**th maximum degree chromatic number** of a graph G , $\chi_d^\Delta(G)$, are summarised in Chapter 3. There are, however, a few bounds known for the $\lambda(k)$ -chromatic number of a graph G for any graph parameter λ satisfying Requirements I to IV. These bounds are considered next.

Observe that for all the above graph parameters it follows that $\lambda(K_n) = n - 1 + k_0$ for the complete graph K_n . An exact value of the $\lambda(k)$ -chromatic number of K_n for a graph parameter λ satisfying the requirement that $\lambda(K_n) = n - 1 + k_0$ over and above the requirements I–IV is given in the following theorem for which a proof may be found in [22, p 6].

Theorem 2.15 *If λ is a graph parameter satisfying Requirements I–IV as well as the additional requirement that $\lambda(K_n) = n - 1 + k_0$ for $n \in \mathbb{N}$, then $\chi_k^\lambda(K_n) = \lceil n/(k + 1 - k_0) \rceil$. ■*

From the $\lambda(k)$ -chromatic number of a complete graph, K_n , in Theorem 2.15, the following upper bound on the $\lambda(k)$ -chromatic number of a general graph may be derived, as was done in [22, p 6].

Corollary 2.2 *If λ is a graph parameter satisfying Requirements I–IV as well as the additional requirement that $\lambda(K_n) = n - 1 + k_0$ for $n \in \mathbb{N}$, then for any graph G ,*

$$\chi_k^\lambda(G) \leq \left\lceil \frac{p(G)}{k + 1 - k_0} \right\rceil. \quad \blacksquare$$

The following lower bound on the $\lambda(k)$ -chromatic number of a graph G may be established in terms of the clique number of G , $\omega(G)$. This bound is a generalization of the first lower bound for the classical chromatic number in Theorem 2.8 and a proof of this result may be found in [22, p 6].

Theorem 2.16 *If λ is a graph parameter satisfying Requirements I–IV as well as the additional requirement that $\lambda(K_n) = n - 1 + k_0$ for $n \in \mathbb{N}$, then for any graph G ,*

$$\chi_k^\lambda(G) \geq \left\lceil \frac{\omega(G)}{k + 1 - k_0} \right\rceil. \quad \blacksquare$$

Unfortunately, the result of Theorem 2.16 is not a very good bound, since the difference between $\chi_k^\lambda(G)$ and $\omega(G)$ can be made arbitrarily large as indicated by the next theorem. Given a graph F , the result in the original theorem by Folkman [41, Theorem 2] is closely related to the generalized chromatic number of a graph where the generalized colouring is such that the colour classes should be F -free. Theorem 2.17 restates this result by Folkman in terms of generalized graph colourings formulated by means of a graph parameter.

Theorem 2.17 *Let λ be any graph parameter and let F be a graph such that $\lambda(F) > k$. Then, for every integer $x \geq 1$, there exists a graph G such that $\chi_k^\lambda(G) > x$ and $\omega(G) = \omega(F)$. ■*

2.4.2 Path Colourings

In **path colourings** the colouring rule \mathcal{R} is the requirement that for each colour class \mathcal{C} , $\tau(\langle \mathcal{C} \rangle) \leq k$ for some specified $k \in \mathbb{N}$, and colourings satisfying this rule are also referred to as $\tau(k, x)$ -colourings. In other words, Harary [54] formulated the graph property in path colourings to be such that the graph induced by each colour class should be P_{k+1} -free. The k -th path chromatic number of a graph G , $\chi_k^\tau(G)$, is then the smallest integer x for which a $\tau(k, x)$ -colouring of G exists (for some fixed value of k). Since $P_2 = K_1 + K_1$, $P_3 = K_1 + \overline{K}_2$ and \overline{P}_n is 2-connected for $n \geq 5$, the problem $\text{-P}_n\mathbf{xGenCN}$ is **NP-complete** by Theorem 2.14 for (a) $k \geq 3$ if $n = 2, 3$ and for (b) $k \geq 2$ if $n \geq 5$. Brown [19, Proposition 4.5] additionally proved that $\text{-P}_4\mathbf{xGenCN}$ is **NP-complete** for all $k \geq 3$ by polynomial time reducing **CN** to $\text{-P}_4\mathbf{xGenCN}$.

As mentioned before, the k -th path chromatic number of a graph G was introduced by Chartrand, *et al.* [28] and two of their first results are summarised in the following theorem.

Theorem 2.18 *For any graph G and any integer $k \geq 1$, $1 \leq \chi_{k+1}^\tau(G) \leq \chi_k^\tau(G)$. ■*

The lower bound in Theorem 2.18 is obtained when $k \geq \tau(G)$ and the graph may thus be coloured with one colour. It also follows that if G has order n , then $\chi_k^\tau(G) = 1$ for all $k \geq n$ [28, Proposition 2]. The second inequality in Theorem 2.18 states that $\chi_k^\tau(G)$ is non-increasing as the parameter k increases [28, Proposition 1]. The classical chromatic number may be bound in terms of the k -th path chromatic number, as indicated in the following theorem [28, Corollary 3d].

Theorem 2.19 *For any graph G and any integer $k \geq 1$, $\chi_k^\tau(G) \leq \chi(G) \leq k\chi_k^\tau(G)$. ■*

Chartrand, *et al.* [28, Theorem 2] also established an upper bound on the $\tau(k)$ -chromatic number of a graph G in terms of the parameter $\tau(G)$.

Theorem 2.20 *For any graph G and any integer $1 \leq k \leq \tau(G)$, $\chi_k^\tau(G) \leq \lfloor (\tau(G) - 1 - k)/2 \rfloor + 2$. ■*

However, Frick and Bullock [45, Theorem 2.4] improved on this bound on $\chi_k^\tau(G)$ in terms of the parameter $\tau(G)$ for $k \geq 2$, as stated in the following theorem.

Theorem 2.21 *Let G be any graph and let k be any integer such that $2 \leq k \leq \tau(G)$. Then $\chi_k^\tau(G) \leq \lceil (\tau(G) - k) / \lceil (2k + 2)/3 \rceil \rceil + 1$. ■*

Frick and Bullock [45, Corollary 2.7 & Theorem 2.10] also determined a further bound on $\chi_k^\tau(G)$ in terms of the parameter $\tau(G)$ for various values of k , as listed in the following theorem.

Theorem 2.22 *For any graph G ,*

- (i) $\chi_k^\tau(G) \leq \lceil \tau(G)/k \rceil$ if $k \leq g(G) + 1$, where $g(G)$ is the girth of the graph G as before,
- (ii) $\chi_k^\tau(G) \leq \lceil \tau(G)/k \rceil$ if $k \leq 6$, and
- (iii) $\chi_k^\tau(G) \leq \lceil \tau(G)/k \rceil$ for every $k \geq 1$ if $\tau(G) \leq 11$. ■

Since the theory of graph colourings had its origin in the four-colour problem, it seems natural to search for bounds on the $\lambda(k)$ -chromatic number for the class of planar graphs specifically. In the case of path colourings, Chartrand, *et al.* [28, Theorem 5] proved the existence of a planar graph G such that $\chi_k^\tau(G) = 4$ for any $k \in \mathbb{N}$. Hence, Theorem 2.19 together with the four-colour theorem (Theorem 2.7) imply that for any planar graph G , $\chi_k^\tau(G) \leq 4$ for all $k \in \mathbb{N}$.

The interested reader is referred to [7, 70, 114] for further results on path colourings of graphs.

2.4.3 Clique Colourings

If the colouring rule \mathcal{R} requires that no clique of order $k + 1$ is monocoloured, *i.e.* for each colour class \mathcal{C} , $\omega(\langle \mathcal{C} \rangle) \leq k$ for some specified $k \in \mathbb{N}$, then the notion of a **clique colouring**, or a $\omega(k, x)$ -colouring emerges. In the formulation by Harary [54], the graph property is that the colour classes should be K_n -free. The k -th clique chromatic number of a graph G , $\chi_k^\omega(G)$, is then the smallest integer x for which a $\omega(k, x)$ -colouring of G exists (for some fixed value of k). By Theorem 2.14, the problem **$\text{-K}_n\mathbf{xGenCN}$** is **NP-complete** for (a) $k \geq 3$ if $n = 2$ (*i.e.* **CN**) and for (b) $k \geq 2$ if $n \geq 3$, since $K_2 = K_1 + K_1$ and K_n is 2-connected for $n \geq 3$ [19].

Clique colourings were introduced by Sachs [100] and were studied intensively by Frick [14, 15, 16, 17, 44]. A few of the main results on clique colourings are reviewed next.

In §2.4.1 some results on critical $\lambda(k, x)$ -chromatic graphs for a graph parameter λ satisfying Requirements I–IV were discussed. Frick [44, Theorem 6.6] (Broere and Frick [15, Theorems 1–3]) proved the following theorem concerning critical $\omega(k, x)$ -chromatic graphs in particular.

Theorem 2.23

- (a) *The smallest critical $\omega(k, x)$ -chromatic graph is the complete graph $K_{k(x-1)+1}$.*
- (b) *A critical $\omega(k, x)$ -chromatic graph of order $k(x-1) + r$ for $r \in \{2, \dots, k+1\}$ does not exist.*
- (c) *The graph $\overline{C}_{2k+3} + K_{k(x-3)+k-1}$ is the only critical $\omega(k, x)$ -chromatic graph of order $xk + 2$. ■*

Furthermore, Frick [44, Constructions 6.7, 6.9 & Theorems 6.8, 6.10] used various constructions to show that there exist critical $\omega(k, x)$ -chromatic graphs of order greater than $xk + 2$. Frick [44, Theorem 6.13(ii)] (Frick [43, Theorem 15(ii)]) also proved that if G is a critical $\omega(k, x)$ -chromatic graph, then $\delta(G) \geq k(x-1)$, which is a generalization of Theorem 2.11.

The following upper bound on the k -th clique chromatic number may be established in terms of the classical chromatic number and a proof of this result may be found in [44, Lemma 3.4].

Theorem 2.24 *For any graph G and any $k \in \mathbb{N}$, $\chi_k^\omega(G) \leq \lceil \chi(G)/k \rceil$. ■*

The following lower bound on $\chi_k^\omega(G)$ for a graph G in terms of k and $\omega(G)$ follows directly from Theorem 2.16.

Theorem 2.25 *For any graph G and any integer $k \geq 1$, $\chi_k^\omega(G) \geq \lceil \omega(G)/k \rceil$. ■*

It is obvious that, for any complete graph K_n , it follows from Theorems 2.24 and 2.25 that $\lceil \omega(K_n)/k \rceil = \chi_k^\omega(K_n) = \lceil \chi(K_n)/k \rceil$ [44, p 17]. However, as in the case of the bounds on the classical chromatic number in Theorem 2.8, the difference between $\chi_k^\omega(G)$ and the bounds in Theorems 2.24 and 2.25 may be arbitrarily large. First, since the difference between $\chi(G)$ and $\omega(G)$ can be made arbitrarily large as discussed in §2.3, the difference between $\chi_k^\omega(G)$ and $\chi(G)$ can also be made arbitrarily large if $k > 1$, because if $\omega(G) = n$ ($\omega(G) = 2$ in the case of triangle-free graphs) then $\chi_k^\omega(G) = 1$ for all $k \geq n$ [44, p 18]. In the same paper by Sachs [100, §5] in which he introduced the parameter $\chi_k^\omega(G)$, he proved the following theorem which is a generalization of the above fact from §2.3 that the difference between $\chi(G)$ and $\omega(G)$ can be made arbitrarily large.

Theorem 2.26 *For all integers $x \geq 1$ and $k \geq 1$ there exists a graph G satisfying $\chi_k^\omega(G) = x$ and $\omega(G) = k + 1$. ■*

Frick [44, Theorem 3.17] (Broere and Frick [17, Theorem 1]) also proved the following theorem via a construction which supports the fact that there is no upper bound on $\chi_k^\omega(G)$ for a graph G in terms of k and $\omega(G)$ alone.

Theorem 2.27 *For all integers $x \geq 2$ and $k \geq 1$ there exists a graph G satisfying $\chi_k^\omega(G) = x$ and $\omega(G) = n$ if and only if $k + 1 \leq n \leq xk$. ■*

To conclude this section, the values of the $\omega(k)$ -chromatic number for planar graphs are given for all values of k . Thus, let G be any planar graph. Then, from the Four-colour theorem it follows that $\chi_1^\omega(G) = \chi(G) \leq 4$. From Theorem 2.24 it follows directly that $\chi_2^\omega(G) \leq 2$ as well as $\chi_3^\omega(G) \leq 2$ and $\chi_k^\omega(G) = 1$ for all $k \geq 4$ [22, p 51].

2.4.4 Chromatic Sequences

The λ -chromatic sequence of a graph G is the sequence of integers representing the number of colours, x , required in a χ_k^λ -colouring of G as the parameter k increases, *i.e.* the sequence

$$\chi_{k_0}^\lambda(G), \chi_{k_0+1}^\lambda(G), \chi_{k_0+2}^\lambda(G), \dots,$$

where k_0 is that integer satisfying $\chi_{k_0}^\lambda(G) = \chi(G)$. It is obvious that the sequence is non-increasing and that the sequence has an infinite tail of ones, since $\chi_k^\lambda(G) = 1$ for all $k \geq m$ if $\lambda(G) = m$ [43].

Theorems 2.18, 2.19 and 2.20 provide necessary conditions for a sequence of positive integers to be the τ -chromatic sequence of some graph as stated in the following theorem. A proof of this theorem may be found in [22, p 50].

Theorem 2.28 *If x_1, x_2, x_3, \dots is the τ -chromatic sequence of some graph G , then the following conditions are satisfied.*

- (i) $x_n = 1$ for some integer n .
- (ii) $x_i \leq x_j(\lfloor (j-i-1)/2 \rfloor + 2)$ if $2 \leq i \leq j$.
- (iii) $x_1 \leq jx_j$ if $j \geq 1$. ■

To the best knowledge of the author, sufficient conditions for a sequence of positive integers to be the τ -chromatic sequence of some graph, are yet to be found. From condition (ii) of Theorem 2.28, it follows that $x_i \leq 2x_{i+1}$ for $i \geq 2$ [28, Corollary 3c], so that the difference between two consecutive terms x_i and x_{i+1} (with $i \geq 2$) in a τ -chromatic sequence cannot be made arbitrarily large.

Frick [44, Theorem 4.4] (Broere and Frick [14, Theorem 2]) provided a characterization of the ω -chromatic sequence of a graph G . This characterization, given in Theorem 2.29, may be proved by means of a graph construction.

Theorem 2.29 *A sequence of positive integers x_1, x_2, x_3, \dots is the ω -chromatic sequence of some graph G if and only if the following two conditions are satisfied:*

- (i) $x_n = 1$ for some $n \in \mathbb{N}$.
- (ii) $\lceil x_i/(x_j - 1) \rceil > j/i$ for all i and j with $1 \leq i < j \leq n$. ■

Frick [44, Theorem 4.5] (Broere and Frick [14, Theorem 3]) also considered the length of a constant subsequence of an ω -chromatic sequence given in the following theorem.

Theorem 2.30 *Let $i, j, x \in \mathbb{N}$ be such that $i < j$ and $x \geq 2$. Then there exists a graph G for which $\chi_i^\omega(G) = \chi_{i+1}^\omega(G) = \dots = \chi_j^\omega(G) = x$ if and only if $j \leq 2i - 1$. ■*

Finally, Broere and Frick [14, Theorem 4] also proved the following result on consecutive terms of an ω -chromatic sequence.

Theorem 2.31

- (a) *Given $i, x, y \in \mathbb{N}$ with $i \geq 2$ and $x \geq y$ there exists a graph G with $\chi_i^\omega(G) = x$ and $\chi_{i+1}^\omega(G) = y$.*
- (b) *Given $x, y \in \mathbb{N}$ there exists a graph G with $\chi_1^\omega(G) = x$ and $\chi_2^\omega(G) = y$ if and only if $y \leq \lceil x/2 \rceil$. ■*

2.5 Chapter Summary

This chapter was devoted to the basic terminology and most important results in the different areas applicable to this dissertation. First, the necessary fundamentals from graph theory was given in §2.1 in order to facilitate the definition of the notion of graph colouring . This was followed in §2.2 by certain basic complexity theoretic concepts necessary to evaluate and compare graph algorithms, after which the notion of graph colouring was introduced in §2.3. Some well-known graph colouring algorithms found in the literature were also provided in §2.3. The chapter closed with a short literature survey on certain generalized colourings in §2.4 and in particular graph theoretic definitions of generalized colourings, requirements and bounds on generalized chromatic numbers in general in §2.4.1, for path colourings in §2.4.2, and for clique colourings in §2.4.3. Finally the notion of a chromatic sequence was introduced in §2.4.4.

Chapter 3

Maximum Degree Colourings

“Imagination is more important than knowledge...”

Albert Einstein (1879–1955)

“Imagination is the beginning of creation. You imagine what you desire, you will what you imagine and at last you create what you will.”

George Bernard Shaw (1856–1950)

This chapter opens in §3.1 with the specific notion of maximum degree colourings as derived from the notion of generalized colourings given in §2.4.1. This is followed by an overview in §3.2 of what has already appeared in the literature on maximum degree colourings. In §3.3 an inverted strategy towards determining maximum degree colourings is explained. In this strategy one fixes the number of colours that may be used in a maximum degree colouring and then seeks to minimize the maximum degree of the colour-induced subgraphs rather than determining the number of colours in a maximum degree colouring of a graph G for a fixed value of the maximum degree of the colour-induced subgraphs, *i.e.* the objective is to minimize the maximum degree over all the colour-induced subgraphs.

3.1 Maximum Degree Colourings in Graphs

For a **maximum degree colouring** the colouring rule \mathcal{R} is the requirement that for each colour class \mathcal{C} , $\Delta(\langle \mathcal{C} \rangle) \leq d$, where $d \in \mathbb{N}_0$. An x -colouring of a graph G satisfying this requirement is referred to as a $\Delta(d, x)$ -**colouring** of G , *i.e.* a $\Delta(d, x)$ -colouring of a graph G is a d -admissible colouring of G with respect to the parameter Δ . A graph is $\Delta(d, x)$ -**colourable** if a $\Delta(d, x)$ -colouring of the graph exists. The smallest integer x for which there exists a $\Delta(d, x)$ -colouring of G (for some fixed value of $d \in \mathbb{N}_0$) is called the $\Delta(d)$ -**chromatic number** of G , denoted by $\chi_d^\Delta(G)$, and such an optimal colouring is called a χ_d^Δ -**colouring** of G . Finally, a graph G with $\Delta(d)$ -chromatic number x is referred to as $\Delta(d, x)$ -**chromatic**, and such a graph is said to be χ_d^Δ -**colourable**.¹ In the context of the early formulation by Harary [54] given in §2.4.1, the graph property of a maximum degree colouring is that the colour classes should be $K_{1,n}$ -free, where $n = d + 1$ for the specific choice of d in the formulation above.

The following result establishes a growth property of maximum degree chromatic numbers of a graph with respect to an increase in its parameter value.

¹In contrast to the notation of a $\chi_k^\lambda(G)$ -colouring of a graph G and a $\chi_k^\lambda(G)$ -colourable graph G defined in §2.4.1, in the notation of a $\chi_d^\Delta(G)$ -colouring of a graph G and the graph G being $\chi_d^\Delta(G)$ -colourable, the reference to G will be omitted for the sake of brevity when the context is clear.

Theorem 3.1 (Growth property) *For any graph G and any $d \in \mathbb{N}_0$,*

$$1 \leq \chi_{d+1}^\Delta(G) \leq \chi_d^\Delta(G) \leq \chi(G). \quad (3.1.1)$$

Proof: Consider a $\Delta(d, x)$ -colouring of a graph G and let the resulting partition of $V(G)$ into the x colour classes be denoted by $\mathcal{C}_1, \dots, \mathcal{C}_x$, where $x \in \mathbb{N}$.

The lower bound is trivial. This follows from the non-emptiness of the vertex set in the definition of a graph and is attained when $d \geq \Delta(G)$.

From the definition of $\chi_d^\Delta(G)$, the following must hold for $d = 0$: $\deg_{(\mathcal{C}_j)}(v_i) = 0$ for all $v_i \in \mathcal{C}_j$ and all $j = 1, 2, \dots, x$. Thus, all adjacent vertices must be assigned distinct colours (the maximum number of colours to be used), which results in a proper colouring of the graph G , yielding $\chi_0^\Delta(G) = \chi(G)$. Hence, $\chi_d^\Delta(G) \leq \chi(G)$ for all $d \in \mathbb{N}_0$.

Finally, let $\chi_d^\Delta(G) = \chi^*$. Then any (χ^*, d) -colouring of G is also a $(\chi^*, d+1)$ -colouring of G . Therefore $\chi_{d+1}^\Delta(G) \leq \chi^*$. ■

In order to study the complexity of maximum degree colourings, the specific case of $F = K_{1,d+1}$ in the generalized colouring problem -FxGenCN described in §2.4.1, is considered first. Since $K_{1,d+1} = K_1 + \overline{K}_{d+1}$ the problem $\text{-K}_{1,d+1}\text{xGenCN}$ is **NP-complete** by Theorem 2.14(a) for any $x \geq 3$ and $d \geq 0$. However, since $K_{1,d+1}$ is 1-connected and $\overline{K}_{1,d+1}$ is already disconnected, Theorem 2.14(b) does not supply any information about the complexity of determining whether or not a graph G is $\Delta(d, 2)$ -colourable.

Cowen *et al.* [34] studied the complexity of the $\Delta(d, x)$ -colouring problem of a general graph G as well as characterized the complexity of $\Delta(d, x)$ -colouring problems for planar graphs. First, Cowen *et al.* [34, Theorem 4.9] proved that $\text{-K}_{1,d+1}\text{xGenCN}$ is **NP-complete** for $d = 1$ and $x = 2$ even if restricted to a planar graph or a graph G with $\Delta(G) \leq 4$. This proof was achieved by polynomial time reduction from 3-SAT. Cowen *et al.* [34, Theorem 4.10(a)] then proved that $\text{-K}_{1,d+1}\text{xGenCN}$ for planar graphs is **NP-complete** for $d \geq 1$ and $x = 2$ by polynomial time reduction from $\text{-K}_{1,2}\text{2GenCN}$ for planar graphs. They also showed that $\text{-K}_{1,d+1}\text{xGenCN}$ for a graph with maximum degree at most $2(d+1)$ is **NP-complete** with $d \geq 1$ and $x = 2$ by polynomial time reduction from $\text{-K}_{1,2}\text{2GenCN}$ for a graph G with $\Delta(G) \leq 4$. Cowen *et al.* [34] finally established the **NP-completeness** of the problem $\text{-K}_{1,d+1}\text{xGenCN}$ for $d \geq 1$ and $x = 2$.

A $\Delta(0, x)$ -colouring problem reduces to a proper colouring problem and the complexity of such a problem was discussed in §2.3. Finally, in §2.2 the time complexity of Algorithm 1, which determines the maximum degree of a graph G of order n , was ascertained to be $\mathcal{O}(n^2)$. The complexity of testing whether $\Delta(G) \leq d$ for some specified $d \in \mathbb{N}_0$ is constant. Thus, $\Delta(1, x)$ -colouring problems are in the class **P**. The complexity of the problem $\text{-K}_{1,d+1}\text{xGenCN}$ is summarised below.

Theorem 3.2 $\text{-K}_{1,d+1}\text{xGenCN}$ is

- (i) **NP-complete** for $x \geq 3$ and $d \geq 0$, and for $x = 2$ and $d \geq 1$, and
- (ii) in the class **P** for $x = 2$ and $d = 0$, and for $x = 1$ and $d \geq 0$. ■

In the case of planar graphs, the results by Cowen *et al.* [34, §4.1] may be summarised as follows.

Theorem 3.3 $\text{-K}_{1,d+1}\text{xGenCN}$ for planar graphs is

- (i) in the class **P** for $x = 4$ and $d \geq 0$ (a quadratic time algorithm exists to $\Delta(0, 4)$ -colour planar graphs)
- (ii) **NP-complete** for $x = 3$ and $d \in \{0, 1\}$,
- (iii) in the class **P** for $x = 3$ and $d \geq 2$ (a quadratic time algorithm exists to $\Delta(2, 3)$ -colour planar graphs)
- (iv) in the class **P** for $x = 2$ and $d = 0$ (a linear time algorithm exists to $\Delta(0, 2)$ -colour a planar graph, if the colouring exists)
- (v) **NP-complete** for $x = 2$ and $d \geq 1$. ■

3.2 Literature Survey on Maximum Degree Colourings

The notion of a chromatic number was probably first introduced by Brooks in 1941 [18], while, as mentioned in §2.4.1, a generalization of the chromatic number was probably first suggested by Chartrand *et al.* [28] in 1967 for the path-chromatic number specifically. According to Cowen *et al.* [34], the problem of finding the maximum degree-chromatic number² of a graph was first considered independently and almost simultaneously by Andrews and Jacobson [3], Cowen *et al.* [33] and Harary and Jones [55] during the mid 1980s. Since then a substantial amount of work has been done on this parameter. For instance, Andrews and Jacobson [3, Theorem 1] used the upper bound $\chi(G) \leq 1 + \Delta(G)$ in Theorem 2.8 to determine a lower bound on the $\Delta(d)$ -chromatic number of a graph G in terms of the classical chromatic number of G as stated below.

Theorem 3.4 *Let G be a graph and $d \in \mathbb{N}_0$. Then, $\chi_d^\Delta(G) \geq \chi(G)/(d+1)$.* ■

Proof: Consider a χ_d^Δ -colouring of a graph G with $\chi_d^\Delta = x$, and let the resulting partition of $V(G)$ into the x colour classes be denoted by $\mathcal{C}_1, \dots, \mathcal{C}_x$. Then $\Delta(\langle \mathcal{C}_i \rangle) \leq d$ for all $i \in \{1, \dots, x\}$. It therefore follows from Theorem 2.8 that $\chi(\langle \mathcal{C}_i \rangle) \leq 1 + \Delta(\langle \mathcal{C}_i \rangle) \leq 1 + d$ for each $i \in \{1, \dots, x\}$. Hence $\chi(G) \leq x(1+d)$. ■

Theorem 3.4 holds with equality for the complete graph of order $d+1$. This follows from the fact that $\chi(K_{d+1}) = d+1$ and since $\Delta(K_{d+1}) = d$, it follows from Theorem 3.1 that $\chi_d^\Delta(K_{d+1}) = 1$. However, if K_{d+1} is not a subgraph of a graph G , then the lower bound in Theorem 3.4 may be improved. More specifically, Andrews and Jacobson [3, Theorem 2] proved that if $K_{d+1} \not\subseteq G$, then $\chi_d^\Delta(G) \geq \chi(G)/d$ for $d \in \mathbb{N} \setminus \{2\}$.

Andrews and Jacobson [3, §IV] also established the following lower bound on the $\Delta(d)$ -chromatic number of a graph G in terms of the order and the size of G .

Theorem 3.5 *Let G be a (p, q) -graph and $d \in \mathbb{N}_0$. Then, $\chi_d^\Delta(G) \geq p^2/(pd + p^2 - 2q)$.* ■

Again this theorem holds with equality for a complete graph. Consider the complete graph $K_{n(d+1)}$ where $n \in \mathbb{N}$. Then $\chi(K_{n(d+1)}) = n(d+1)$ and $\chi_d^\Delta(K_{n(d+1)}) = n$. Also, $p(K_{n(d+1)}) = n(d+1)$ and $q(K_{n(d+1)}) = \binom{n(d+1)}{2}$, so that the right-hand side of the inequality in Theorem 3.5 becomes

$$\frac{n^2(d+1)^2}{n(d+1)d + n^2(d+1)^2 - 2\frac{n(d+1)[n(d+1)-1]}{2}} = n.$$

In a second paper Andrews and Jacobson [4] provided a lower bound on the $\Delta(d)$ -chromatic number of a graph G related to the second lower bound in Theorem 2.8 on the classical chromatic number, namely $\chi(G) \geq p(G)/\beta(G)$, where $\beta(G)$ is the independence number of G as before. In this regard, a subset $S \subseteq V(G)$ of a graph G , is said to be d -**dependent** if $\Delta(\langle S \rangle) \leq d$, for some $d \in \mathbb{N}_0$. Then the d -**dependence number**, $\beta_d(G)$, of a graph G , is the maximum cardinality of a maximal d -dependent set of vertices of G [4, §1]. Note that if $d = 0$, then the subset S is an independent set and $\beta_0(G) = \beta(G)$. The lower bound on the $\Delta(d)$ -chromatic number of a graph G by Andrews and Jacobson [4] may now be stated.

Theorem 3.6 *Let G be a graph and $d \in \mathbb{N}_0$. Then, $\chi_d^\Delta(G) \geq p(G)/\beta_d(G)$.* ■

Cowen *et al.* [34, Theorem 3.6] in their turn proved the following upper bound on the $\Delta(d)$ -chromatic number of a graph G in terms of d and the size of G alone.

Theorem 3.7 *Let G be a graph and $d \in \mathbb{N}_0$. Then, $\chi_d^\Delta(G) \leq \lfloor \sqrt{2q(G)}/(d+1) + 2 \rfloor$.* ■

A result of Lovász [79, Theorem 2] from the 1960s, given here as Theorem 3.8, which has been rediscovered since then, is central in establishing an upper bound on $\chi_d^\Delta(G)$ for a graph G in terms of d and $\Delta(G)$.

²The parameter χ_d^Δ is often also sometimes called the **defective chromatic number** in the literature.

Theorem 3.8 (Lovász's Theorem) *Let G be a graph and $d_1, \dots, d_x \in \mathbb{N}_0$ such that $\sum_{i=1}^x d_i \geq \Delta(G) - x + 1$. Then $V(G)$ may be partitioned into x sets $\mathcal{C}_1, \dots, \mathcal{C}_x$ such that $\Delta(\langle \mathcal{C}_i \rangle) \leq d_i$ for each $i \in \{1, \dots, x\}$ [112, p 203].*

Proof: The proof is by induction over the parameter x . Clearly the statement is true for $x = 1$. Suppose $x = 2$ and let $d_1, d_2 \in \mathbb{N}_0$ such that $d_1 + d_2 \geq \Delta(G) - 1$. If $d_1 = d_2 = 0$, then $\Delta(G) \leq 1$ implying that G consists of end-vertices and isolated vertices only, in which case the theorem is true.

Suppose now that $d_1 + d_2 \geq 1$. For an arbitrary partition of $V(G)$ into two sets V_1 and V_2 , consider the expression $\Phi(V_1, V_2) = d_1[2q(\langle V_2 \rangle) - p(\langle V_2 \rangle)] + d_2[2q(\langle V_1 \rangle) - p(\langle V_1 \rangle)]$. Let $\mathcal{C}_1, \mathcal{C}_2$ be a partition of $V(G)$ that minimizes $\Phi(\mathcal{C}_1, \mathcal{C}_2)$. To show that $\Delta(\langle \mathcal{C}_1 \rangle) \leq d_1$ and $\Delta(\langle \mathcal{C}_2 \rangle) \leq d_2$, let v be any vertex of \mathcal{C}_1 and let v be adjacent to a and b vertices of $\langle \mathcal{C}_1 \rangle$ and $\langle \mathcal{C}_2 \rangle$, respectively. Then $\deg_G(v) = a + b$ and therefore $a + b \leq \Delta(G)$. Let $W_1 = \mathcal{C}_1 \setminus \{v\}$ and $W_2 = \mathcal{C}_2 \cup \{v\}$. Then

$$\begin{aligned} \Phi(W_1, W_2) &= d_1[2(q(\langle \mathcal{C}_2 \rangle) + b) - (p(\langle \mathcal{C}_2 \rangle) + 1)] + d_2[2(q(\langle \mathcal{C}_1 \rangle) - a) - (p(\langle \mathcal{C}_1 \rangle) - 1)] \\ &= \Phi(\mathcal{C}_1, \mathcal{C}_2) + [d_1(2b - 1) - d_2(2a - 1)], \end{aligned}$$

and because of the minimality of $\Phi(\mathcal{C}_1, \mathcal{C}_2)$, it follows that $d_1(2b - 1) - d_2(2a - 1) \geq 0$. Using $a + b \leq \Delta(G) \leq d_1 + d_2 + 1$ to eliminate b , it follows that $2a(d_1 + d_2) \leq 2d_1(d_1 + d_2) + (d_1 + d_2)$. Since $d_1 + d_2 \geq 0$ it is admissible to divide the latter expression by $2(d_1 + d_2)$, to obtain $a \leq d_1 + 1/2$. Thus, $a \leq d_1$ since $a, d_1 \in \mathbb{N}$. Since v was any vertex in \mathcal{C}_1 , it follows that $\Delta(\langle \mathcal{C}_1 \rangle) \leq d_1$. By symmetry it follows that $\Delta(\langle \mathcal{C}_2 \rangle) \leq d_2$. Hence, the theorem is true for $x = 2$.

Now suppose the theorem is true for $x - 1$, where $x > 2$. Let $d'_1 = d_1$ and $d'_2 = \sum_{i=2}^x d_i + x - 2$. Then $d'_1 + d'_2 \geq \Delta(G) - 1$ and because the theorem is true for $x = 2$, there exists a partition $\mathcal{C}'_1, \mathcal{C}'_2$ of $V(G)$ such that $\Delta(\langle \mathcal{C}'_1 \rangle) \leq d'_1$ and $\Delta(\langle \mathcal{C}'_2 \rangle) \leq d'_2$. Therefore, for the graph $\langle \mathcal{C}'_2 \rangle$ and $d_2, \dots, d_x \in \mathbb{N}_0$, it follows that $\sum_{i=2}^x d_i \geq \Delta(\langle \mathcal{C}'_2 \rangle) - (x - 1) + 1$. Hence, from the induction hypothesis there exists a partition $\mathcal{C}_2, \dots, \mathcal{C}_x$ of \mathcal{C}'_2 such that $\Delta(\langle \mathcal{C}_i \rangle) \leq d_i$ for each $i \in \{2, \dots, x\}$. Consequently, the partition $\mathcal{C}'_1, \mathcal{C}_2, \dots, \mathcal{C}_x$ of $V(G)$ satisfies the requirements of the theorem [22, pp 31–33]. ■

The result of Theorem 3.8 is illustrated by means of an example.

Example 3.1 *For the circulant $C_{12}\langle 1, 2, 4 \rangle$ it holds that $\Delta(C_{12}\langle 1, 2, 4 \rangle) = 6$. Suppose $d_i = i$ for $i = 1, 2, 3$. Then $\sum_{i=1}^3 d_i = 6 > \Delta(C_{12}\langle 1, 2, 4 \rangle) - 3 + 1 = 4$. Thus, by Theorem 3.8, there exists a partition of $V(C_{12}\langle 1, 2, 4 \rangle)$ into three sets $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ such that $\Delta(\langle \mathcal{C}_i \rangle) \leq i$ for each $i = 1, 2, 3$. One such a partition is shown in Figure 3.1. ■*

By taking $d \in \mathbb{N}$ and $d_i = d - 1$ in Theorem 3.8 for all $i \in \{1, \dots, x\}$ the following corollary is obtained.

Corollary 3.1 *Let G be a graph and $d, x \in \mathbb{N}$ such that $xd > \Delta(G)$. Then $V(G)$ may be partitioned into x sets $\mathcal{C}_1, \dots, \mathcal{C}_x$ such that $\Delta(\langle \mathcal{C}_i \rangle) \leq d - 1$ for each $i \in \{1, \dots, x\}$, i.e. there exists a $\Delta(d - 1, x)$ -colouring of G . ■*

In terms of the maximum degree chromatic number, Corollary 3.1 states that if $d, x \in \mathbb{N}$ such that $xd > \Delta(G)$ for some graph G , then $\chi_{d-1}^\Delta(G) \leq x$. This, in turn, leads to an upper bound on the maximum degree chromatic number, $\chi_d^\Delta(G)$, given in the next corollary [46, Corollary 4.3].

Corollary 3.2 *Let G be a graph. Then $\chi_d^\Delta(G) \leq \lceil (\Delta(G) + 1)/(d + 1) \rceil$ for all $d \in \mathbb{N}_0$. ■*

The bounds in Corollaries 2.2 and 3.2 as well as Theorems 2.16 and 3.4–3.7 may be summarised as

$$\max \left\{ \frac{\chi(G)}{d+1}, \frac{\omega(G)}{d+1}, \frac{p(G)}{\beta_d(G)}, \frac{[p(G)]^2}{p(G)d + [p(G)]^2 - 2q(G)} \right\} \leq \chi_d^\Delta(G) \quad (3.2.1)$$

and

$$\chi_d^\Delta(G) \leq \min \left\{ \left\lfloor \frac{\sqrt{2q(G)}}{d+1} + 2 \right\rfloor, \left\lfloor \frac{p(G)}{d+1} \right\rfloor, \left\lfloor \frac{\Delta(G) + 1}{d+1} \right\rfloor \right\}. \quad (3.2.2)$$

Theorem 3.11 *Suppose G is a critical $\Delta(d, x)$ -chromatic graph with $x \in \mathbb{N} \setminus \{1\}$ and $d \in \mathbb{N}_0$. Then $\delta(G) \geq x - 1$. \blacksquare*

Cowen *et al.* [33, §3] focussed on graphs embedded on various surfaces and among others gave a characterization of all $d \in \mathbb{N}_0$ and $x \in \mathbb{N}$ for which every planar graph is $\Delta(d, x)$ -colourable. Their results are summarised in the following theorem.

Theorem 3.12

- (a) *There exists a planar graph that is not $\Delta(d, 2)$ -colourable for each $d \in \mathbb{N}$.*
- (b) *There exists a planar graph that is not $\Delta(1, 3)$ -colourable.*
- (c) *Every planar graph is $\Delta(2, 3)$ -colourable.*
- (d) *Every planar graph is $\Delta(1, 4)$ -colourable.*

Let $d \in \mathbb{N}$. Then, for each $d \in \mathbb{N}$, the non- $\Delta(d, 2)$ -colourable planar graph mentioned in Theorem 3.12(a) and constructed in the proof by Cowen *et al.* [33, Theorem 3], is the graph G formed by the join of a single vertex v with the union of $d + 1$ disjoint copies of the $(d + 2)$ -star $K_{1, d+1}$. The non- $\Delta(1, 3)$ -colourable planar graph in Theorem 3.12(b) is given in Figure 3.2. Let G be a planar graph. Then from the Four-colour Theorem (Theorem 2.7) it is already known that if $d \geq 0$ then $\chi_d^\Delta(G) \leq 4$. Furthermore, from Theorem 3.12(d), which was proved without using the Four-colour Theorem, it follows that if $d \geq 1$ then $\chi_d^\Delta(G) \leq 4$ and the construction in Theorem 3.12(b) shows that this bound is sharp when $d = 1$. Finally, from Theorem 3.12(c) it follows that if $d \geq 2$ then $\chi_d^\Delta(G) \leq 3$ and the construction in Theorem 3.12(a) shows that this bound is sharp for every $d \geq 2$.

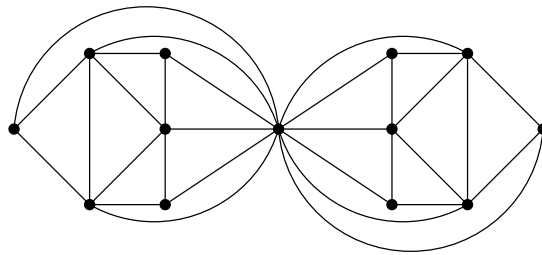


FIGURE 3.2: The non- $\Delta(1, 3)$ -colourable planar graph constructed by Cowen *et al.* [33].

Goddard [51, Theorem 1] further explored the result in Theorem 3.12(c) and proved that every planar graph is $\Delta(2, 3)$ -colourable such that each colour class induced subgraph is a disjoint union of paths. This result gave rise to a new kind of $\Delta(2, x)$ -colouring, namely a generalized colouring in which the colouring rule \mathcal{R} is the requirement that each colour class induced subgraph is a disjoint union of paths (of order at most some specified $k \in \mathbb{N}$)³.

The interested reader is also referred to a number of papers providing certain results on maximum degree colourings which fall beyond the scope of this dissertation. These papers include the second paper by Andrews and Jacobson [4] in which the relation between the parameter χ_d^Δ and Ramsey theory is explored. Frick and Henning [46, §2] also studied uniquely $\Delta(d, x)$ -colourable graphs.⁴ Cowen *et al.* [33, §2 & 4] also gave a characterization of all $d \in \mathbb{N}_0$ and $x \in \mathbb{N}$ for which every *outerplanar* graph is $\Delta(d, x)$ -colourable, as well as the smallest value of d for which a graph embedded on a connected surface is $\Delta(d, 4)$ -colourable. The latter result was then improved by Archdeacon [6] who determined the smallest value of d for which a graph embedded on a connected surface is $\Delta(d, 3)$ -colourable. Woodall [114] considered the question whether any graph G without certain subcontractions is $\Delta(d, x)$ -colourable.

The λ -chromatic sequence of a graph G (for some parameter λ) was introduced in §2.4.4. This section is now concluded with a discussion on the Δ -chromatic sequence of a graph. The following result by Frick and Henning [46] provides necessary conditions for a sequence of positive integers to be the Δ -chromatic sequence of some graph G .

³Unfortunately, this generalized colouring also become known in the literature as **path colourings** (see, for example, [38, 56]).

⁴A graph G is said to be uniquely $\Delta(d, x)$ -colourable if every $\Delta(d, x)$ -colouring of G produces the same colour classes.

Theorem 3.13 *If x_0, x_1, x_2, \dots is the Δ -chromatic sequence of some graph G , then*

- (i) $x_n = 1$ for some integer n , and
- (ii) $x_j \leq x_i \leq x_j \lceil (j+1)/(i+1) \rceil$ if $0 \leq i < j$.

Proof: For any graph G it follows that $\chi_d^\Delta(G) = 1$ if $d \geq \Delta(G)$. Thus, (i) is satisfied with $n = \Delta(G)$.

Since any $\Delta(d, x)$ -colouring of a graph G is also a $\Delta(d+1, x)$ -colouring of G , the first inequality in (ii) is satisfied. Let $\mathcal{C}_1, \dots, \mathcal{C}_{x_j}$ be the colour classes of a $\Delta(j, x_j)$ -colouring of a graph G . Then $\Delta(\langle \mathcal{C}_\ell \rangle) \leq j$ for all $\ell \in \{1, \dots, x_j\}$. Hence it follows by Corollary 3.2 that $\chi_i^\Delta(\langle \mathcal{C}_\ell \rangle) \leq \lceil (j+1)/(i+1) \rceil$ for all $\ell \in \{1, \dots, x_j\}$. Since there are x_j disjoint subsets of vertices $\mathcal{C}_1, \dots, \mathcal{C}_{x_j}$ coloured in this manner for a $\Delta(i, x_i)$ -colouring of G , it follows that $x_i = \chi_i^\Delta(G) \leq x_j \lceil (j+1)/(i+1) \rceil$ [22, p 40]. ■

Burger and Grobler [22, pp 42–48] constructed a table containing a list of sequences (x_i) that satisfy the conditions in Theorem 3.13 and have at most 12 terms greater than 1. (In their table only the first one of the infinite tail of ones is listed.) This table is reproduced as Table C.1 in Appendix C. They claim in [22, p 41] that every sequence with at most 12 terms greater than 1 that satisfy the conditions in Theorem 3.13 either appears in their list or may be obtained from a sequence in the list by increasing the number of twos in the sequence. The validity of their claim follows from the following result.

Proposition 3.1 *For any sequence (x_i) in Table C.1, the sequence obtained by increasing the number of twos in (x_i) , also satisfies the conditions in Theorem 3.13.*

Proof: Consider the sequence $(x_i) = x_0 x_1 x_2 \dots x_{n-1} 1 1 \dots$ with $x_{n-1} = 2$, satisfying the conditions in Theorem 3.13. Hence, $x_i \leq \lceil (n+1)/(i+1) \rceil$ if $0 \leq i < n$. Adding another two into the sequence leads to the sequence $(x'_i) = x_0 x_1 x_2 \dots x_{n-1} x'_n x'_{n+1} 1 1 \dots$, where $x'_n = 2$ and $x'_{n+1} = 1$. Since $x_i \leq \lceil (n+1)/(i+1) \rceil$ if $0 \leq i < n$, it follows that $x_i \leq x'_n \lceil (n+1)/(i+1) \rceil = 2 \lceil (n+1)/(i+1) \rceil$ if $0 \leq i < n$. Furthermore, $x_i \leq x'_{n+1} \lceil ((n+1)+1)/(i+1) \rceil = \lceil (n+2)/(i+1) \rceil$ if $0 \leq i < n$. Finally, $x'_n = 2 \leq x'_{n+1} \lceil (n+2)/(n+1) \rceil$.

Since $x_i \geq 2$ for all $i = 1, \dots, n-1$, it follows that $x_i \geq x'_n = 2$ if $0 \leq i < n$ and $x_i \geq x'_{n+1} = 1$ if $0 \leq i < n$. Finally, $x'_n \geq x'_{n+1}$. The sequence (x'_i) therefore satisfies the conditions of Theorem 3.13. ■

For most of the sequences in Table C.1 one may obtain a graph G for which the specific sequence is the Δ -chromatic sequence. These graphs are also listed in Table C.1 [22]. Furthermore, if a graph for a specific sequence (x_i) in Table C.1 is known, then a graph may also be constructed with the Δ -chromatic sequence (x_i) into which a number of twos has been inserted.

Proposition 3.2 *Let G be a graph with Δ -chromatic sequence (x_i) . Then the graph constructed from G by joining m isolated vertices to a vertex v_i of G for which $\deg_G(v_i) = \Delta(G)$, has the Δ -chromatic sequence obtained from (x_i) by inserting m twos into (x_i) .*

Proof: Let G be a graph of order n with $k = \Delta(G)$ and let v_i be a vertex of G for which $\deg_G(v_i) = \Delta(G)$. Let the Δ -chromatic sequence of G be the sequence $(x_i) = x_0 x_1 x_2 \dots x_{k-1} 1 1 \dots$ with $x_{k-1} = 2$. Construct the graph $G'(G, m)$ by joining m isolated vertices, $v_{n+1}, v_{n+2}, \dots, v_{n+m}$, to the vertex v_i of G as illustrated in Figure 3.3. Say colour 1 was assigned to vertex v_i in a χ_d^Δ -colouring of G in x colours, then for $0 \leq d \leq (k-1)$ a χ_d^Δ -colouring of G' utilising x colours may be obtained by colouring the vertices v_1, v_2, \dots, v_n the same as in the χ_d^Δ -colouring of G and by assigning any of the x colours, other than colour 1, to the vertices $v_{n+1}, v_{n+2}, \dots, v_{n+m}$. The first k terms of the Δ -chromatic sequence of G' , therefore, are the same as those of G .

In the Δ -chromatic sequence of G , $x_k = 1$, but one colour is insufficient for a χ_k^Δ -colouring of G' , since $\deg_{G'}(v_i) = k + m$. However, a χ_d^Δ -colouring of G' utilising two colours may be obtained by colouring the vertices v_1, v_2, \dots, v_n with one colour (*i.e.* the same as in the χ_d^Δ -colouring of G) and the vertices $v_{n+1}, v_{n+2}, \dots, v_{n+m}$ with a second colour, for all $k \leq d \leq (k+m-1)$. If $d = k+m$, then $\chi_d^\Delta(G') = 1$. The Δ -chromatic sequence of G' therefore has m more twos than the Δ -chromatic sequence of G . ■

The properties of Δ -chromatic sequences are very different from those of ω -chromatic sequences. For example, Theorem 2.30 shows that the length of a constant subsequence $x_i = x_{i+1} = \dots = x_j \geq 2$ of an

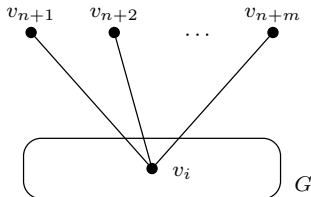


FIGURE 3.3: The graph construction $G'(G, m)$ where m isolated vertices are joined with a vertex in G of maximum degree.

ω -chromatic sequence is restricted [22], whereas Proposition 3.2 indicates that at least a subsequence of arbitrary length of twos can occur in a Δ -chromatic sequence. Furthermore, Theorem 2.31 implies that the difference between any two consecutive terms x_i and x_{i+1} with $i \geq 2$ in an ω -chromatic sequence can be made arbitrarily large, but this is not the case for Δ -chromatic sequences. In particular, it follows by condition (ii) of Theorem 3.13 that $x_i \leq 2x_{i+1}$ for $i \geq 0$ [22]. A similar result holds for τ -chromatic sequences.

The necessary conditions in Theorem 3.13 are also sufficient for sequences with at most 11 terms greater than 1 to be the Δ -chromatic sequence of some graph. However, as evident from the open spaces in the graph-column in Table C.1, it is unknown whether Theorem 3.13 also provides sufficient conditions for a sequence of more than 11 positive integers greater than 1 to be the Δ -chromatic sequence of some graph G , or whether other sufficient conditions must be found. This is the topic of Chapter 6 of this dissertation.

3.3 The parameters χ_d^Δ and D_x^Δ

The problem of finding the $\Delta(d)$ -chromatic number of a graph G may, in a sense, be inverted by rather fixing the number of colours, x , that may be used, and then seeking to minimize the maximum colour class induced maximum degree (*i.e.* seeking the smallest value of d for which there exists a $\Delta(d, x)$ -colouring of G , for some fixed value of x). Denote the answer to this problem by $D_x^\Delta(G)$ and call such an optimal colouring a D_x^Δ -colouring of G . Note that a χ_d^Δ -colouring of G is not necessarily a D_x^Δ -colouring of G . It is possible to establish the following relationship between $\chi_d^\Delta(G)$ and $D_x^\Delta(G)$, by noting that if there exists a $\Delta(d, x)$ -colouring of G , for some fixed values $d, x \in \mathbb{N}$, then the inequalities $\chi_d^\Delta(G) \leq x$ and $D_x^\Delta(G) \leq d$ both immediately follow.

Proposition 3.3 For any graph G ,

- (a) $\chi_d^\Delta(G) \leq x$ if and only if $D_x^\Delta(G) \leq d$.
- (b) $\chi_d^\Delta(G) = x$ if and only if $D_x^\Delta(G) \leq d$ and $D_{x-1}^\Delta(G) > d$.
- (c) $D_x^\Delta(G) = d$ if and only if $\chi_d^\Delta(G) \leq x$ and $\chi_{d-1}^\Delta(G) > x$.

Proof:

(a) Suppose $\chi_d^\Delta(G) \leq x$ for some graph G . Then there exists a $\Delta(d, x)$ -colouring of G , implying that $D_x^\Delta(G) \leq d$. Conversely, suppose $D_x^\Delta(G) \leq d$. Then again there exists a $\Delta(d, x)$ -colouring of G , implying that $\chi_d^\Delta(G) \leq x$.

(b) Suppose $\chi_d^\Delta(G) = x$ for some graph G . Let $D_x^\Delta(G) > d$ or $D_{x-1}^\Delta(G) \leq d$. From (a) and its contra-positive it follows that $\chi_d^\Delta(G) \leq x-1$ or $\chi_d^\Delta(G) > x$. Conversely, suppose $D_x^\Delta(G) \leq d$ and $D_{x-1}^\Delta(G) > d$. Let $\chi_d^\Delta(G) > x$ or $\chi_d^\Delta(G) \leq x-1$. Then again from (a) and its contra-positive, $D_{x-1}^\Delta(G) \leq d$ or $D_x^\Delta(G) > d$.

(c) This result follows from (a) and its contra-positive in a manner similar to the proof of (b). ■

Recall the growth property in Theorem 3.1 for the $\Delta(d)$ -chromatic number of a graph G . The related inversion number, $D_x^\Delta(G)$, satisfies a similar growth property, as stated below.

Proposition 3.4 (Growth property) For any graph G and any $x \in \mathbb{N}$,

$$0 \leq D_{x+1}^\Delta(G) \leq D_x^\Delta(G) \leq \Delta(G). \quad (3.3.1)$$

Both outer bounds are attainable, for all graphs.

Proof: The upper bound is trivial and is attained when $x = 1$. The lower bound is also trivial and is the minimum value of the maximum degree of a graph. Let $D_x^\Delta(G) = d^*$. Then, for any $\Delta(d^*, x)$ -colouring of a graph G , there exists a $\Delta(d^*, x+1)$ -colouring of G . Hence $D_{x+1}^\Delta(G) \leq d^*$. ■

Bounds on the $\Delta(d)$ -chromatic number, $\chi_d^\Delta(G)$, of a graph G were summarised in (3.2.1) and (3.2.2). Using Proposition 3.3(a) these bounds may be converted to the bounds

$$\max \left\{ \frac{\chi(G) - x}{x}, \frac{\omega(G) - x}{x}, \frac{[p(G)]^2(1-x) + 2xq(G)}{xp(G)} \right\} \leq D_x^\Delta(G) \quad (3.3.2)$$

and

$$D_x^\Delta(G) \leq \min \left\{ \left\lfloor \frac{\sqrt{2q(G)}}{x-1} - 1 \right\rfloor, \left\lceil \frac{p(G) - x}{x} \right\rceil, \left\lceil \frac{\Delta(G) + 1 - x}{x} \right\rceil \right\} \quad (3.3.3)$$

on the parameter $D_x^\Delta(G)$.

Finally, consider the Δ -chromatic sequence (x_i) of a graph G . The sequence (x_i) often has a large amount of repetition and it therefore makes sense to document at which values of i the sequence values change and by how much, rather than to record all the repeated values. The value of the parameter $D_x^\Delta(G)$ gives this *position* in the Δ -chromatic sequence where the number of colours x appears for the first time as illustrated in Example 3.2.

Example 3.2 The Δ -chromatic sequence of the graph $C_{12}\langle 1, 2, 4 \rangle$ in Example 3.1 is given below.

$$\begin{array}{cccccccc} x_0 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & \dots \\ 3 & 3 & 3 & 2 & 2 & 2 & 1 & 1 & \dots \end{array}$$

From the sequence above it is clear that $\chi_0^\Delta(C_{12}\langle 1, 2, 4 \rangle) = \chi_1^\Delta(C_{12}\langle 1, 2, 4 \rangle) = \chi_2^\Delta(C_{12}\langle 1, 2, 4 \rangle) = 3$, $\chi_3^\Delta(C_{12}\langle 1, 2, 4 \rangle) = \chi_4^\Delta(C_{12}\langle 1, 2, 4 \rangle) = \chi_5^\Delta(C_{12}\langle 1, 2, 4 \rangle) = 2$ and $\chi_i^\Delta(C_{12}\langle 1, 2, 4 \rangle) = 1$ for all $i \geq 6$. The sequence above may also be captured by only documenting the three values $D_3^\Delta(C_{12}\langle 1, 2, 4 \rangle) = 0$, $D_2^\Delta(C_{12}\langle 1, 2, 4 \rangle) = 3$ and $D_1^\Delta(C_{12}\langle 1, 2, 4 \rangle) = 6$. ■

3.4 Chapter Summary

In this chapter the notion of maximum degree colourings was introduced and bounds on the maximum degree chromatic number found in the literature were reviewed (in §3.1 and §3.2 respectively). Some basic results in the literature on Δ -chromatic sequences were also provided in §3.2. In §3.3 an inverted strategy towards determining maximum degree colourings was explained, and bounds were established on the new inverted chromatic parameter $D_x^\Delta(G)$. The section closed with the relation between the parameters $\chi_d^\Delta(G)$ and $D_x^\Delta(G)$ and the significance of the parameter $D_x^\Delta(G)$ when documenting the Δ -chromatic sequence of a graph G .

Chapter 4

Simple graph structure classes

“Science may set limits to knowledge, but should not set limits to imagination.”

Bertrand Russell (1872–1970)

“Mathematics, rightly viewed, possess not only truth, but supreme beauty — a beauty cold and austere, like that of sculpture.”

Bertrand Russell (1872–1970)

Exact values for the maximum degree chromatic number may be established for graphs from various structure classes. In the first part of this chapter exact values for the maximum degree chromatic number for some of these structure classes are established, including numbers for bipartite graphs (§4.1), cycles and wheels (§4.2), complete graphs (§4.3), products of paths and cycles and some numbers for products of complete graphs (§4.4) and some circulants (§4.5). In the second part of the chapter upper and lower bounds for the maximum degree chromatic number are established for graph classes for which the maximum degree chromatic number could not be obtained exactly, including some numbers for products of complete graphs (§4.4) and complete balanced multipartite graphs (§4.6).

4.1 Bipartite Graphs

The values for the maximum degree chromatic number of a bipartite graph may be obtained easily as stated in the proof of the following proposition.

Proposition 4.1

Suppose $d \in \mathbb{N}_0$. Let G be a bipartite graph, then

$$\chi_d^\Delta(G) = \begin{cases} 1 & \text{if } d \geq \Delta(G), \\ 2 & \text{otherwise.} \end{cases}$$

Proof: Since a graph is (properly) 2-colourable if it is bipartite, only two colours are necessary if $d \in \{0, \dots, \Delta(G) - 1\}$. If $d \geq \Delta(G)$, all vertices of G may be coloured with one colour. ■

The values for the related inversion number, $D_x^\Delta(G)$, as determined directly from Proposition 3.3 are given below.

Corollary 4.1 Suppose $x \in \mathbb{N}$. Then, for the bipartite graph G ,

$$D_x^\Delta(G) = \begin{cases} 0 & \text{if } x = 2, \\ \Delta(G) & \text{if } x = 1. \end{cases} \quad \blacksquare$$

Because trees are bipartite graphs, the following results immediately follow from Proposition 4.1 and Corollary 4.1 via Proposition 3.3.

Corollary 4.2 *Suppose $n, x \in \mathbb{N}$ and $d \in \mathbb{N}_0$. Then*

$$\chi_d^\Delta(T_n) = \begin{cases} 1 & \text{if } d \geq \Delta(T_n), \\ 2 & \text{if } d < \Delta(T_n), \end{cases} \quad \text{and} \quad D_x^\Delta(T_n) = \begin{cases} 0 & \text{if } x = 2, \\ \Delta(T_n) & \text{if } x = 1, \end{cases}$$

where T_n denotes a tree of order n . ■

Therefore, although determining values of graph parameters for trees are sometimes intricate, the maximum degree chromatic number for a tree of order n is very simple to determine. If $d < \Delta(T_n)$, it is also very simple to determine a χ_d^Δ -colouring of a tree in 2 colours. In particular, root the tree and colour the root of the tree as well as all vertices at even distances from the root with one colour and colour all vertices at odd distances from the root with a second colour as demonstrated in Figure 4.1.

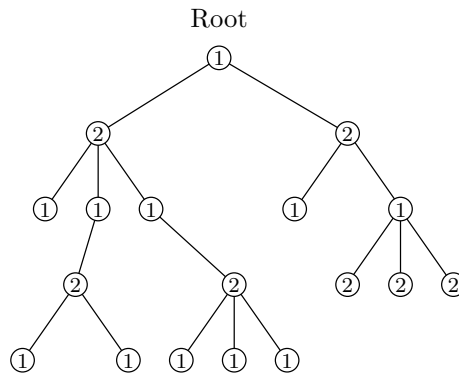


FIGURE 4.1: An illustration of a χ_d^Δ -colouring for the tree T_{18} in 2 colours, where $d < \Delta(T_{18}) = 4$.

Because paths and stars are trees, the following results immediately follow from Corollary 4.2.

Corollary 4.3 *Suppose $n, x \in \mathbb{N}$ and $d \in \mathbb{N}_0$. Then*

$$\chi_d^\Delta(P_n) = \begin{cases} 1 & \text{if } d \geq 2, \\ 2 & \text{if } d < 2, \end{cases} \quad \text{and} \quad D_x^\Delta(P_n) = \begin{cases} 0 & \text{if } x = 2, \\ 2 & \text{if } x = 1, \end{cases}$$

where P_n denotes a path of order n . ■

Corollary 4.4 *Suppose $n, x \in \mathbb{N}$ and $d \in \mathbb{N}_0$. Then*

$$\chi_d^\Delta(S_n) = \begin{cases} 1 & \text{if } d \geq n - 1, \\ 2 & \text{if } d < n - 1, \end{cases} \quad \text{and} \quad D_x^\Delta(S_n) = \begin{cases} 0 & \text{if } x = 2, \\ n - 1 & \text{if } x = 1, \end{cases}$$

where S_n denotes a star of order n . ■

4.2 Cycles and Wheels

As expected, the values for the maximum degree chromatic number for a cycle of order n are as intuitive as those for a tree of order n .

Proposition 4.2 *Suppose $d \in \mathbb{N}_0$ and $n \in \mathbb{N}$. Let C_n denote a cycle of order n . Then*

$$\chi_d^\Delta(C_n) = \begin{cases} 1 & \text{if } d \geq 2, \\ 2 & \text{if } d = 1 \text{ or if } (d = 0 \text{ and } n \text{ is even}), \\ 3 & \text{if } d = 0 \text{ and } n \text{ is odd.} \end{cases}$$

Proof: For each vertex v in the cycle C_n , $\deg_{C_n}(v) = 2$. Thus, if $d \geq 2$ then $\chi_d^\Delta(C_n) = 1$, because in this case one colour may be used to colour all the vertices of C_n .

If $d = 1$, then one colour is insufficient for a χ_d^Δ -colouring of C_n , since $d < \Delta(C_n) = 2$. However, if $d = 1$ a colouring of $C_n : v_1v_2 \dots v_nv_1$ using two colours, may be obtained by colouring all odd indexed vertices with one colour and all even indexed vertices with another colour.

If $d = 0$ and n is even, then the same colouring strategy as for the case where $d = 1$ yields an optimal χ_0^Δ -colouring of C_n . Finally, if $d = 0$ and n is odd, then two colours are insufficient for a χ_0^Δ -colouring of C_n , because at least two adjacent vertices will receive the same colour if only two colours are used, resulting in a colour induced maximum degree of at least $1 > d$. However, a χ_0^Δ -colouring of $C_n : v_1v_2 \dots v_nv_1$ using three colours may be obtained by colouring all odd indexed vertices (except v_n) with one colour and all even indexed vertices with a second colour. The vertex v_n may finally be coloured with the third colour. ■

An example of a χ_1^Δ -colouring of a cycle is given in Figure 4.2(a). For the case where $d = 0$ the number of colours necessary for a χ_0^Δ -colouring of a cycle C_n is illustrated in Figure 4.2(b) [and (c), respectively] where the cycle has an even number of vertices [an odd number of vertices, respectively].

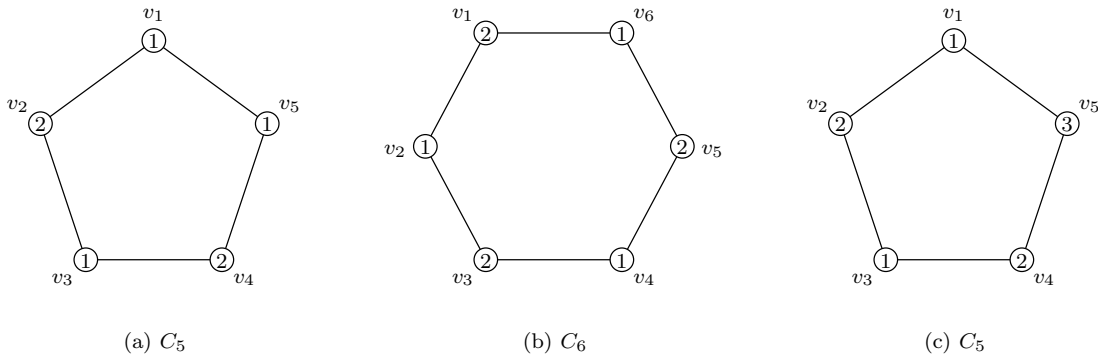


FIGURE 4.2: Examples of the colouring strategy in the proof of Proposition 4.2 where (a) $d = 1$, (b) $d = 0$ and n is even, and (c) $d = 0$ and n is odd.

The values for the related inversion number, $D_x^\Delta(C_n)$, follows directly from Proposition 4.2 via Proposition 3.3 as stated below.

Corollary 4.5 Suppose $x, n \in \mathbb{N}$. Then

$$D_x^\Delta(C_n) = \begin{cases} 0 & \text{if } (x = 3 \text{ and } n \text{ is odd}) \text{ or if } (x = 2 \text{ and } n \text{ is even}), \\ 1 & \text{if } x = 2 \text{ and } n \text{ is odd}, \\ 2 & \text{if } x = 1, \end{cases}$$

where C_n denotes a cycle of order n . ■

Using the arguments in the proof of Proposition 4.2 for the maximum degree chromatic number for a cycle of order n , the maximum degree chromatic number for a wheel of order n may also be determined.

Proposition 4.3 Suppose $d \in \mathbb{N}_0$ and $n \in \mathbb{N}$. Let W_n denote a wheel of order n . Then

$$\chi_d^\Delta(W_n) = \begin{cases} 1 & \text{if } d \geq n - 1, \\ 2 & \text{if } 2 \leq d < n - 1, \\ 3 & \text{if } d = 1 \text{ or if } (d = 0 \text{ and } n - 1 \text{ is even}), \\ 4 & \text{if } d = 0 \text{ and } n - 1 \text{ is odd.} \end{cases}$$

Proof: Since $\Delta(W_n) = n - 1$, it is clear that $\chi_d^\Delta(W_n) = 1$ if $d \geq n - 1$, because in this case one colour may be used to colour all the vertices of W_n .

If $d < n - 1$, then one colour is insufficient for a χ_d^Δ -colouring of W_n , since $d < \Delta(W_n) = n - 1$. However, if $2 \leq d < n - 1$ a colouring of $W_n = C_{n-1} + \langle v_n \rangle$ using two colours, may be obtained by colouring C_{n-1} with one colour since $\Delta(C_{n-1}) = 2$, and by using a second colour for v_n .

If $d = 1$, then two colours are insufficient for a χ_d^Δ -colouring of W_n , because in $W_n = C_{n-1} + \langle v_n \rangle$ the vertex v_n will receive the same colour as at least two other vertices from C_{n-1} if only two colours are used, resulting in a colour induced maximum degree of at least $2 > d$. However, if $d = 1$ a χ_1^Δ -colouring of W_n using three colours, may be obtained by colouring C_{n-1} according to the strategy in Proposition 4.2, and by using a third colour for v_n .

If $d = 0$ and $n - 1$ is even, then the same colouring strategy as for the case where $d = 1$ yields an optimal χ_0^Δ -colouring of W_n . Finally, if $d = 0$ and $n - 1$ is odd, then three colours are insufficient for a χ_0^Δ -colouring of W_n , because at least two adjacent vertices will receive the same colour if only three colours are used, resulting in a colour induced maximum degree of at least $1 > d$. However, a χ_0^Δ -colouring of $W_n = C_{n-1} + \langle v_n \rangle$ using four colours may be obtained by colouring C_{n-1} according to the strategy in Proposition 4.2, and by using a fourth colour for v_n . ■

An example of a χ_d^Δ -colouring of a wheel for each of the cases where $2 \leq d < n - 1$, $d = 1$, and $d = 0$ and $n - 1$ is odd, are given in Figures 4.3(a), (b) and (c) respectively.

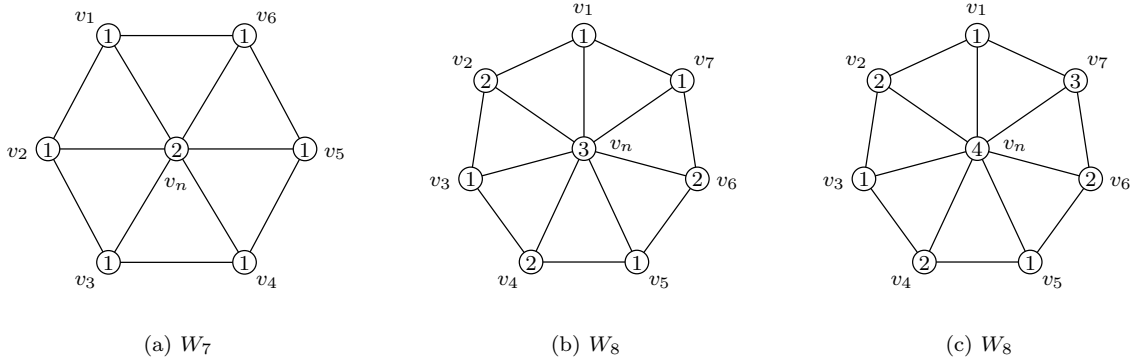


FIGURE 4.3: Examples of the colouring strategy in the proof of Proposition 4.3: (a) A χ_4^Δ -colouring of W_7 where $\chi_4^\Delta(W_7) = 2$. (b) A χ_1^Δ -colouring of W_8 where $\chi_1^\Delta(W_8) = 3$ and (c) A χ_0^Δ -colouring of W_8 where $\chi_0^\Delta(W_8) = 4$.

The values for the related inversion number, $D_x^\Delta(W_n)$, follows directly from Proposition 4.3 via Proposition 3.3 as stated below.

Corollary 4.6 Suppose $x, n \in \mathbb{N}$. Then

$$D_x^\Delta(W_n) = \begin{cases} 0 & \text{if } (x = 4 \text{ and } n - 1 \text{ is odd}) \text{ or if } (x = 3 \text{ and } n - 1 \text{ is even}), \\ 1 & \text{if } x = 3 \text{ and } n - 1 \text{ is odd,} \\ 2 & \text{if } x = 2, \\ n - 1 & \text{if } x = 1, \end{cases}$$

where W_n denotes a wheel of order n . ■

4.3 Complete Graphs

The maximum degree chromatic number for a complete graph, as stated below, follows directly from Theorem 2.15.

Proposition 4.4 Suppose $d \in \mathbb{N}_0$ and $n \in \mathbb{N}$. Let K_n denote a complete graph of order n . Then

$$\chi_d^\Delta(K_n) = \left\lceil \frac{n}{d+1} \right\rceil. \quad \blacksquare$$

Let $\mathcal{C}_1, \dots, \mathcal{C}_x$ denote the x colour classes of a χ_d^Δ -colouring of the complete graph K_n . Then it is clear that $\langle \mathcal{C}_i \rangle$ is a complete graph for all $i \in \{1, \dots, x\}$. Therefore the colouring strategy in Proposition 4.4 is to partition $V(K_n)$ arbitrarily into colour classes of cardinality $d+1$ where the last colour class formed may possibly have cardinality smaller than $d+1$. Three examples of χ_d^Δ -colourings for the complete graph K_8 are given in Figure 4.4. First, a χ_4^Δ -colouring of K_8 is illustrated in Figure 4.4(a). Only one complete subgraph of order 5 is formed during this colouring and the remaining three vertices are coloured with colour 2 (the last colour class formed). In a χ_2^Δ -colouring of K_8 , $\lceil 8/3 \rceil = 2$ complete subgraphs of order 3 may be formed and the remaining two vertices in K_8 are coloured with colour 3, as illustrated in Figure 4.4(b). Finally, in a χ_1^Δ -colouring of K_8 as illustrated in Figure 4.4(c); a total of $\lceil 8/2 \rceil = 4$ complete subgraphs of order 2 may be formed without any remaining vertices that still need to be coloured.

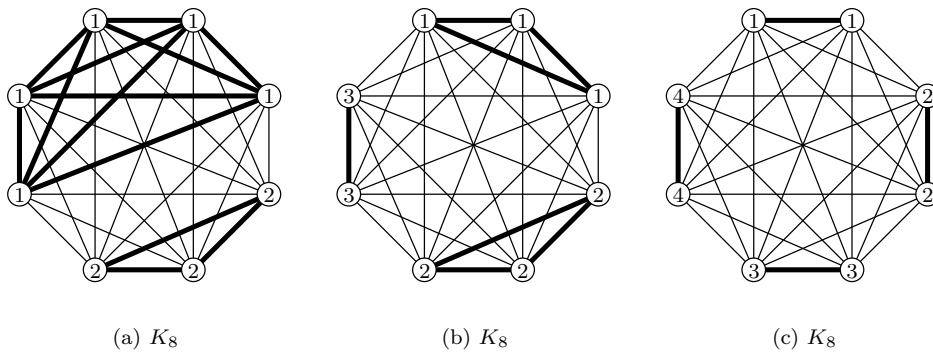


FIGURE 4.4: Examples of χ_d^Δ -colourings of K_8 . The bold edges indicate the complete colour-induced subgraphs formed during each χ_d^Δ -colouring of K_8 . (a) A χ_4^Δ -colouring of K_8 , showing that $\chi_4^\Delta(K_8) = 2$, (b) a χ_2^Δ -colouring of K_8 , showing that $\chi_2^\Delta(K_8) = 3$ and (c) a χ_1^Δ -colouring of K_8 , showing that $\chi_1^\Delta(K_8) = 4$.

The values for the related inversion number, $D_x^\Delta(K_n)$, as obtained via Proposition 3.3 is given below.

Corollary 4.7 Suppose $x, n \in \mathbb{N}$. Let K_n denote a complete graph of order n . Then

$$D_x^\Delta(K_n) = \left\lceil \frac{n-x}{x} \right\rceil. \quad \blacksquare$$

4.4 Products of Paths, Cycles and Complete Graphs

In this section the maximum degree chromatic number for cartesian products of paths, cycles and complete graphs are considered. First, consider the cartesian product of two paths, say P_m and P_n . Since $P_2 \times P_2 \cong C_4$, only cartesian products where at least one of m or n , say n , is greater than 2, are considered.

Proposition 4.5 Suppose $d \in \mathbb{N}_0$ and $m, n \in \mathbb{N}$ with $n \geq m$ and $n \geq 3$. Let P_m and P_n denote paths of orders m and n respectively. Then

$$\chi_d^\Delta(P_m \times P_n) = \begin{cases} 1 & \text{if } (d \geq 4 \text{ and } m \geq 3) \text{ or if } (d \geq 3 \text{ and } m = 2), \\ 2 & \text{if } (d < 4 \text{ and } m \geq 3) \text{ or if } (d < 3 \text{ and } m = 2). \end{cases}$$

Proof: Since $\Delta(P_m \times P_n) = 4$ if $m \geq 3$, it is clear that $\chi_d^\Delta(P_m \times P_n) = 1$ if $m \geq 3$ and $d \geq 4$, because in this case all the vertices of $P_m \times P_n$ may be coloured with the same colour. Similarly, $\Delta(P_2 \times P_n) = 3$ and it follows that $\chi_d^\Delta(P_2 \times P_n) = 1$ if $d \geq 3$, because in this case all the vertices of $P_2 \times P_n$ may be coloured with the same colour. Otherwise, let $\{u_1, \dots, u_m\}$ and $\{v_1, \dots, v_n\}$ denote the vertices of P_m and P_n respectively and assign colour 1 to all the vertices of $P_m \times P_n$ labelled (u_i, v_j) for which $i + j$ is even and assign colour 2 to all the vertices of $P_m \times P_n$ labelled (u_i, v_j) for which $i + j$ is odd. This colouring with two colours is best possible, because if only one colour were used the colour induced maximum degree would be $\Delta(P_m \times P_n) > d$. ■

A χ_d^Δ -colouring of $P_m \times P_n$ for $m \geq 3$ and $d < 4$ is shown in Figure 4.5(a). The values for the related inversion number, $D_x^\Delta(P_m \times P_n)$, follow directly from Proposition 4.5 via Proposition 3.3 as stated below.

Corollary 4.8 *Suppose $x, m, n \in \mathbb{N}$ with $n \geq m$ and $n \geq 3$. Then*

$$D_x^\Delta(P_m \times P_n) = \begin{cases} 0 & \text{if } x = 2, \\ 3 & \text{if } x = 1 \text{ and } m = 2, \\ 4 & \text{if } x = 1 \text{ and } m \geq 3, \end{cases}$$

where P_m and P_n denote paths of orders m and n respectively. ■

Next, consider the cartesian product of a path P_m and a cycle C_n .

Proposition 4.6 *Suppose $d \in \mathbb{N}_0$ and $m, n \in \mathbb{N}$ with $n \geq 3$. Let P_m and C_n denote respectively a path of order m and a cycle of order n . Then*

$$\chi_d^\Delta(P_m \times C_n) = \begin{cases} 1 & \text{if } (d \geq 4 \text{ and } m \geq 3) \text{ or if } (d \geq 3 \text{ and } m = 2), \\ 2 & \text{if } (d < 4, m \geq 3 \text{ and } n \text{ is even}) \text{ or if } (d < 3, m = 2 \text{ and } n \text{ is even}), \\ 3 & \text{if } d = 0 \text{ and } n \text{ is odd.} \end{cases}$$

Proof: Since $\Delta(P_m \times C_n) = 4$ if $m \geq 3$, it is clear that $\chi_d^\Delta(P_m \times C_n) = 1$ if $m \geq 3$ and $d \geq 4$, because in this case all the vertices of $P_m \times C_n$ may be coloured with the same colour. Similarly, $\Delta(P_2 \times C_n) = 3$ and it follows that $\chi_d^\Delta(P_2 \times C_n) = 1$ if $d \geq 3$, because in this case all the vertices of $P_2 \times C_n$ may also be coloured with the same colour.

For the remainder of this proof let $\{u_1, \dots, u_m\}$ and $\{v_1, \dots, v_n\}$ denote the vertices of P_m and C_n respectively. If $d < 4$, $m \geq 3$ and n is even, then one colour is insufficient for a χ_d^Δ -colouring of $P_m \times C_n$, since $d < \Delta(P_m \times C_n) = 4$. Similarly, if $d < 3$ and n is even, then one colour is insufficient for a χ_d^Δ -colouring of $P_2 \times C_n$, since $d < \Delta(P_2 \times C_n) = 3$. However, if either $d < 4$, $m \geq 3$ and n is even or $d < 3$, $m = 2$ and n is even, a colouring of $P_m \times C_n$ using two colours, may be obtained by colouring all the vertices of $P_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even with one colour and all the vertices of $P_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is odd with another colour.

If $d = 0$ and n is odd, then two colours are insufficient for a χ_0^Δ -colouring of $P_m \times C_n$, because at least two adjacent vertices will receive the same colour if only two colours are used, resulting in a colour induced maximum degree of at least $1 > d$. However, a χ_0^Δ -colouring of $P_m \times C_n$ using three colours may be obtained by colouring all the vertices of $P_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even and $j \neq n$ [$i + j$ is odd and $j \neq 1$, respectively] with colour 1 [colour 2, respectively]. Finally, all the vertices of $P_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even and $j = n$ as well as those for which $i + j$ is odd and $j = 1$, are coloured with colour 3. ■

A χ_0^Δ -colouring of $P_m \times C_n$ with n odd, is shown in Figure 4.5(b). The values for the related inversion number, $D_x^\Delta(P_m \times C_n)$, as obtained via Proposition 3.3 are given below.

Corollary 4.9 *Suppose $x, m, n \in \mathbb{N}$ with $n \geq 3$. Then*

$$D_x^\Delta(P_m \times C_n) = \begin{cases} 0 & \text{if } (x = 3 \text{ and } n \text{ is odd}) \text{ or if } (x = 2 \text{ and } n \text{ is even}), \\ 1 & \text{if } x = 2 \text{ and } n \text{ is odd}, \\ 3 & \text{if } x = 1 \text{ and } m = 2, \\ 4 & \text{if } x = 1 \text{ and } m \geq 3, \end{cases}$$

where P_m and C_n denote respectively a path of order m and a cycle of order n . ■

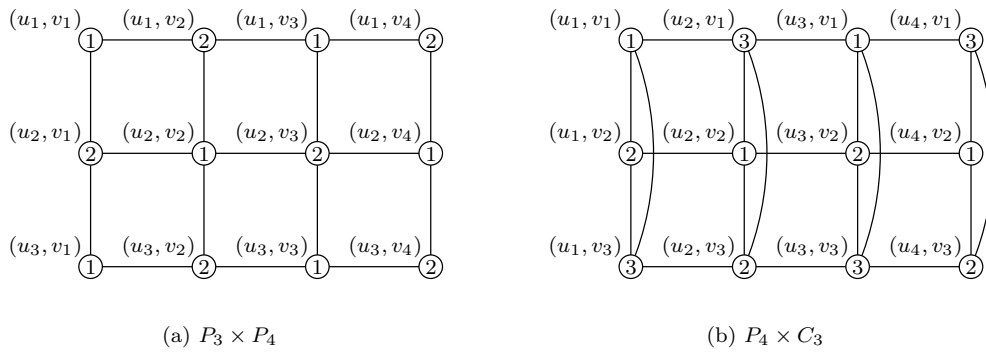


FIGURE 4.5: (a) A χ_d^Δ -colouring of $P_3 \times P_4$ according to the colouring strategy in the proof of Proposition 4.5 for $m \geq 3$ and $d < 4$, where $\chi_d^\Delta(P_3 \times P_4) = 2$. (b) A χ_0^Δ -colouring of $P_4 \times C_3$ according to the colouring strategy in the proof of Proposition 4.6, where $\chi_0^\Delta(P_4 \times C_3) = 3$.

The cartesian product of two cycles, say C_m and C_n , is considered next. Since the maximum degree chromatic number for a cycle C_n for $d = 0$ depends on whether n is even or odd, it is expected that the maximum degree chromatic number for the cartesian product of two cycles C_m and C_n will also be dependent on whether m and n are even or odd. This is confirmed in the following proposition.

Proposition 4.7 *Suppose $d \in \mathbb{N}_0$ and $m, n \in \mathbb{N}$ with $m, n \geq 3$. Let C_m and C_n denote cycles of orders m and n respectively. Then*

$$\chi_d^\Delta(C_m \times C_n) = \begin{cases} 1 & \text{if } d \geq 4, \\ 2 & \text{if } (d \leq 3 \text{ and } m \text{ and } n \text{ are both even}), \\ & \text{or if } (d \in \{1, 2, 3\} \text{ and exactly one of } m \text{ or } n \text{ is even}), \\ & \text{or if } (d \in \{2, 3\} \text{ and } m \text{ and } n \text{ are both odd}), \\ 3 & \text{if } d = 0 \text{ and exactly one of } m \text{ or } n \text{ is even),} \\ & \text{or if } (d \in \{0, 1\} \text{ and } m \text{ and } n \text{ are both odd}). \end{cases}$$

Proof: Since $\Delta(C_m \times C_n) = 4$, it is clear that $\chi_d^\Delta(C_m \times C_n) = 1$ if $d \geq 4$, because in this case all the vertices of $C_m \times C_n$ may be coloured with the same colour.

For the remainder of this proof let $\{u_1, \dots, u_m\}$ and $\{v_1, \dots, v_n\}$ denote the vertices of C_m and C_n respectively. If $d \leq 3$, then one colour is insufficient for a χ_d^Δ -colouring of $C_m \times C_n$, since $d < \Delta(C_m \times C_n) = 4$. However, if $d \leq 3$ and m and n are both even, a colouring of $C_m \times C_n$ using two colours, may be obtained by colouring all the vertices of $C_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even with one colour and all the vertices of $C_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is odd with another colour. If either $d \in \{1, 2, 3\}$ and exactly one of m or n is even, or $d \in \{2, 3\}$ and m and n are both odd, then the same colouring strategy as for the case where $d \leq 3$ with m and n both even yields a χ_d^Δ -colouring of $C_m \times C_n$.

If $d = 0$ and exactly one of m or n is even, then two colours are insufficient for a χ_0^Δ -colouring of $C_m \times C_n$, because at least two adjacent vertices will receive the same colour if only two colours are used, resulting in a colour induced maximum degree of at least $1 > d$. Suppose, m is even and n is odd (if n is even and m is odd, exchange the indices i and j), then a χ_0^Δ -colouring of $C_m \times C_n$ using three colours may be obtained by colouring all the vertices of $C_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even and $j \neq n$ [$i + j$ is odd and $j \neq 1$, respectively] with colour 1 [colour 2, respectively]. Finally, all the vertices of $C_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even and $j = n$ as well as those for which $i + j$ is odd and $j = 1$, are coloured with colour 3.

Similarly, if $d \in \{0, 1\}$ and m and n are both odd, then two colours are insufficient for a χ_d^Δ -colouring of $C_m \times C_n$, because if only two colours are used at least one vertex will receive the same colour as two other vertices that are adjacent to this first vertex, resulting in a colour induced maximum degree of at least $2 > d$. However, in this case a colouring of $C_m \times C_n$ using three colours is possible by first colouring

all the vertices of $C_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even, $i \neq 1$ and $j \neq n$ [$i + j$ is odd, $i \neq m$ and $j \neq 1$, respectively] with colour 1 [colour 2, respectively]. Next, all the vertices of $C_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is odd and either $i = m$ or $j = 1$ are coloured with colour 3. All the vertices of $C_m \times C_n$ labelled (u_i, v_j) for which $i + j$ is even and either $i = 1$, but $j \neq 1$, or $j = n$, but $i \neq m$, are also coloured with colour 3. Finally, the vertices (u_1, v_1) and (u_m, v_n) may be coloured with colour 1 and the vertex (u_m, v_1) may be coloured with colour 2. ■

Two examples of χ_d^Δ -colourings of $C_m \times C_n$ are illustrated in Figures 4.6(a) and (b), namely for $d \in \{1, 2, 3\}$ where exactly one of m or n is even, and for $d \in \{0, 1\}$ where m and n are both odd, respectively. Again, the values for the related inversion number, $D_x^\Delta(C_m \times C_n)$, follow directly from Proposition 4.7 via Proposition 3.3 as stated below.

Corollary 4.10 *Suppose $x, m, n \in \mathbb{N}$ with $m, n \geq 3$. Then*

$$D_x^\Delta(C_m \times C_n) = \begin{cases} 0 & \text{if } x = 3 \text{ and at least one of } m \text{ or } n \text{ is odd,} \\ & \text{or if } x = 2 \text{ and } m \text{ and } n \text{ are both even,} \\ 1 & \text{if } x = 2 \text{ and exactly one of } m \text{ or } n \text{ is even,} \\ 2 & \text{if } x = 2 \text{ and } m \text{ and } n \text{ are both odd,} \\ 4 & \text{if } x = 1, \end{cases}$$

where C_m and C_n denote cycles of orders m and n respectively. ■

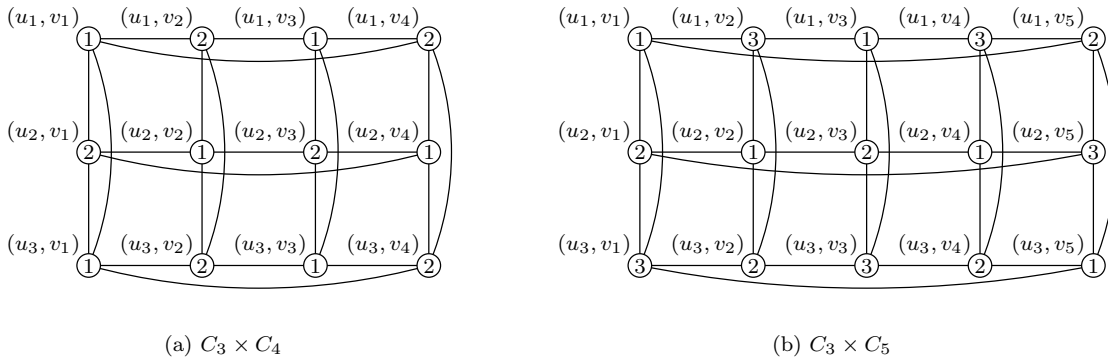


FIGURE 4.6: χ_d^Δ -colourings of $C_m \times C_n$ according to the colouring strategy in the proof of Proposition 4.7 with (a) $d \in \{1, 2, 3\}$ where one of m or n is even and the other odd, and $\chi_d^\Delta(C_3 \times C_4) = 2$ and with (b) $d \in \{0, 1\}$ where m and n are both odd, and $\chi_d^\Delta(C_3 \times C_5) = 3$.

Finally, consider the cartesian product of two complete graphs, say K_m and K_n . The problem of determining precisely the maximum degree chromatic number for the cartesian product of two complete graphs, K_m and K_n , for all values of $d \in \mathbb{N}_0$ seems to be a very hard problem. The maximum degree chromatic number of $K_m \times K_n$ is therefore considered separately for different values of d , starting with $d = 0$.

Proposition 4.8 *Suppose $m, n \in \mathbb{N}$ with $m \leq n$. Let K_m and K_n denote complete graphs of orders m and n respectively. Then $\chi_0^\Delta(K_m \times K_n) = n$ and $D_n^\Delta(K_m \times K_n) = 0$.*

Proof: Since $K_n \subseteq K_m \times K_n$ it follows that $\chi_0^\Delta(K_m \times K_n) \geq \chi(K_n) = n$. However, a χ_0^Δ -colouring of $K_m \times K_n$ in n colours is possible: let $\{u_1, \dots, u_m\}$ denote the vertices of K_m and let $\{v_1, \dots, v_n\}$ denote the vertices of K_n . Then for all $i = 1, \dots, m$ and all $j = 1, \dots, n$ colour the vertex (u_i, v_j) of $K_m \times K_n$ with colour $k \equiv i + j - 1 \pmod{n}$. ■

The colouring strategy in the proof of Proposition 4.8 is demonstrated in Figure 4.7 for the cartesian product of K_5 and K_7 .

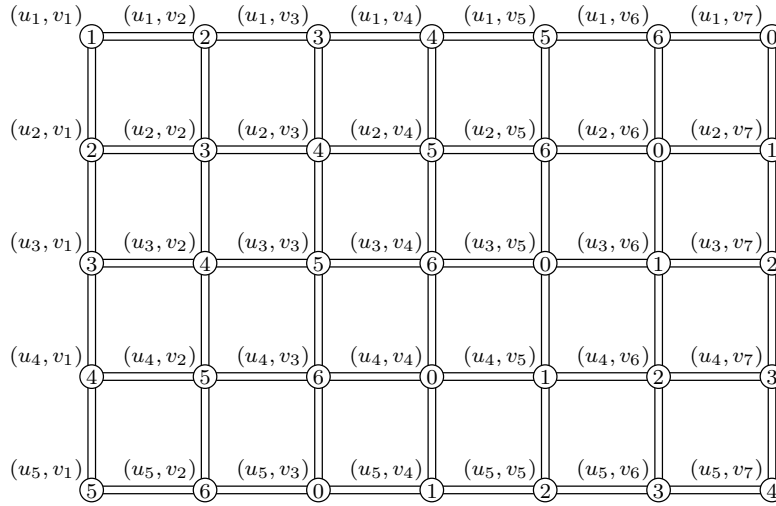


FIGURE 4.7: A graphical representation of a χ_0^Δ -colouring as described in the proof of Proposition 4.8 for the cartesian product of K_5 and K_7 . The double lines between vertices indicate that all the vertices in the same row or in the same column are adjacent, since each row and each column represents a copy of K_7 and K_5 respectively.

For the case of determining $\chi_1^\Delta(K_m \times K_n)$, it is first noticed that, in any χ_1^Δ -colouring of $K_m \times K_n$, at most two vertices from each copy of K_m and K_n may be coloured with the same colour. Furthermore, if two vertices of a copy of K_m are coloured with one colour, say colour i , then two copies of K_n each contains one of these two vertices coloured with colour i . Hence, no other vertex in either of these two copies of K_n may be coloured with colour i , since the allowable colour class induced maximum degree of 1 would otherwise be compromised. This observation leads to the maximum number of vertices in a colour class of a χ_1^Δ -colouring of $K_m \times K_n$, as stated below.

Lemma 4.1 *Suppose $m, n \in \mathbb{N}$ with $m \leq n$. Let K_m and K_n denote complete graphs of orders m and n respectively. Then an upper bound on the number of vertices in a colour class of a χ_1^Δ -colouring of $K_m \times K_n$ is*

$$\eta = \begin{cases} 2m & \text{if } n \geq 2m, \\ 2\left(\lfloor \frac{n}{2} \rfloor + \lfloor \frac{y}{3} \rfloor\right) + (n \pmod{2}) + k & \text{if } n \leq 2m - 1, \end{cases}$$

where $y = m - \lfloor n/2 \rfloor$ and where

$$k = \begin{cases} 1 & \text{if } (y \equiv 2 \pmod{3}) \text{ or if } (y \equiv 1 \pmod{3}) \text{ and } n \text{ is odd} \\ 0 & \text{otherwise.} \end{cases}$$

Proof: If $n \geq 2m$, the stated upper bound of $\eta = 2m$ on the number of vertices in a colour class may be achieved by colouring two vertices in each of the m copies of K_n such that no copy of K_m contains more than one vertex in the same colour class. No more than $\eta = 2m$ vertices may be in any colour class, because then at least one copy of K_n must contain three (or more) vertices in the same colour class, thus violating the allowable colour class induced maximum degree of 1.

Suppose $n \leq 2m - 1$ and suppose one colour class, \mathcal{C}_i , is formed, as before, by monocolouring a K_2 in as many as possible copies of K_n (bar one if n is odd, in which case only one vertex is coloured in one copy of K_n), such that exactly one vertex in each of the n copies of K_m is coloured with this colour, say colour i . Then, except when $n = 2m - 1$, copies of K_n exist in which no vertex is coloured with colour i . No vertex in these copies may be coloured with colour i , since then the allowable colour class induced maximum degree of 1 would be compromised. However, for every three copies of K_n containing no vertex in \mathcal{C}_i , more vertices may be coloured with colour i if the monocolouring of a K_2 in one copy of K_n , say copy j , is replaced by a monocolouring of a K_2 in each of two copies of K_m such that one vertex of one of these two new monocoloured copies of K_2 is in copy j of K_n , and the other three vertices of

these two new monocoloured copies of K_2 are each in one copy of K_n not containing any vertex in \mathcal{C}_i . The number of these replacements is determined by $\lfloor y/3 \rfloor$. Furthermore, in each of these replacements four vertices are now coloured with colour i .

After the replacement described above, the number of copies of K_n still containing a monochromatic K_2 coloured with colour i , is $\lfloor n/2 \rfloor - \lfloor y/3 \rfloor$ and if n is odd, then one vertex in one copy of K_n is also coloured with colour i as described above. Let this vertex be in the ℓ -th copy of K_m . Furthermore, after the replacement described above, the following cases may result:

- (i) All copies of K_m and K_n contain at least one vertex in \mathcal{C}_i in which case no further improvements on the number of vertices in \mathcal{C}_i can be made.
- (ii) One copy of K_n does not contain any vertex in \mathcal{C}_i . If n is odd, then the vertex in this copy of K_n and in the ℓ -th copy of K_m may still be coloured with colour i . On the other hand, if n is even, no vertex in this copy of K_n may be coloured with colour i without compromising the allowable colour class induced maximum degree of 1.
- (iii) Two copies of K_n does not contain any vertex in \mathcal{C}_i . If n is odd, then the vertex in one of these copies of K_n and in the ℓ -th copy of K_m may still be coloured with colour i . Finally, if n is even, the monochromatic K_2 in one copy of K_n , say copy p , may be replaced by the colouring of three vertices as follows. Let the vertices of this K_2 coloured with colour i in the p -th copy of K_n be in the q -th and r -th copies of K_m . Recolour the vertex in the p -th copy of K_n and the q -th copy of K_m with colour i again. Then colour the K_2 in the r -th copy of K_m and in the two copies of K_n not yet containing any vertex in \mathcal{C}_i , with colour i .

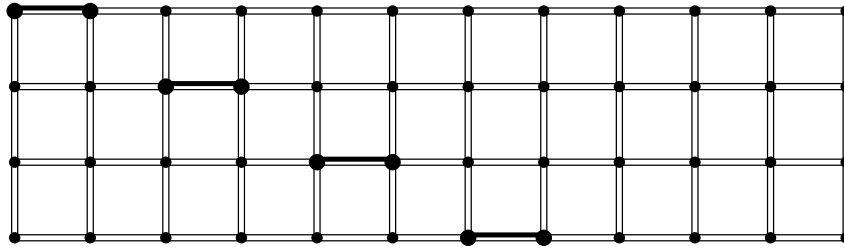
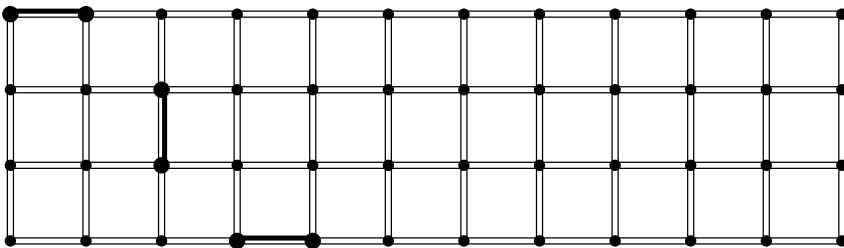
(a) $K_4 \times K_{12}$ (b) $K_4 \times K_{12}$

FIGURE 4.8: A graphical representation of the attained upper bound on the number of vertices in a colour class via the strategy in Lemma 4.1, when $n \geq 2m$. The double lines between vertices indicate that all the vertices in the same row or in the same column are adjacent, since each row and each column represents a copy of K_{12} and K_4 respectively. The bold vertices may be coloured with the same colour. (a) The upper bound on the number of vertices per colour class in a χ_1^Δ -colouring of $K_4 \times K_{12}$ is attained when each colour class contains two vertices from each copy of K_{12} , in which case $\eta = 8$. (b) The total number of vertices in a colour class is $6 < 8 = \eta$ if two copies of K_{12} do not contain two vertices coloured with the same colour.

The cases above result in the value of k stated in the lemma. No more vertices may be coloured with colour i without compromising the allowable colour class induced maximum degree of 1, and no further improvements on the number of vertices in \mathcal{C}_i can be made. ■

The strategy in Lemma 4.1 to obtain an upper bound on the number of vertices, η , in a colour class when $n \geq 2m$ is illustrated in Figure 4.8(a). Figure 4.8(b) illustrates how the total number of vertices in a colour class decreases when not all copies of K_n have two vertices per colour class for the case $n \geq 2m$.

An illustration of how to obtain the upper bound on the number of vertices, η , per colour class via Lemma 4.1 for the cartesian product of K_{11} and K_{12} when $n \leq 2m - 1$, may be found in Figure 4.9. A partial $(1, x)$ -colouring of $K_{11} \times K_{12}$ is shown in Figure 4.9(a) where one colour class contains two vertices from some copies of K_{12} and exactly one vertex in each copy of K_{11} , so that $y = 5$. Thus, the monochromatic K_2 in one copy of K_{12} , say copy 5, in the colouring in Figure 4.9(a) may be replaced by a monochromatic K_2 of the same colour in each of two copies of K_{11} not yet containing a K_2 of that colour. The remaining number of copies of K_2 of the particular colour in question in copies of K_{12} is five. Finally, since $y \equiv 2 \pmod{3}$, it follows that $k = 1$, because in this case the colouring of a monochromatic K_2 in one copy of K_{12} , say copy 6, may be replaced by the colouring of a K_2 in a copy of K_{11} not yet containing a monochromatic K_2 of that colour (copy 12), as well as one more vertex in the final copy of K_{11} not yet containing a K_2 of that colour (copy 11). This results in the colouring of one more vertex than before. Therefore, the upper bound on the number of vertices in a colour class of a χ_1^Δ -colouring of $K_{11} \times K_{12}$ is $\eta = 15$ as illustrated in Figure 4.9(b).

Two more examples of the strategy in Lemma 4.1 are shown in Figure 4.10 for the case where $n \leq 2m - 1$. In Figure 4.10(a), $n = 2m - 1$ so that $y = 1$ and $k = 1$, while the case $n = 2(m - 1)$ is illustrated in Figure 4.10(b), where $y = 1$ and $k = 0$.

Using Lemma 4.1, the following lower bound on $\chi_1^\Delta(K_m \times K_n)$ may now be established.

Proposition 4.9 *Suppose $m, n \in \mathbb{N}$ with $m \leq n$. Let K_m and K_n denote complete graphs of orders m and n respectively. Then*

$$\chi_1^\Delta(K_m \times K_n) \geq \left\lceil \frac{mn}{\eta} \right\rceil \quad \text{and} \quad D_{\lceil mn/\eta \rceil}^\Delta(K_m \times K_n) \geq 1,$$

where η is the upper bound on the number of vertices in a colour class in Lemma 4.1.

Proof: If each colour class contains η vertices, as determined in Lemma 4.1, then the smallest number of colour classes is the order of $K_m \times K_n$ divided by η . ■

In some cases, the exact value of $\chi_1^\Delta(K_m \times K_n)$ may be determined, as stated in the proposition below.

Proposition 4.10 *Suppose $m, n \in \mathbb{N}$ with $m \leq n$. Let K_m and K_n denote complete graphs of orders m and n respectively. If $n \geq 2m - 1$, then*

$$\chi_1^\Delta(K_m \times K_n) = \left\lceil \frac{mn}{\eta} \right\rceil \quad \text{and} \quad D_{\lceil mn/\eta \rceil}^\Delta(K_m \times K_n) = 1,$$

where η is the upper bound on the number of vertices in a colour class in Lemma 4.1.

Proof: Suppose $K_m \times K_n$ is represented in a grid, where each row represents a copy of K_n and each column represents a copy of K_m . Let $\{u_1, \dots, u_m\}$ denote the vertices of K_m and let $\{v_1, \dots, v_n\}$ denote the vertices of K_n . Also, let $\lceil mn/\eta \rceil = x$, where η is the upper bound in Lemma 4.1.

Let $V_j = \{(u_1, v_j), (u_2, v_j), \dots, (u_m, v_j)\}$ for all $j = 1, \dots, n$ and let $W_\ell = V_{2\ell-1} \cup V_{2\ell}$ for all $\ell = 1, \dots, \lfloor n/2 \rfloor$. If n is odd, also let $W_{(n+1)/2} = V_n$. Suppose $n = 2m - 1$. Then by Lemma 4.1, $\eta = n$ so that $x = m$. In this case a total of $\lfloor (2m - 1)/2 \rfloor + 1 = m = x$ sets W_ℓ are formed. Now suppose $n \geq 2m$. Then by Lemma 4.1, $\eta = 2m$ so that $x = \lceil n/2 \rceil$ and again a total of x sets W_ℓ are formed.

A χ_1^Δ -colouring of $K_m \times K_n$ in x colours may now be constructed as follows. For each $\ell = 1, \dots, x$, colour the two vertices $(u_i, v_{2\ell-1})$ and $(u_i, v_{2\ell})$ in W_ℓ [the vertex $(u_i, v_{2\ell-1})$ in W_ℓ if $\ell = x$ and n is odd] with

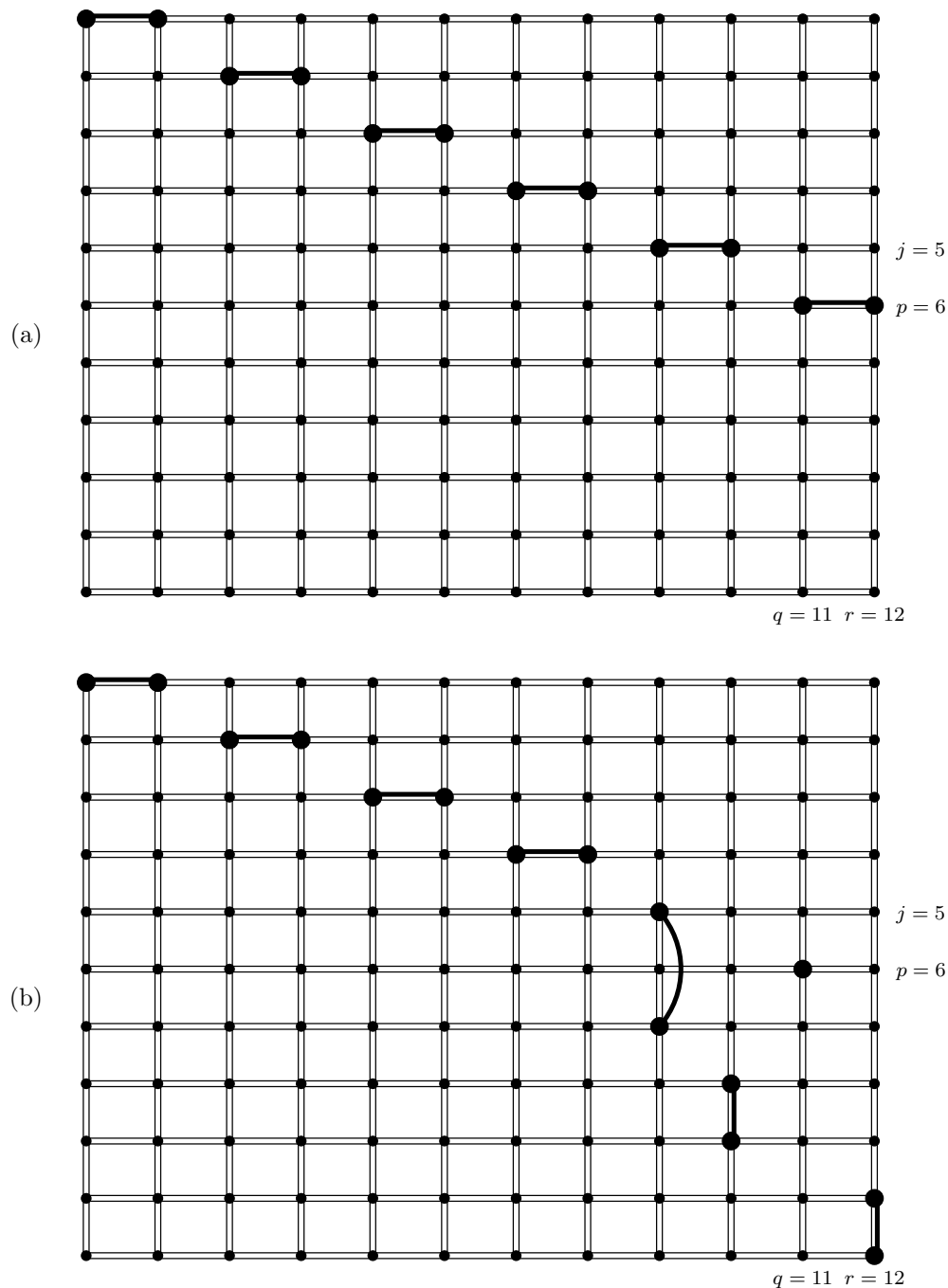


FIGURE 4.9: A graphical representation of how to obtain an upper bound on the number of vertices, η , per colour class via Lemma 4.1 when $n \leq 2m - 1$. As before, the double lines between vertices indicate that all the vertices in the same row or in the same column are adjacent, since each row and each column represents a copy of K_{12} and K_{11} respectively. The bold vertices represent all the vertices that may be coloured with the same colour. The specific choices for the parameters j, p, q and r in the proof of Lemma 4.1 used to replace monochromatic copies of K_2 with other vertices coloured with the particular colour in question are indicated in the figures as well. (a) The number of vertices per colour class in a $(1, x)$ -colouring of $K_{11} \times K_{12}$ when one colour class contains two vertices from some copies of K_{12} and exactly one vertex in each copy of K_{11} . (b) The upper bound on the number of vertices per colour class of a χ_1^Δ -colouring of $K_{11} \times K_{12}$ after all replacements described in the proof of Lemma 4.1 have been performed, is $\eta = 15$.

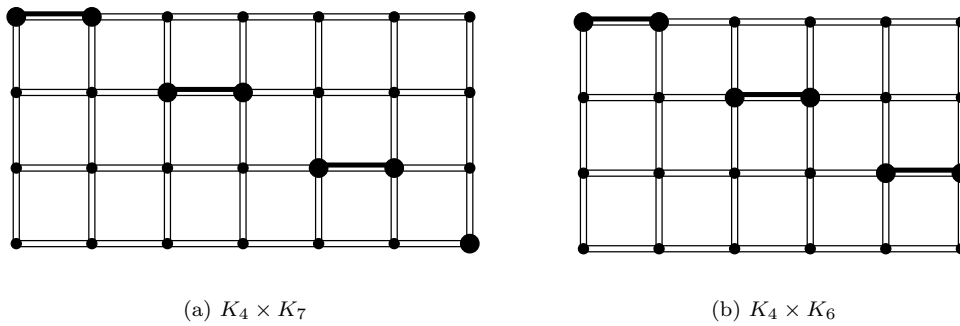


FIGURE 4.10: A graphical representation of the strategy in Lemma 4.1 to obtain an upper bound on the number of vertices in a colour class of a χ_1^Δ -colouring of $K_m \times K_n$, where (a) $n = 2m - 1$ as illustrated by the case $K_4 \times K_7$ (where $y = 1, k = 1$ and $\eta = 7$), and (b) $n = 2(m - 1)$ as illustrated by the case $K_4 \times K_6$ (where $y = 1, k = 0$ and $\eta = 6$).

colour $y \equiv \ell + i - 1 \pmod{x}$, for all $i = 1, \dots, m$. Therefore $\chi_1^\Delta(K_m \times K_n) \leq x$. From Proposition 4.9 it then follows that $\chi_1^\Delta(K_m \times K_n) = \lceil (m \times n)/\eta \rceil$ if $n \geq 2m - 1$. ■

The colouring strategy in the proof of Proposition 4.10 is demonstrated in Figure 4.11 for the cartesian product of K_4 and K_9 . Since $9 > 2 \times 4$, $\eta = 2 \times 4 = 8$ by Lemma 4.1, and a total of $x = \lceil (4 \times 9)/8 \rceil = 5$ sets W_ℓ are formed, so that $\chi_1^\Delta(K_4 \times K_9) = 5$.

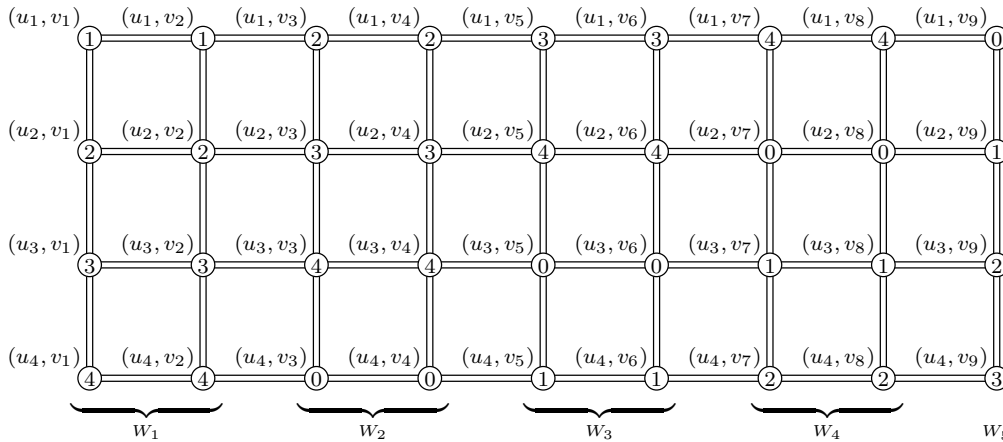


FIGURE 4.11: A graphical representation of a χ_1^Δ -colouring in 5 colours, as described in the proof of Proposition 4.10, for the cartesian product of K_4 and K_9 . The brackets indicate the sets W_ℓ formed during the colouring strategy. Note that since 9 is odd, the final set W_5 contains only one column from the original grid representation of $K_4 \times K_9$.

The problems of determining $\chi_1^\Delta(K_m \times K_n)$ for $n \leq 2m$ and of determining $\chi_d^\Delta(K_m \times K_n)$ for $d \geq 2$ remain for future investigations — it is anticipated that such investigations will necessarily be exceedingly technical. However, it is easy to determine when one colour suffices and this section closes with this final result.

Proposition 4.11 *Suppose $m, n \in \mathbb{N}$ with $m \leq n$. Let K_m and K_n denote complete graphs of orders m and n respectively. Then*

$$\chi_{(m+n-2)}^\Delta(K_m \times K_n) = 1 \quad \text{and} \quad D_1^\Delta(K_m \times K_n) = m + n - 2.$$

Proof: Since $\Delta(K_m \times K_n) = \Delta(K_m) + \Delta(K_n) = m + n - 2$ it follows that $K_m \times K_n$ may be coloured with one colour if $d = m + n - 2$. Also, if only one colour may be used to colour $K_m \times K_n$, then $D_1^\Delta(K_m \times K_n) = \Delta(K_m \times K_n)$. ■

4.5 Circulants

The problem of determining the maximum degree chromatic number of a circulant of order n , $C_n\langle i_1, \dots, i_z \rangle$, with arbitrary connection set $\{i_1, \dots, i_z\}$ for all values of $d \in \mathbb{N}_0$ seems to be a very hard problem. In fact, to the best knowledge of the author, the classical chromatic number of a circulant of order n , $C_n\langle i_1, \dots, i_z \rangle$, for all choices of the connection set $\{i_1, \dots, i_z\}$, has not yet been determined [59, 81]. Therefore, only a few results on the maximum degree chromatic number for a circulant of order n are listed here, starting with the maximum degree chromatic number of an elementary circulant of order n .

Proposition 4.12 *Suppose $d \in \mathbb{N}_0$ and $i, n \in \mathbb{N}$. Let $m = \gcd(n, i)$ and let $C_n\langle i \rangle$ denote an elementary circulant of order n . Then*

$$\chi_d^\Delta(C_n\langle i \rangle) = \begin{cases} 1 & \text{if } d \geq 2 \text{ or if } (d = 1 \text{ and } i = n/2), \\ 2 & \text{if } (d = 1 \text{ and } i \neq n/2) \text{ or if } (d = 0 \text{ and } i = n/2), \\ & \text{or if } (d = 0, n \text{ is even, and } n \text{ \& } i \text{ are coprime}), \\ & \text{or if } (d = 0, m > 1, \text{ and } r = n/m \text{ is even}), \\ 3 & \text{if } (d = 0, n \text{ is odd, and } n \text{ \& } i \text{ are coprime}), \\ & \text{or if } (d = 0, m > 1, \text{ and } r = n/m \text{ is odd}), \end{cases}$$

Proof: Since $\Delta(C_n\langle i \rangle) \leq 2$ for all choices of n and i , it follows that if $d \geq 2$ then $\chi_d^\Delta(C_n\langle i \rangle) = 1$, because in this case one colour may be used to colour all the vertices of $C_n\langle i \rangle$. If $i = n/2$, then it follows by (2.1.3) and Corollary 4.3 that $\chi_1^\Delta(C_n\langle i \rangle) = 1$ and $\chi_0^\Delta(C_n\langle i \rangle) = 2$. The other cases where $\chi_d^\Delta(C_n\langle i \rangle) = 2$ and the case where $\chi_d^\Delta(C_n\langle i \rangle) = 3$ follow directly from (2.1.2), (2.1.4) and Proposition 4.2. ■

As before, the values of the related inversion number, $D_x^\Delta(C_n\langle i \rangle)$, follow directly from Proposition 4.12 via Proposition 3.3 as stated below.

Corollary 4.11 *Let $x, n, i \in \mathbb{N}$ and $m = \gcd(n, i)$. Then*

$$D_x^\Delta(C_n\langle i \rangle) = \begin{cases} 0 & \text{if } (x = 3, n \text{ is odd, and } n \text{ \& } i \text{ are coprime}), \\ & \text{or if } (x = 3, m > 1, \text{ and } r = n/m \text{ is odd}), \\ & \text{or if } (x = 2, n \text{ is even, and } n \text{ \& } i \text{ are coprime}), \\ & \text{or if } (x = 2, m > 1, \text{ and } r = n/m \text{ is even}), \\ & \text{or if } (x = 2, \text{ and } i = n/2), \\ 1 & \text{if } (x = 2, n \text{ is odd, and } n \text{ \& } i \text{ are coprime}), \\ & \text{or if } (x = 2, m > 1, \text{ and } r = n/m \text{ is odd}), \\ & \text{or if } (x = 1, \text{ and } i = n/2), \\ 2 & \text{if } (x = 1, \text{ and } i \neq n/2), \end{cases}$$

where $C_n\langle i \rangle$ denotes an elementary circulant of order n . ■

Recall from §2.1 that $C_n\langle i \rangle \cong C_n\langle n - i \rangle$ and that a composite circulant of order n , $C_n\langle i_1, \dots, i_z \rangle$, may be viewed as a construction from two or more elementary circulants via an edge union (*i.e.* $C_n\langle i_1, \dots, i_z \rangle = \bigoplus_{k=1}^z C_n\langle i_k \rangle$). Therefore, without loss of generality, it is sufficient to study composite circulants with connection set $\{i_1, \dots, i_z\}$, where $i_1 < i_2 < \dots < i_z \leq n/2$. Composite circulants with a connection set of cardinality two are considered first. Since the only composite circulants of orders 4 and 5 are $C_4\langle 1, 2 \rangle \cong K_4$ and $C_5\langle 1, 2 \rangle \cong K_5$ respectively, only composite circulants of orders $n \geq 6$ will be considered. Heuberger [65, Theorem 3.2] proved the following concerning the parameter $\chi_0^\Delta(C_n\langle i_1, i_2 \rangle)$.

Proposition 4.13 *Suppose $n, i_1, i_2 \in \mathbb{N}$ where $n \geq 6$ and $i_1 < i_2 \leq n/2$. Then*

$$\chi_0^\Delta(C_n\langle i_1, i_2 \rangle) = \begin{cases} 2 & \text{if } i_1 \text{ and } i_2 \text{ are odd, and } n \text{ is even,} \\ 4 & \text{if } (3 \nmid n \text{ and } i_2 = 2i_1) \text{ or if } (n = 13 \text{ and } i_2 = 5i_1), \\ 3 & \text{otherwise,} \end{cases}$$

where $C_n\langle i_1, i_2 \rangle$ denotes a connected composite circulant of order n with connection set $\{i_1, i_2\}$. ■

In the same paper Heuberger [65, Theorem 2.1] proved the following lemma on the bipartiteness of circulants.

Lemma 4.2 *Suppose $n, z, i_1, \dots, i_z \in \mathbb{N}$ where $i_1 < i_2 < \dots < i_z \leq n/2$. Let $\gcd(n, i_1, \dots, i_z) = 1$. Then a connected composite circulant, $C_n\langle i_1, \dots, i_z \rangle$, of order n is bipartite if and only if i_1, \dots, i_z are odd and n is even. ■*

Using Lemma 4.2, the maximum degree chromatic number of a circulant with connection set of cardinality 2, satisfying certain conditions, may be obtained.

Proposition 4.14 *Suppose $d \in \mathbb{N}_0$, $x, n, i_1, i_2 \in \mathbb{N}$ where $n \geq 6$ is even, i_1 and i_2 are odd, $i_1 < i_2 < n/2$ and $\gcd(n, i_1, i_2) = 1$. Let $C_n\langle i_1, i_2 \rangle$ denote a non-singular (connected) composite circulant of order n with connection set $\{i_1, i_2\}$. Then*

$$\chi_d^\Delta(C_n\langle i_1, i_2 \rangle) = \begin{cases} 1 & \text{if } d \geq 4, \\ 2 & \text{if } d < 4, \end{cases} \quad \text{and} \quad D_x^\Delta(C_n\langle i_1, i_2 \rangle) = \begin{cases} 0 & \text{if } x = 2, \\ 4 & \text{if } x = 1. \end{cases}$$

Proof: From Lemma 4.2 $C_n\langle i_1, i_2 \rangle$ is bipartite. Since $\Delta(C_n\langle i_1, i_2 \rangle) = 4$, the result follows directly from Proposition 4.1 and Corollary 4.1. ■

Similarly to the result of Proposition 4.14 for a non-singular composite circulant satisfying the specific conditions stated, the case of a singular composite circulant satisfying the same conditions, may be obtained.

Proposition 4.15 *Suppose $d \in \mathbb{N}_0$, $x, n, i_1 \in \mathbb{N}$ where $n \geq 6$ is even, and both $i_1 < n/2$ and $n/2$ are odd. Let $C_n\langle i_1, n/2 \rangle$ denote a singular composite circulant of order n with connection set $\{i_1, n/2\}$. Then*

$$\chi_d^\Delta(C_n\langle i_1, n/2 \rangle) = \begin{cases} 1 & \text{if } d \geq 3, \\ 2 & \text{if } d < 3, \end{cases} \quad \text{and} \quad D_x^\Delta(C_n\langle i_1, n/2 \rangle) = \begin{cases} 0 & \text{if } x = 2, \\ 3 & \text{if } x = 1. \end{cases}$$

Proof: From Lemma 4.2 $C_n\langle i_1, n/2 \rangle$ is bipartite. Since in this case $\Delta(C_n\langle i_1, n/2 \rangle) = 3$, the result follows directly from Proposition 4.1 and Corollary 4.1. ■

Values of the maximum degree chromatic number of a circulant with connection set $\{1, 2\}$ are stated below.

Proposition 4.16 *Suppose $n, d \in \mathbb{N}$ where $n \geq 6$. Let $C_n\langle 1, 2 \rangle$ denote a non-singular composite circulant of order n with connection set $\{1, 2\}$. Then*

$$\chi_d^\Delta(C_n\langle 1, 2 \rangle) = \begin{cases} 1 & \text{if } d \geq 4, \\ 2 & \text{if } (d \in \{1, 2, 3\} \text{ and } n \equiv 0 \pmod{4}), \\ & \text{or if } (d \in \{2, 3\} \text{ and } n \not\equiv 0 \pmod{4}), \\ 3 & \text{if } (d = 1 \text{ and } n \not\equiv 0 \pmod{4}). \end{cases}$$

Proof: Since $\Delta(C_n\langle 1, 2 \rangle) = 4$ for $n \geq 6$, it is clear that $\chi_d^\Delta(C_n\langle 1, 2 \rangle) = 1$ if $d \geq 4$, because in this case all vertices of $C_n\langle 1, 2 \rangle$ may be coloured with one colour.

For the remainder of this proof let $\{u_1, \dots, u_n\}$ denote the vertices of $C_n\langle 1, 2 \rangle$. If $d \leq 3$, then one colour is insufficient for a χ_d^Δ -colouring of $C_n\langle 1, 2 \rangle$, since $d < \Delta(C_n\langle 1, 2 \rangle) = 4$. However, if $n \pmod{4} = 0$ then

a χ_1^Δ -colouring of $C_n\langle 1, 2 \rangle$ in two colours may be obtained by colouring the vertices u_{4k-3} and u_{4k-2} with colour 1 and the vertices u_{4k-1} and u_{4k} with colour 2 for all $k = 1, \dots, n/4$.

Since u_i, u_{i+1} and u_{i+2} are all pairwise adjacent for all $i = 1, \dots, n-2$, only two vertices in each of these triplets may receive the same colour. Therefore, if u_i and u_{i+2} receive the same colour, say colour 1, then u_{i+1}, u_{i+3} and u_{i+4} may *not* be coloured with colour 1 and they may also not be coloured with the same colour since then u_{i+3} is adjacent to two vertices coloured with the same colour as itself. Thus, in this case a third colour has to be introduced. If u_i and u_{i+1} receive the same colour, say colour 1, then u_{i+2} and u_{i+3} may *not* be coloured with colour 1, but they may be coloured with the same colour, while u_{i+4} and u_{i+5} may be coloured with colour 1 again. By repeating this colouring pattern it is found that if $n \not\equiv 0 \pmod{4}$ then either u_n (if $n \equiv 1 \pmod{4}$) or u_n and u_{n-1} (if $n \equiv 2 \pmod{4}$ or $n \equiv 3 \pmod{4}$) may not be coloured with either colour 1 or colour 2. Hence, two colours are insufficient for a χ_1^Δ -colouring of $C_n\langle 1, 2 \rangle$ if $n \not\equiv 0 \pmod{4}$. However, in this case a χ_1^Δ -colouring of $C_n\langle 1, 2 \rangle$ in three colours is obtained by colouring either u_n , or u_n and u_{n-1} with colour 3. Note that if $n \equiv 3 \pmod{4}$, then u_{n-4} and u_{n-3} are still coloured with colour 2 as described above, but only u_{n-2} is then coloured with colour 1.

If $2 \leq d < 4$ and $n \not\equiv 0 \pmod{4}$, then a χ_2^Δ -colouring of $C_n\langle 1, 2 \rangle$ in two colours is possible: Colour all the vertices u_{2j-1} with colour 1 and colour all the vertices u_{2j} with colour 2, for all $j = 1, \dots, \lfloor n/2 \rfloor$. If n is odd, finally colour vertex u_n also with colour 1. ■

The colouring strategy in the proof of Proposition 4.16 is demonstrated in Figure 4.12 for the cases (a) $d = 1$ with $n \equiv 0 \pmod{4}$, (b) $d = 1$ with $n \equiv 3 \pmod{4}$ and (c) $d = 2$ with n odd.

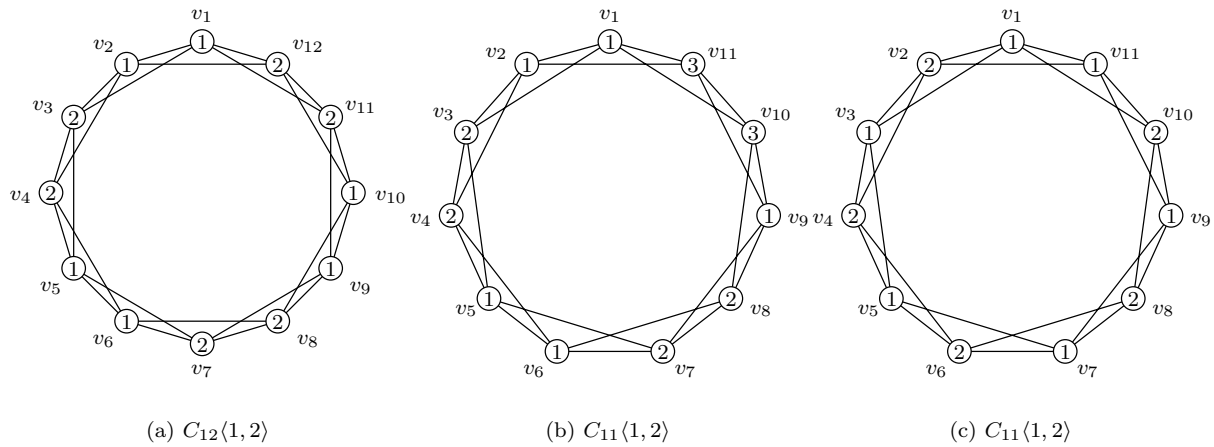


FIGURE 4.12: Examples of χ_d^Δ -colourings of $C_n\langle 1, 2 \rangle$. (a) A χ_1^Δ -colouring of $C_{12}\langle 1, 2 \rangle$ where $n \equiv 12 \equiv 0 \pmod{4}$ showing that $\chi_1^\Delta(C_{12}\langle 1, 2 \rangle) = 2$, (b) a χ_1^Δ -colouring of $C_{11}\langle 1, 2 \rangle$ where $n \equiv 11 \equiv 3 \pmod{4}$ showing that $\chi_1^\Delta(C_{11}\langle 1, 2 \rangle) = 3$ and (c) a χ_2^Δ -colouring of $C_{11}\langle 1, 2 \rangle$ showing that $\chi_2^\Delta(C_{11}\langle 1, 2 \rangle) = 2$.

Some values of the maximum degree chromatic number of a circulant with connection set of cardinality two, satisfying the condition that n must be even, are stated below. The discussion on the maximum degree chromatic number for a circulant with connection set of cardinality two is concluded with this result and the remaining cases are left for future investigations.

Proposition 4.17 Suppose $d, n, i_1, i_2 \in \mathbb{N}$ where $n \geq 6$ is even, and $i_1 < i_2 \leq n/2$. Let $C_n\langle i_1, i_2 \rangle$ denote a composite connected circulant of order n with connection set $\{i_1, i_2\}$. Then

$$\chi_d^\Delta(C_n\langle i_1, i_2 \rangle) = \begin{cases} 1 & \text{if } (d \geq 4 \text{ and } i_2 < n/2) \text{ or if } (d \geq 3 \text{ and } i_2 = n/2), \\ 2 & \text{if } (d \in \{2, 3\} \text{ and } i_2 < n/2) \text{ or if } (d = 2 \text{ and } i_2 = n/2). \end{cases}$$

Proof: Since $\Delta(C_n\langle i_1, i_2 \rangle) = 4$ if $n \geq 6$ and $i_2 < n/2$, and $\Delta(C_n\langle i_1, i_2 \rangle) = 3$ if $n \geq 6$ and $i_2 = n/2$, it is clear that $\chi_d^\Delta(C_n\langle 1, 2 \rangle) = 1$ if either $d \geq 4$ and $i_2 < n/2$, or $d \geq 3$ and $i_2 = n/2$, because in these cases all vertices of $C_n\langle i_1, i_2 \rangle$ may be coloured with one colour.

For the remainder of this proof let $\{u_1, \dots, u_n\}$ denote the vertices of $C_n\langle i_1, i_2 \rangle$. If $d \leq 3$ and $i_2 < n/2$, or if $d \leq 2$ and $i_2 = n/2$, then one colour is insufficient for a χ_d^Δ -colouring of $C_n\langle i_1, i_2 \rangle$, since $d < \Delta(C_n\langle i_1, i_2 \rangle) = 4$ or 3 respectively. However, a χ_2^Δ -colouring of $C_n\langle i_1, i_2 \rangle$ in two colours is possible. If i_1 and i_2 are both even, then $\gcd(n, i_1, i_2) = 2$ and $C_n\langle i_1, i_2 \rangle$ is therefore disconnected. If i_1 and i_2 are both odd, then from Lemma 4.2 $C_n\langle i_1, i_2 \rangle$ is bipartite and a χ_2^Δ -colouring of $C_n\langle i_1, i_2 \rangle$ follows from Proposition 4.14. Finally, if exactly one of i_1 and i_2 is even, then a χ_2^Δ -colouring of $C_n\langle i_1, i_2 \rangle$ in two colours may be obtained by colouring all the vertices u_{2j-1} with colour 1 and all the vertices u_{2j} with colour 2, for all $j = 1, \dots, n/2$. ■

The colouring strategy of $C_n\langle i_1, i_2 \rangle$ in the proof of Proposition 4.17 where exactly one of i_1 and i_2 are even, is demonstrated in Figure 4.13 for the cases (a) $i_2 < n/2$ and (b) $i_2 = n/2$.

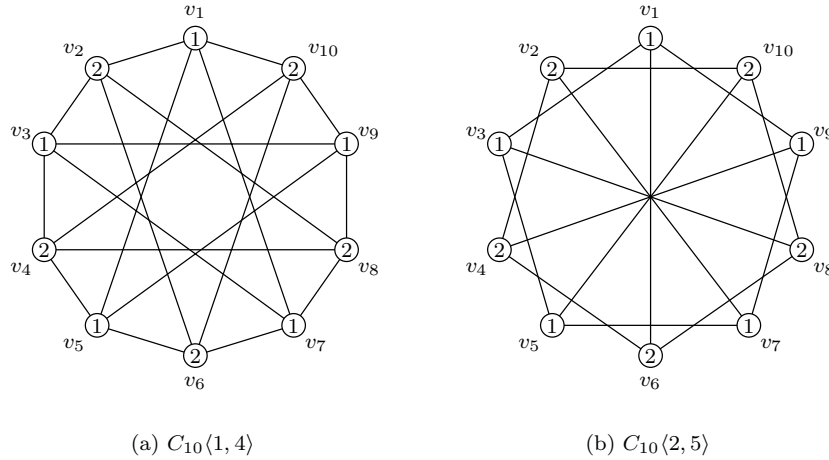


FIGURE 4.13: Examples of χ_2^Δ -colourings of $C_{10}\langle i_1, i_2 \rangle$. (a) A χ_2^Δ -colouring of $C_{10}\langle 1, 4 \rangle$ where $i_2 = 4 < 5 = n/2$ showing that $\chi_2^\Delta(C_{10}\langle 1, 4 \rangle) = 2$ and (b) a χ_2^Δ -colouring of $C_{10}\langle 2, 5 \rangle$ where $i_2 = 5 = n/2$ showing that $\chi_2^\Delta(C_{10}\langle 2, 5 \rangle) = 2$ as well.

Considering composite circulants with a connection set of cardinality three, one notices that the only composite circulants of orders 6 and 7 with a connection set of cardinality three are $C_6\langle 1, 2, 3 \rangle \cong K_6$ and $C_7\langle 1, 2, 3 \rangle \cong K_7$ respectively. Therefore, only composite circulants of orders $n \geq 8$ need to be considered. Meszka *et al.* [82, Theorem 4] proved the following concerning the parameter $\chi_0^\Delta(C_n\langle i_1, i_2, i_3 \rangle)$ where $i_3 = i_1 + i_2 \neq n/2$.

Proposition 4.18 *Suppose $n, i_1, i_2, i_3 \in \mathbb{N}$ where $n \geq 8$ and $i_1 < i_2 < i_3 < n/2$ such that $i_1 + i_2 = i_3$. Then*

$$\chi_0^\Delta(C_n\langle i_1, i_2, i_3 \rangle) = \begin{cases} 3 & \text{if } (3|n \text{ and none of } i_1, i_2 \text{ or } i_3 \text{ is divisible by } 3), \\ 5 & \text{if } (n \neq 7, 11 \text{ is not divisible by } 4 \text{ and } i_1 = 1, i_2 = 2, i_3 = 3), \\ & \text{or if } (C_n\langle i_1, i_2, i_3 \rangle \text{ is isomorphic to } C_{13}\langle 1, 3, 4 \rangle, C_{17}\langle 1, 3, 4 \rangle, C_{18}\langle 1, 3, 4 \rangle, \\ & C_{19}\langle 1, 7, 8 \rangle, C_{25}\langle 1, 3, 4 \rangle, C_{26}\langle 1, 7, 8 \rangle, C_{33}\langle 1, 6, 7 \rangle \text{ or } C_{37}\langle 1, 10, 11 \rangle), \\ 6 & \text{if } n = 11 \text{ and } i_1 = 1, i_2 = 2, i_3 = 3, \\ 4 & \text{otherwise,} \end{cases}$$

where $C_n\langle i_1, i_2, i_3 \rangle$ denotes a connected composite circulant of order n with connection set $\{i_1, i_2, i_3\}$. ■

Meszka *et al.* [81] also proved the following concerning the parameter $\chi_0^\Delta(C_n\langle i_1, i_2, n/2 \rangle)$.

Proposition 4.19 *Suppose $n, i_1, i_2 \in \mathbb{N}$ where $n \geq 8$ is even and $i_1 < i_2 < n/2$. Let $g = \gcd(i_1, i_2, n)$.*

Then

$$\chi_0^\Delta(C_n\langle i_1, i_2, n/2 \rangle) = \begin{cases} 2 & \text{if } (n \text{ is not divisible by } 4 \text{ and both } i_1, i_2 \text{ are odd}), \\ 4 & \text{if } (4|n \text{ and } i_1 + i_2 = n/2), \\ & \text{or if } (i_1 = n/4 \text{ or } i_2 = n/4), \\ & \text{or if } (g = 1 \text{ or } n \text{ is not divisible by } 6, i_2 = 2i_1 \text{ and } C_n\langle i_1, i_2, n/2 \rangle \\ & \text{is not isomorphic to } C_{10}\langle 2, 4, 5 \rangle \text{ or } C_6\langle 1, 2, 3 \rangle), \\ & \text{or if } (n = 16, 24, 28 \text{ and } i_2 = 5i_1) \\ & \text{or if } (n = 26, g = 2 \text{ and } i_2 = 5i_1) \\ & \text{or if } (n = 20 \text{ and } i_2 = 6i_1) \\ & \text{or if } (n = 22, g = 1 \text{ and } i_2 = 8i_1) \\ & \text{or if } (n = 28 \text{ and } i_2 = 8i_1) \\ & \text{or if } (n = 40 \text{ and } i_2 = 11i_1) \\ 5 & \text{if } n = 10 \text{ and } i_1 = 2, i_2 = 4, \\ & \text{or if } (n \text{ is not divisible by } 4 \text{ and } i_1 + i_2 = n/2), \\ 3 & \text{otherwise,} \end{cases}$$

where $C_n\langle i_1, i_2, n/2 \rangle$ denotes a connected composite circulant of order n with connection set $\{i_1, i_2, n/2\}$, where n is even. ■

Propositions 4.18 and 4.19 do not include the case of determining $\chi_0^\Delta(C_n\langle i_1, i_2, i_3 \rangle)$ where $i_3 \neq i_1 + i_2$ or $i_3 \neq n/2$; yet a large number of cases have already emerged. The problem of determining $\chi_d^\Delta(C_n\langle i_1, i_2, i_3 \rangle)$ for all $d \geq 1$ is thus left open — it is anticipated that even a larger number of cases will emerge. To the best knowledge of the authors of Propositions 4.18 and 4.19 the case of determining the classical chromatic number of circulants with connection set of cardinality four has as yet not been solved [81]. Therefore, the following result for a general circulant is the only one obtained in this study of maximum degree chromatic numbers.

Proposition 4.20 *Suppose $n, z, i_1, \dots, i_z \in \mathbb{N}$ where $n \geq 6$ and $i_1 < i_2 < \dots < i_z \leq n/2$. Let $C_n\langle i_1, \dots, i_z \rangle$ denote a composite circulant of order n with connection set $\{i_1, \dots, i_z\}$. If $C_n\langle i_1, \dots, i_z \rangle$ is singular, then*

$$\chi_{2z-1}^\Delta(C_n\langle i_1, \dots, i_z \rangle) = 1 \quad \text{and} \quad D_1^\Delta(C_n\langle i_1, \dots, i_z \rangle) = 2z - 1,$$

while if $C_n\langle i_1, \dots, i_z \rangle$ is non-singular, then

$$\chi_{2z}^\Delta(C_n\langle i_1, \dots, i_z \rangle) = 1 \quad \text{and} \quad D_1^\Delta(C_n\langle i_1, \dots, i_z \rangle) = 2z.$$

Proof: Since, from §2.1, a singular composite circulant is $(2z-1)$ -regular and a non-singular composite circulant is $2z$ -regular, one colour suffices if $d = 2z - 1$ and $d = 2z$ respectively. ■

This section on the maximum degree chromatic number of a circulant is concluded with the following result for the dense composite circulant, $C_n\langle 2, \dots, n/2 \rangle$, of order n with connection set $\{2, \dots, n/2\}$ where $n \in \mathbb{N}$ and $n \geq 4$. Note that $C_n\langle 2, \dots, n/2 \rangle = \overline{C}_n$, the complement of the cycle of order n .

Proposition 4.21 *Suppose $d \in \mathbb{N}_0$, $x, n \in \mathbb{N}$ where $n \geq 4$. Let $\overline{C}_n = C_n\langle 2, \dots, n/2 \rangle$ denote a composite circulant of order n with connection set $\{2, \dots, n/2\}$. Then*

$$\chi_d^\Delta(\overline{C}_n) = \begin{cases} 1 & \text{if } d \geq n - 3, \\ \left\lceil \frac{n}{d+2} \right\rceil & \text{if } d < n - 3, \end{cases} \quad \text{and} \quad D_x^\Delta(\overline{C}_n) = \begin{cases} \left\lceil \frac{n}{x} \right\rceil - 2 & \text{if } 2 \leq x \leq n - 1, \\ n - 3 & \text{if } x = 1. \end{cases}$$

Proof: Since $\Delta(\overline{C}_n) = n - 3$ it follows that $\chi_d^\Delta(\overline{C}_n) = 1$ if $d \geq n - 3$, because in this case all vertices of \overline{C}_n may be coloured with one colour.

For the remainder of this proof let $\{v_1, \dots, v_n\}$ denote the vertices of \overline{C}_n . Furthermore, for all $i = 2, \dots, n - 1$, let v_i be the vertex nonadjacent to v_{i-1} and v_{i+1} , and adjacent to all other vertices. Let

v_1 be nonadjacent to v_n and v_2 , and adjacent to v_3, \dots, v_{n-1} , and finally let v_n be nonadjacent to v_{n-1} and v_1 , and adjacent to v_2, \dots, v_{n-2} . Denote a partition of $V(\overline{C}_n)$ into the x colour classes of a $\Delta(d, x)$ -colouring of \overline{C}_n by $\mathcal{C}_1, \dots, \mathcal{C}_x$. Then x may be minimized by maximizing $|\mathcal{C}_j|$ for all $j = 1, \dots, x$ and a given $d \in \mathbb{N}_0$. For a given $d \in \mathbb{N}_0$, suppose the colour class \mathcal{C}_k for some $k \in \{1, \dots, x\}$ is formed by colouring all the vertices $v_i, v_{i+1}, \dots, v_{i+d+1}$, for some $i = 1, \dots, n - d - 1$, with colour k . Then $|\mathcal{C}_k| = d + 2$ and $\deg_{\langle \mathcal{C}_k \rangle}(v_j) = d - 1$ for all $j \in \{i + 1, \dots, i + d\}$, and $\deg_{\langle \mathcal{C}_k \rangle}(v_i) = \deg_{\langle \mathcal{C}_k \rangle}(v_{i+d+1}) = d$. Since each vertex is nonadjacent to two other vertices only, it is impossible to colour $d + 2$ vertices such that the colour class induced maximum degree is $d - 1$. The colouring described above is thus best possible and since $|\mathcal{C}_k| = d + 2$, it follows that $\chi_d^\Delta(\overline{C}_n) \geq \lceil n/(d + 2) \rceil$. But a $\Delta(d, x)$ -colouring of \overline{C}_n may be constructed for which $x = \lceil n/(d + 2) \rceil$ by partitioning $V(\overline{C}_n)$ sequentially into colour classes of cardinality $d + 2$ as described above (the last colour class may possibly have a cardinality smaller than $d + 2$). The result now follows from the fact that $\chi_d^\Delta(\overline{C}_n) \leq x$. ■

Three examples of the colouring strategy in Proposition 4.21 for obtaining χ_d^Δ -colourings of the dense composite circulant, \overline{C}_n , are given in Figure 4.14.

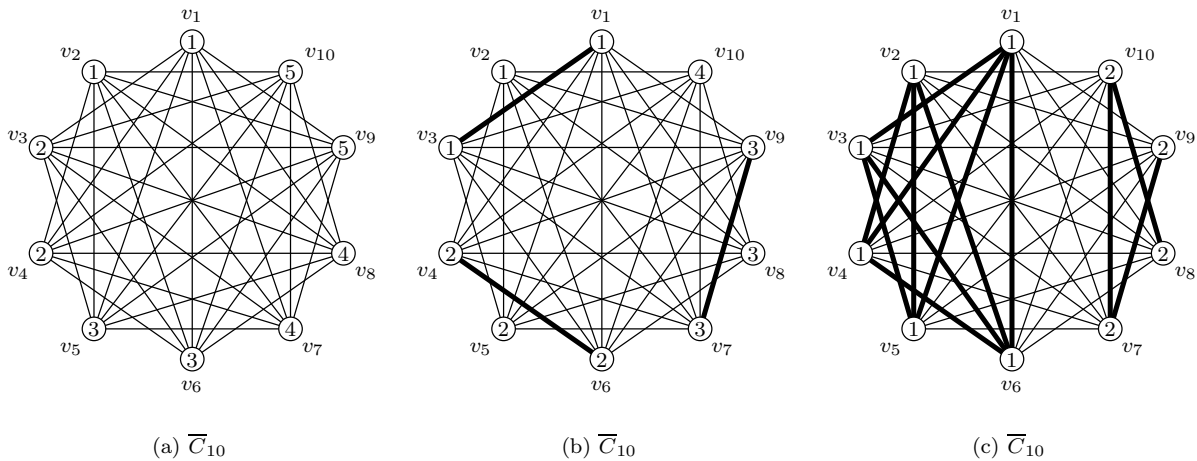


FIGURE 4.14: Examples of χ_d^Δ -colourings of \overline{C}_{10} . The bold edges indicate the colour-induced subgraphs formed during each χ_d^Δ -colouring of \overline{C}_{10} . (a) A χ_0^Δ -colouring of \overline{C}_{10} , showing that $\chi_0^\Delta(\overline{C}_{10}) = 5$, (b) a χ_1^Δ -colouring of \overline{C}_{10} , showing that $\chi_1^\Delta(\overline{C}_{10}) = 4$ and (c) a χ_4^Δ -colouring of \overline{C}_{10} , showing that $\chi_4^\Delta(\overline{C}_{10}) = 2$.

4.6 Complete Balanced Multipartite Graphs

The $\Delta(d)$ -chromatic number of the complete balanced multipartite graph consisting of k partite sets, each of cardinality n , is determined in this section by its related inversion number $D_x^\Delta(K_{k \times n})$ as described in §3.3. For the sake of simplicity, the complete balanced multipartite graph $K_{k \times n}$ will also be represented graphically in a specific manner throughout this section. In this representation, all vertices in the same partite set while be placed in a row as illustrated in Figure 4.15 for the two complete, balanced multipartite graphs $K_{2 \times 3}$ and $K_{3 \times 2}$, where Figure 4.15(b) is thus a different graphical representation of $K_{3 \times 2}$ than the one in Figure 2.8(c). The advantage of this representation is that the edges may be omitted from the graphical representation, because all vertices in the same row are nonadjacent, and any vertex, v , is adjacent to all the vertices in any row different from the row containing v .

From Corollary 4.1 it follows directly that if the number of colours, x , is at least 2, then $D_x^\Delta(K_{n,m}) = 0$. Similarly, for the complete balanced multipartite graph $K_{k \times n}$, if the number of colours, x , is equal to or greater than the number of partite sets, k , then $D_x^\Delta(K_{k \times n}) = 0$. Therefore, only the cases where $x < k$ are considered. Furthermore, the expectation is that in order to determine $D_x^\Delta(K_{k \times n})$, one only needs to find a $\Delta(d, x)$ -colouring of $K_{k \times n}$ such that the colour class induced maximum degrees are the same

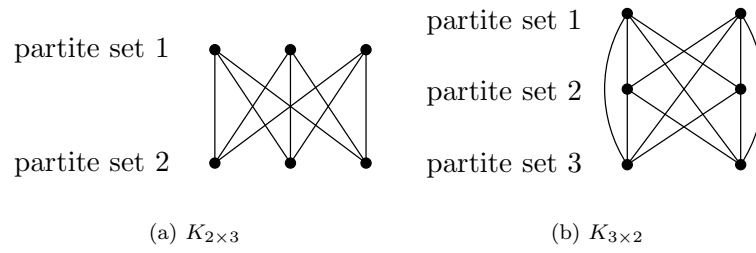


FIGURE 4.15: Graphical representations of two complete balanced multipartite graphs where the vertices of each partite set are placed in a row.

for all the colour classes. Unfortunately, this is *not* the case, as indicated by the example in Figure 4.16. In the colouring of $K_{6 \times 15}$ in Figure 4.16(a) all the colour class induced maximum degrees are the same, namely 15, while in the colouring of $K_{6 \times 15}$ in Figure 4.16(b) all the colour class induced maximum degrees are also the same, but much smaller than that in Figure 4.16(a).

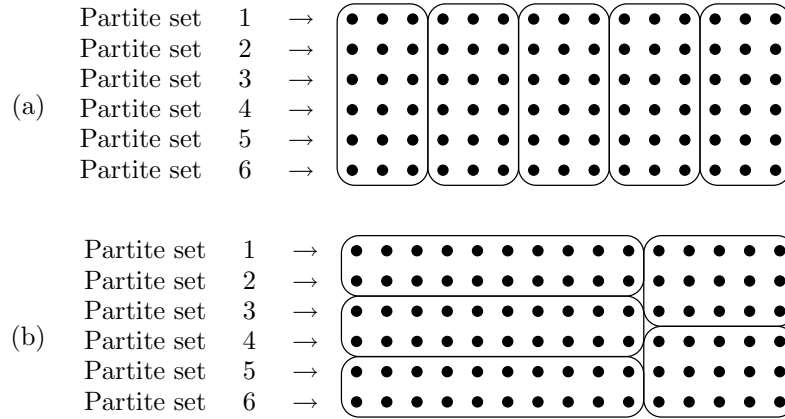


FIGURE 4.16: A conceptual representation of two colourings of $K_{6 \times 15}$. The dots in each row represent the vertices of a partite set of $K_{6 \times 15}$. A colour class is represented by means of a rectangular frame. In the colouring in (a) the colour class induced maximum degrees are all 15, while the colour class induced maximum degrees in (b) are all 10.

The strategy presented here for determining the parameter $D_x^\Delta(K_{k \times n})$ for the complete balanced multipartite graph, $K_{k \times n}$, comprises three parts. In the first part (§4.6.1), ideas from an article by Burger and Van Vuuren [23] were used to determine a normalized k -partite colouring. In the second part of the procedure (§4.6.2) this normalized k -partite colouring is used to derive an ideal $\Delta(d, x)$ -colouring of $K_{k \times n}$ — this colouring is ideal in the sense that fractions of vertices may be coloured. The remainder of §4.6.2 is devoted to the final part of the procedure which corrects this ideal situation by discretizing the results of the second part of the procedure (if necessary) so as to obtain a true $\Delta(d, x)$ -colouring of $K_{k \times n}$.

4.6.1 Normalized k -partite Colourings

The Δ -chromatic sequences of two complete balanced multipartite graphs with the same number of partite sets, namely $K_{4 \times 4}$ and $K_{4 \times 8}$, are shown in Example 4.1. The two Δ -chromatic sequences in Example 4.1 are similar in the sense that the number of times that any specific entry in the first sequence appears is double that in the second sequence. Therefore, the sequence of the related inversion numbers $\{D_x^\Delta(\bullet)\}_{x=1,2,3,\dots}$ for $K_{4 \times 8}$ is double the corresponding numbers for $K_{4 \times 4}$ as indicated by the two sequences in the second table in Example 4.1.

Example 4.1 The Δ -chromatic sequences of the graphs $K_{4 \times 4}$ and $K_{4 \times 8}$ are given below.

d	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...	23	24	...
$\chi_d^\Delta(K_{4 \times 4})$	4	4	4	3	2	2	2	2	2	2	2	2	1	1	...	1	1	...
$\chi_d^\Delta(K_{4 \times 8})$	4	4	4	4	4	4	3	3	2	2	2	2	2	2	...	2	1	...

The sequences of the related inversion numbers $\{D_x^\Delta(\bullet)\}_{x=1,2,3,\dots}$ for $K_{4 \times 4}$ and $K_{4 \times 8}$, given below, follow directly from the above two sequences.

x	1	2	3	4	5	...
$D_x^\Delta(K_{4 \times 4})$	12	4	3	0	0	...
$D_x^\Delta(K_{4 \times 8})$	24	8	6	0	0	...

Notice that the number of repetitions of each number in the Δ -chromatic sequence of $K_{4 \times 8}$ is exactly double the number of repetitions in the Δ -chromatic sequence of $K_{4 \times 4}$, so that the value of $D_x^\Delta(K_{4 \times 8})$ is exactly twice that of $D_x^\Delta(K_{4 \times 4})$ for all x . ■

Graphical representations of D_3^Δ -colourings of $K_{4 \times 4}$ and $K_{4 \times 8}$ are shown in Figure 4.17(a) and (b); notice that the colouring structures in Figures 4.17(a) and (b) are the same. From the figure it is clear that essentially the same problem is solved in both cases. It therefore makes sense to view the problem of determining $D_x^\Delta(K_{k \times n})$ in a continuous setting where each partite set is replaced by a line segment of unit length; hence disregarding the actual number of vertices in each partite set. In such a continuous setting colour classes originally comprising discrete entities (graph vertices) are replaced by collections of (continuous) line subsegments, called **normalized colour classes**, representing the original colour classes in a normalized sense. In this continuous setting the notion of a colour class induced maximum degree is replaced by that of a **normalized colour class induced maximum degree** — the sum total of the lengths of all subsegments in a normalized colour class, except for the shortest subsegment in that class. If the lengths of all subsegments in a normalized colour class are the same, this length is referred to as the **normalized width** of the normalized colour class. If x colours are used and if the largest normalized colour class induced maximum degree is $\tilde{d} \in \mathbb{R}$, then the resulting colouring is referred to as a **normalized $\Delta(\tilde{d}, x)$ k -partite colouring**. A graphical representation of such a normalized $\Delta(\frac{3}{4}, 3)$ 4-partite colouring is shown in Figure 4.17(c).

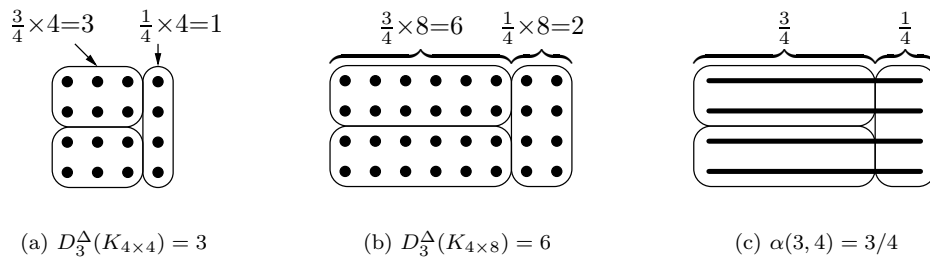


FIGURE 4.17: Graphical representations of D_3^Δ -colourings of (a) $K_{4 \times 4}$ and (b) $K_{4 \times 8}$ with the same colouring pattern. The dots in each row of (a) and (b) represent the vertices of a partite set of $K_{4 \times n}$, while a colour class is represented by means of a rectangular frame. In (c) a similarly structured normalized $\Delta(\frac{3}{4}, 3)$ 4-partite colouring is given where the partite sets are viewed as line segments of unit length.

Let $\alpha(x, k)$ denote the smallest value of \tilde{d} for which a normalized $\Delta(\tilde{d}, x)$ k -partite colouring exists; such an optimal colouring is called a **normalized k -partite D_x^Δ -colouring**. The continuous problem of determining $\alpha(x, k)$ seems to be a hard problem that has only been solved approximately by Burger and Van Vuuren [23], as summarised in Appendix D. Some of these results are stated here for continuity and are discussed in terms of the specific problem of searching for a normalized k -partite D_x^Δ -colouring. The following general bounds on $\alpha(x, k)$ are reproduced from Appendix D (D.2.1).

Proposition 4.22 $k/x - 1 \leq \alpha(x, k) \leq \lceil k/x \rceil - 1$. ■

From the above proposition it is clear that $\alpha(x, k) = k/x - 1$ when $x|k$; this is Case 1 in Table 4.1 (which is essentially the same as Table D.1). However, based on the divisibility properties of k by x , a further three cases arise, as shown in Table 4.1.

	$s = 0$	$0 < s < \lceil \frac{k}{x} \rceil - 1$	$s = \lceil \frac{k}{x} \rceil - 1$
$x' = 0$	Case 1 (Trivial) $\alpha(x, k) = \frac{k}{x} - 1$	Case 2 (Balancing) $\alpha(x, k) = \frac{(k-1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}$	Case 3 (Reduction) $\alpha(x, k) \leq \alpha(x-1, k-s)$
$x' > 0$	Case 4 (Composition) $\alpha(x, k) \leq \frac{1}{\frac{1}{\alpha(x-x', k)} + \frac{1}{\alpha(x', k)}}$		

TABLE 4.1: Values of and upper bounds on $\alpha(x, k)$, depending on the values of $s \equiv k \pmod{\lceil \frac{k}{x} \rceil}$ and $x' = x - \lceil k/\lceil \frac{k}{x} \rceil \rceil$.

For all four cases the strategy is to form $\lceil k/\lceil k/x \rceil \rceil$ normalized colour classes, each comprising $\lceil k/x \rceil$ full partite sets, except possibly for the last utilised colour class, which comprises $s \equiv k \pmod{\lceil k/x \rceil}$ full partite sets if $x \nmid k$ (in the case where $x|k$, all colour classes comprise the same number of partite sets). Upon completion of this partition of the partite sets, $x' = x - \lceil k/\lceil k/x \rceil \rceil$ colours remain unutilised. The values of s and x' are used as decision variables to distinguish between the cases in Table 4.1 in the recursive normalized $\Delta(\tilde{d}, x)$ k -partite colouring strategy given in pseudo-code as Algorithm $\bar{\alpha}(x, k)$ (which is the same as Algorithm 14 in Appendix D), which outputs an upper bound $\bar{\alpha}(x, k)$ on $\alpha(x, k)$.

Algorithm 6 Algorithm $\bar{\alpha}(x, k)$

Input: The number of colours x and the number of partite sets k .

Output: An upper bound $\bar{\alpha}(x, k)$ on $\alpha(x, k)$.

```

1: if  $x|k$  then
2:   stop return  $\frac{k}{x} - 1$ 
3: end if
4:  $x' \leftarrow x - \lceil k/\lceil \frac{k}{x} \rceil \rceil$ ,  $s \leftarrow k \pmod{\lceil \frac{k}{x} \rceil}$ 
5: if  $s = \lceil \frac{k}{x} \rceil - 1$  then
6:   return  $\bar{\alpha}(x-1, k-s)$ 
7: else
8:   if  $x' > 0$  then
9:     return  $1/(\frac{1}{\bar{\alpha}(x-x', k)} + \frac{1}{\bar{\alpha}(x', k)})$ 
10:  else
11:    stop return  $\frac{(k-1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}$ 
12:  end if
13: end if

```

It should be clear from the pseudo-code listing that Algorithm $\bar{\alpha}(x, k)$ functions directly according to the classification of cases in Table 4.1. The colouring substrategy and thus the values of $\bar{\alpha}(x, k)$ attained in each case are discussed separately.

Case 1 (Trivial) The case where $x' = 0$ and $s = 0$ represents a terminating call of Algorithm $\bar{\alpha}(x, k)$. If this is also the first call (*i.e.* if $x|k$), then a colouring of the form depicted in Figure 4.18(a)

results. Otherwise a colouring of this structure will be a subconstruction of the overall colouring strategy involving line subsegments of normalized width $y = \bar{\alpha}(x, k)/(\lceil k/x \rceil - 1)$, because in this case the normalized degree $\bar{\alpha}(x, k)$ of each colour class is the normalized width y multiplied by the number of line subsegments included per colour class ($\lceil k/x \rceil$), less one.

Case 2 (Balancing) In the case where $x' = 0$ and $0 < s < \lceil k/x \rceil - 1$ the normalized maximum degree of the last colour class is sufficiently smaller than that of the other colour classes to allow recolouring of a small proportion y' of each of the line segments of the other colour classes using the last colour, so as to increase the normalized maximum degree of the last colour class and decrease the normalized maximum degree of the other colour classes. The normalized width y' is determined such that the normalized maximum degree of the last colour class is the same as the normalized maximum degree of the other colour classes, *i.e.* such that $ys + (k - 1)y' = \bar{\alpha}(x, k)$. If y is calculated as in Case 1 and substituted in this expression, it follows that $y' = \bar{\alpha}(x, k)(\lceil k/x \rceil - s - 1)/((\lceil k/x \rceil - 1)(k - 1))$. This case is illustrated graphically in Figure 4.18(b), where the inverted, L-shaped frame at the bottom represents the last colour class after balancing.

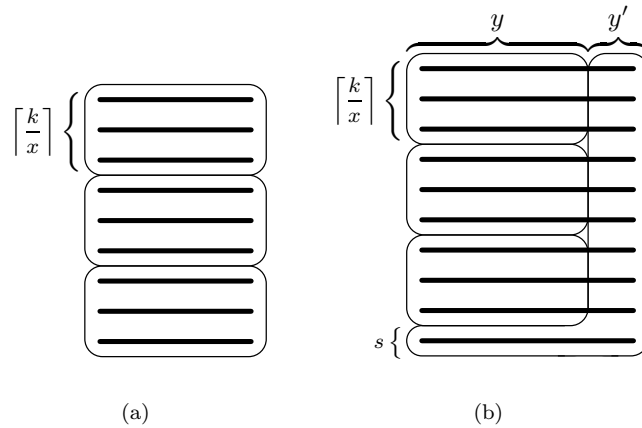


FIGURE 4.18: Graphical representation of the colouring approach in Algorithm $\bar{\alpha}(x, k)$ for (a) Case 1 and (b) Case 2, where a balancing operation is performed on the last colour class. As before, the line segments represent normalized partite sets and colour classes are represented by means of rectangular or inverted, L-shaped frames.

Case 3 (Reduction) In the case where $s = \lceil k/x \rceil - 1$ the last utilised colour class comprises exactly one partite set fewer than the other colour classes, and hence it is not possible to perform a balancing operation (as described in Case 2). In this case the partite sets comprising the last utilised colour class do not form part of future calls of the algorithm. This is equivalent to reducing the problem of determining $\bar{\alpha}(x, k)$ to a smaller problem, namely with one colour fewer and with $k - s$ partite sets. Notice that if $x' = 0$, the next recursive call of Algorithm $\bar{\alpha}(x, k)$ will be the trivial case (Case 1). However, if $x' > 0$, then the next recursive call will be a composition (Case 4 described below). This case is illustrated graphically in conjunction with the case of composition in Figure 4.19.

Case 4 (Composition) If $x' > 0$ and $0 \leq s < \lceil \frac{k}{x} \rceil - 1$, then two problems are solved separately and the results obtained are combined by means of the composition formula given in Case 4 of Table 4.1. All partite sets are divided in two portions, one of length u_1 and one of length u_2 , such that $u_1 + u_2 = 1$. Two normalized $\Delta(\tilde{d}, x)$ k -partite colouring strategies are then combined by using respectively x' and $x - x'$ colours on two collections of shorter k -partite sets — the values of u_1 and u_2 are chosen so that $u_1 \bar{\alpha}(x - x', k) = u_2 \bar{\alpha}(x', k)$. Note that the one problem (resulting in the bound $\bar{\alpha}(x - x', k)$) always terminates immediately either as the trivial or as the balancing case. The other problem (resulting in the bound $\bar{\alpha}(x', k)$) may call any of the four sub-construction cases, and therefore may or may not terminate immediately. This case is illustrated graphically in conjunction with the case of reduction in Figure 4.19.

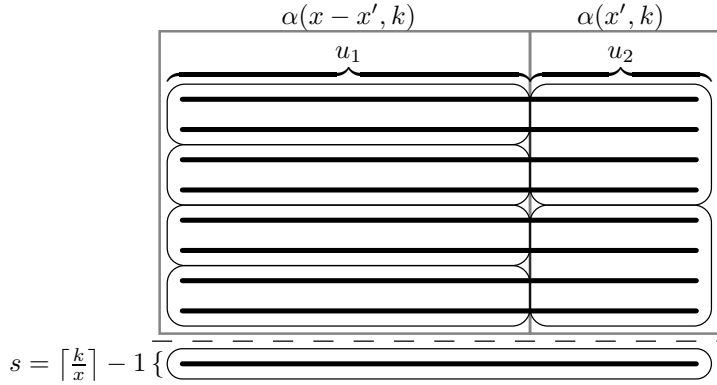


FIGURE 4.19: Graphical representation of the output of Algorithm $\bar{\alpha}(x, k)$ if the first call is a reduction call (Case 3) and the second call is a composition call (Case 4) for which both recursive subcalls are terminating.

The progress of Algorithm $\bar{\alpha}(x, k)$ may be captured by the construction of a so-called **progress tree** in which vertices represent algorithm calls. All vertices in this binary tree may be classified as one of three types, namely (a) a **terminating vertex** of degree 1 corresponding to a Case 1 or 2 terminating call in Table 4.1, (b) a **reduction vertex** of degree 2 corresponding to a Case 3 non-terminating call and (c) a **composition vertex** of degree 3 corresponding to a Case 4 non-terminating call. These types of vertices are shown graphically in Figure 4.20. Note that because the call $\bar{\alpha}(x - x', k)$ following a Case 4 algorithmic call is necessarily a terminating call, the corresponding descendant of the composition vertex will be a terminating vertex — the convention will be to place this descendant on the left. Furthermore, the terminating vertices in the progress tree are numbered in increasing order first from the top level in the tree downwards and from the left to the right within each level.

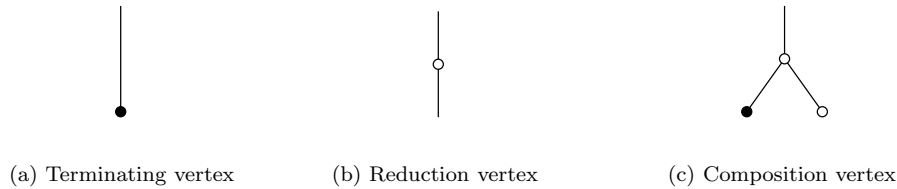


FIGURE 4.20: Graphical representations of the types of vertices in the progress tree of Algorithm $\bar{\alpha}(x, k)$.

Suppose a total of $\ell \in \mathbb{N}$ terminating calls are made during the execution of Algorithm $\bar{\alpha}(x, k)$ and denote the values of x , x' , k and s during the j -th terminating call by x_j , x'_j , k_j and s_j respectively ($j = 1, \dots, \ell$). Furthermore, let

$$a_j = \left\lfloor \frac{k_j}{x_j} \right\rfloor - 1, \quad j = 1, \dots, \ell. \quad (4.6.1)$$

Then $a_j + 1$ is the number of line subsegments in each colour class (except possibly for the last colour class) during the j -th terminating call of Algorithm $\bar{\alpha}(x, k)$. Finally, let y_j be the value of y at the j -th terminating call (as described in Cases 1 and 2), and if the j -th terminating call is a Case 2 (balancing) call, then let y'_j be the value of y' during that call. The working of Algorithm $\bar{\alpha}(x, k)$ is illustrated by means of a simple example.

Example 4.2 Suppose a normalized 11-partite D_{10}^Δ -colouring is sought. The values of the variables x , k , s and x' during each recursive call of Algorithm $\bar{\alpha}(x, k)$ are given in Figure 4.21. A total of four calls of Algorithm $\bar{\alpha}(x, k)$ are made, two of which are terminal, as may be seen in Figure 4.21. It is concluded that $\alpha(10, 11) \leq \bar{\alpha}(10, 11) = 1/(1 + 1/(9/5)) = 9/14$. A graphical representation of a normalized $\Delta(\tilde{d}, 10)$ 11-partite colouring achieving this bound is shown in Figure 4.22. ■

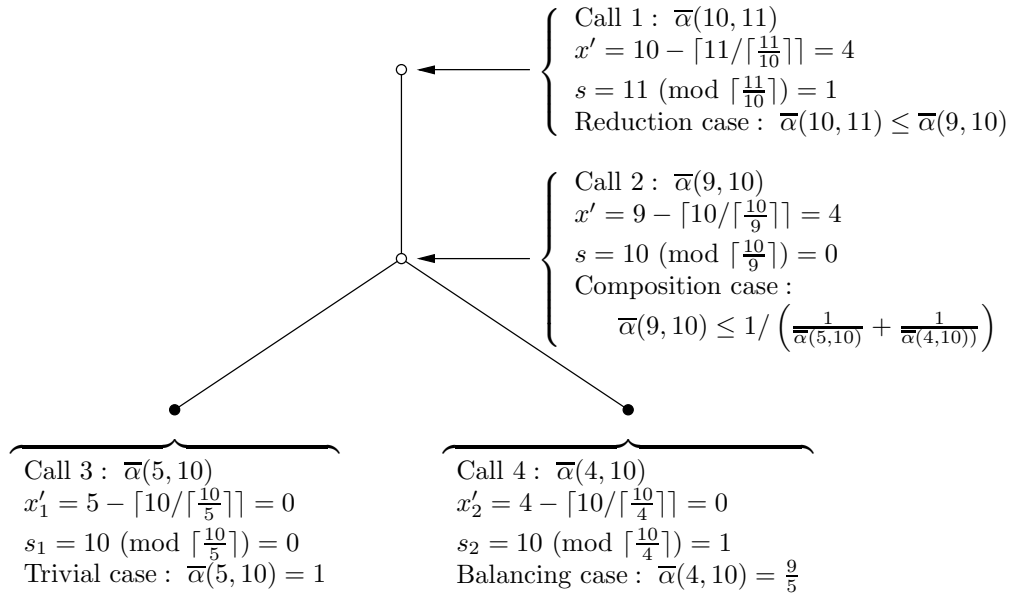


FIGURE 4.21: The progress tree of Algorithm $\bar{\alpha}(x, k)$ for Example 4.2 where a normalized 11-partite D_{10}^Δ -colouring is sought.

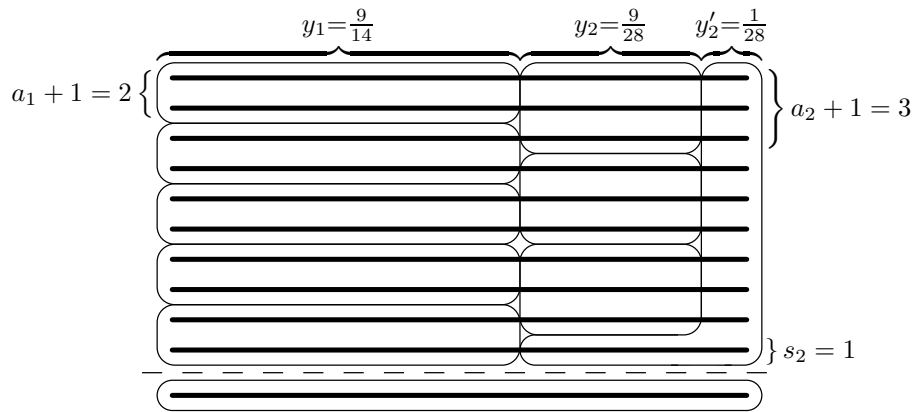


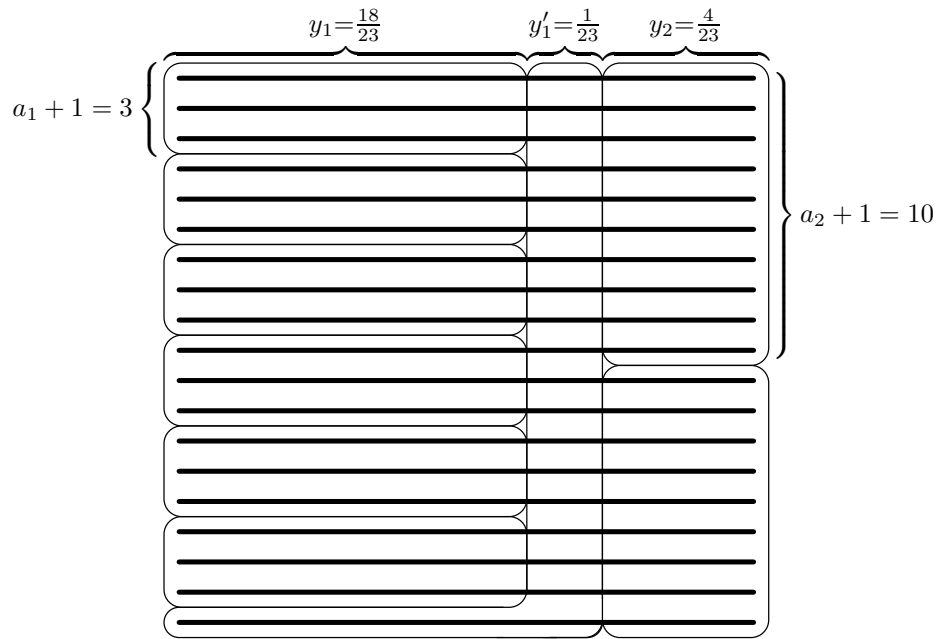
FIGURE 4.22: Graphical representation of a normalized $\Delta(\tilde{d}, 10)$ 11-partite colouring with a normalized colour class induced maximum degree of $\bar{\alpha}(10, 11) = 9/14$.

It is believed that Algorithm $\bar{\alpha}(x, k)$ produces good upper bounds. These bounds are, in fact, exact when only one recursive call is made. However, the upper bound produced by the algorithm is not optimal in general — there are rare cases where it is possible to improve slightly upon the construction produced by the algorithm, as illustrated in the following example.

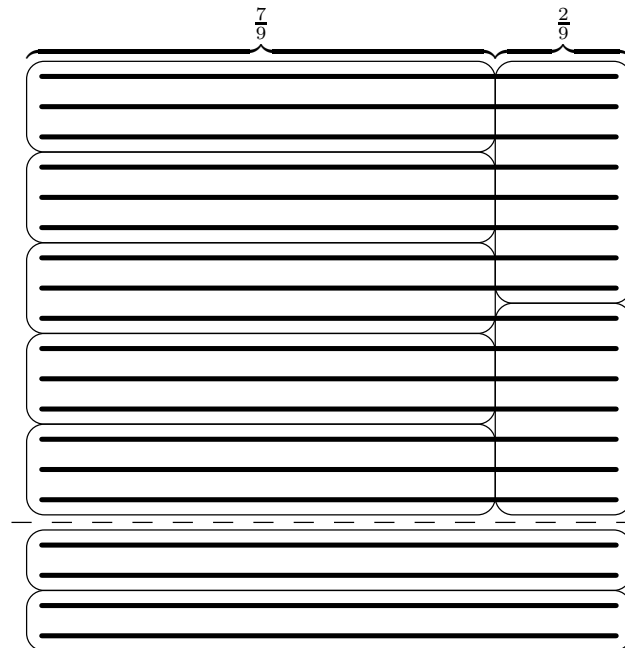
Example 4.3 Suppose a normalized 19-partite D_9^Δ -colouring is sought. The construction produced by Algorithm $\bar{\alpha}(x, k)$, which results in the bound $\alpha(9, 19) \leq \bar{\alpha}(9, 19) = (\bar{\alpha}(7, 19)^{-1} + \bar{\alpha}(2, 19)^{-1})^{-1} = (19/36 + 1/9)^{-1} = 36/23$, is shown in Figure 4.23(a).

However, an alternative normalized $\Delta(\tilde{d}, 9)$ 19-partite colouring corresponding to the bound $\alpha(9, 19) \leq \bar{\alpha}(7, 15) = (\bar{\alpha}(5, 15)^{-1} + \bar{\alpha}(2, 15)^{-1})^{-1} = (1/2 + 1/7)^{-1} = 14/9$ is shown in Figure 4.23(b). The latter colouring represents an improvement of $2/207$ over the upper bound on $\alpha(9, 19)$ established by the colouring produced by Algorithm $\bar{\alpha}(x, k)$. ■

As is evident from the above example, it is sometimes more beneficial to perform a “double reduction” instead of performing a balancing operation as dictated by Algorithm $\bar{\alpha}(x, k)$. It is not clear exactly when



(a)



(b)

FIGURE 4.23: A graphical representation of one of the rare cases, where $x = 9$ and $k = 19$, for which the colouring approach in Algorithm $\bar{\alpha}(x, k)$ may be improved upon slightly. In (a) the colouring corresponding to Algorithm $\bar{\alpha}(x, k)$ is given for which $\bar{\alpha}(9, 19) = 36/23$. A different colouring construction, where a “double reduction” is performed instead of a balancing subconstruction, is given in (b), resulting in the bound $\alpha(9, 19) \leq 14/9$.

this scenario occurs, but the smallest value of k for which this phenomenon occurs is $(x, k) = (9, 19)$. Other known cases are $(7, 22)$, $(10, 22)$ and $(12, 25)$.

Algorithm $\bar{\alpha}(x, k)$ exhibits a worst-case time complexity of $\mathcal{O}(\log \log x)$ (see Appendix D).

4.6.2 Discrete Colourings for $K_{k \times n}$

In this section the objective is to establish an upper bound $\overline{D}_x^\Delta(K_{k \times n})$ on $D_x^\Delta(K_{k \times n})$ for various values of n in terms of the upper bound $\bar{\alpha}(x, k)$ on $\alpha(x, k)$ determined in the previous section. Note that, if $y_j n$ and $y'_j n$ are integers for all $j = 1, \dots, \ell$, then $\bar{\alpha}(x, k)n$ is of course an upper bound on $D_x^\Delta(K_{k \times n})$, as illustrated by means of a numerical example below. However, in other cases the (continuous) colouring strategy of the previous section has to be discretized in order to avoid the situation where fractions of vertices are coloured. In cases where fractions of vertices are coloured, the colouring is referred to as an **ideal $\Delta(d, x)$ -colouring** with an **ideal colour class induced maximum degree**.

Example 4.4 In Example 4.2 a normalized 11-partite D_{10}^Δ -colouring was sought. The bound $\bar{\alpha}(10, 11) = 9/14$ was determined by means of Algorithm $\bar{\alpha}(x, k)$. It may be verified that $y_1 = 9/14$, $y_2 = 9/28$ and $y'_2 = 1/28$ in this case.

(a) Suppose $n = 28$. Then $y_1 n = 18$, $y_2 n = 9$ and $y'_2 n = 1$ are integral. Thus, $\overline{D}_{10}^\Delta(K_{11 \times 28}) = \bar{\alpha}(10, 11)n = 18$ is an appropriate upper bound on $D_{10}^\Delta(K_{11 \times 28})$, as witnessed by the $\Delta(18, 10)$ -colouring of $K_{11 \times 28}$ shown graphically in Figure 4.24.

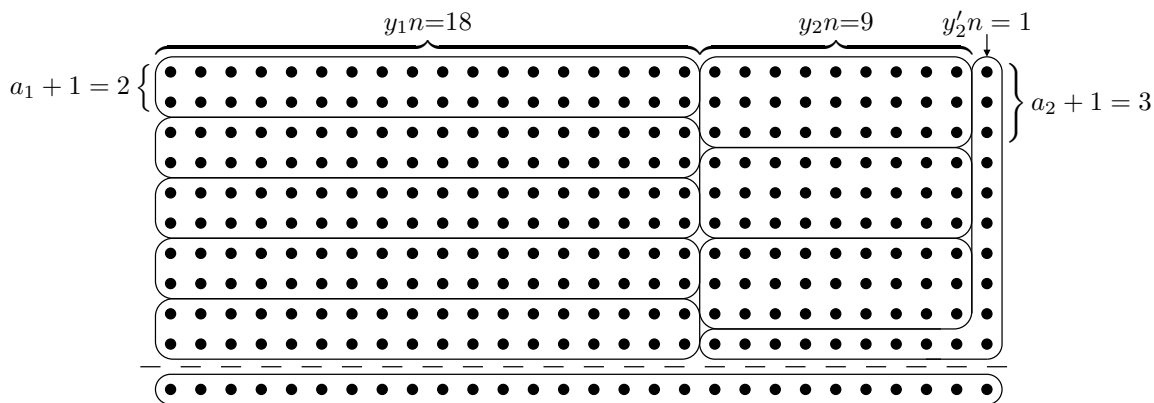


FIGURE 4.24: Graphical representation of a $\Delta(18, 10)$ -colouring of $K_{11 \times 28}$.

(b) Suppose now $n = 14$. Then $y_1 n = 9$, $y_2 n = 9/2$ and $y'_2 n = 1/2$ are not all integral. Thus, an upper bound on $D_{10}^\Delta(K_{11 \times 14})$ remains to be determined. However, a graphical representation of an ideal $\Delta(9, 10)$ -colouring of $K_{11 \times 14}$ may be found in Figure 4.25. ■

The values of $y_2 n$ and $y'_2 n$ in Example 4.4(b) are not integral; hence fractions of vertices may be found in some colour classes if the graph were to be coloured according to the colour structure shown in Figure 4.25. When the colouring is discretized, $\bar{\alpha}(x, k)n$ may not necessarily be an upper bound on $D_x^\Delta(K_{k \times n})$. One possible strategy to determine an upper bound on $D_x^\Delta(K_{k \times n})$ would be to successively attempt achieving colour class induced maximum degrees $\lceil \bar{\alpha}(x, k)n \rceil$, $\lceil \bar{\alpha}(x, k)n \rceil + 1$, $\lceil \bar{\alpha}(x, k)n \rceil + 2, \dots$ using x colours, until a value \overline{D} is found for which no colour class induced maximum degree exceeds \overline{D} . In the remainder of this section it is shown that if a value of $\lceil \bar{\alpha}(x, k)n \rceil$ for \overline{D} does not suffice, then $\overline{D} = \lceil \bar{\alpha}(x, k)n \rceil + 1$ will certainly suffice. However, $\overline{D} = \lceil \bar{\alpha}(x, k)n \rceil$ seems to be sufficient in most cases.

In a bid to formalise the discretization process some terminology and notation is introduced. The normalized width y_j is scaled with a factor n to arrive at the **ideal width** $y_j n$ (where fractional vertex colouring is allowed). However, if the ideal colour class induced maximum degree, $\bar{\alpha}(x, k)n$, is discretized

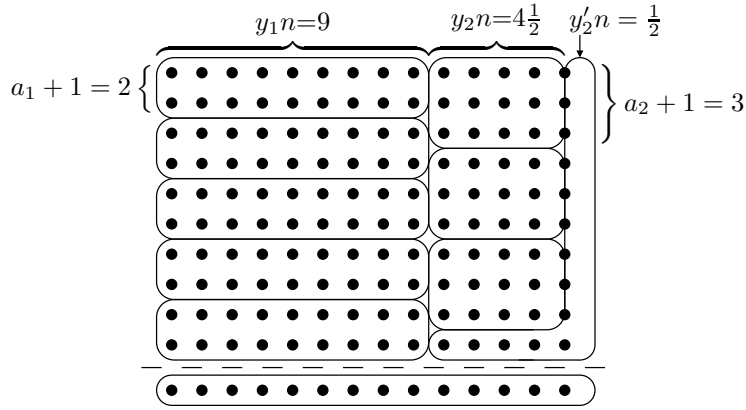


FIGURE 4.25: Graphical representation of an ideal $\Delta(9, 10)$ -colouring of $K_{11 \times 14}$.

to \bar{D} , the ideal width associated with the degree of that colour class is potentially increased. This width is called the **ideal width associated with degree \bar{D}** . Let z_j represent the integer part of the ideal width associated with degree \bar{D} . A total of t_j vertices (at most one vertex per partite set) may then be added to each colour class formed during the j -th terminating call of Algorithm $\bar{\alpha}(x, k)$ such that $\bar{D} = a_j z_j + t_j$, resulting in the desired maximum degree for the colour class. See Figure 4.26(a) for a graphical illustration of the meaning of the parameters z_j and t_j . If balancing occurs, the value of z_j for the last colour class is denoted by z'_j . The parameter t'_j has a similar definition, as illustrated in Figure 4.26(b).

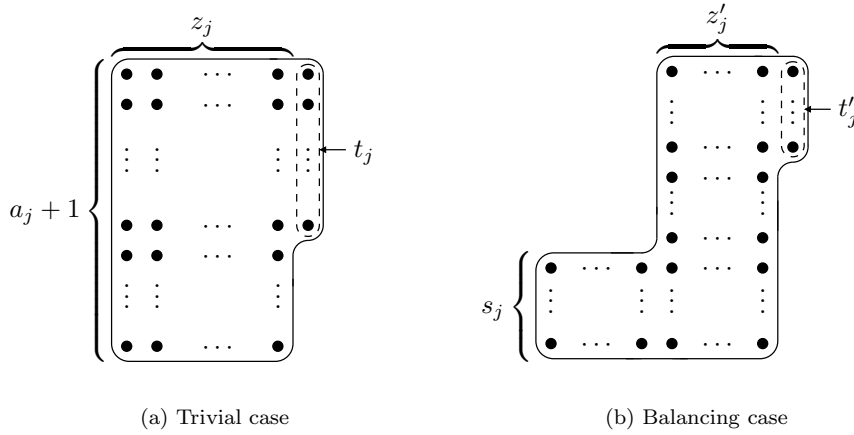


FIGURE 4.26: Graphical representation of the discretization process.

It is easy to verify that $z_j = \lfloor \bar{D}/a_j \rfloor$. Furthermore, $t_j = \bar{D} - a_j \lfloor \bar{D}/a_j \rfloor$. (Note that t_j cannot have the value a_j .) Let W_j denote the **average discretized width** of each colour class at the j -th terminating call of Algorithm $\bar{\alpha}(x, k)$. Then $W_j = \lfloor \bar{D}/a_j \rfloor + t_j/(a_j + 1) = (\lfloor \bar{D}/a_j \rfloor + \bar{D})/(a_j + 1)$ is an approximation of the ideal width \bar{D}/a_j associated with the degree \bar{D} . Let \tilde{e}_j denote the error in this approximation. It is easy to see that this error is at most $1/(a_j + 1)$, *i.e.*

$$0 \leq \underbrace{\frac{\bar{D}}{a_j}}_{\tilde{e}_j} - W_j \leq \frac{1}{a_j + 1}. \quad (4.6.2)$$

Let e_j denote the error incurred when the colour classes with ideal width $y_j n$ are discretized, *i.e.* $y_j n = W_j + e_j$.

Similarly, for the last colour class formed during the j -th terminating call of Algorithm $\bar{\alpha}(x, k)$ in the case where balancing is performed, hereafter referred to as a **balancing colour class** (see Figure 4.26(b)), $z'_j = \lfloor (\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor) / (k_j - 1) \rfloor$ and $t'_j = \bar{D} - (z'_j s_j + z'_j (k_j - 1))$ are obtained. Following the same procedure as for the case above, the approximation

$$W'_j = \left\lfloor \frac{\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j - 1} \right\rfloor + \frac{t'_j}{k_j} = \frac{\lfloor \frac{\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j - 1} \rfloor + \bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j}$$

for the ideal width $(\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor) / (k_j - 1)$ of the balancing colour class associated with the degree $\bar{D} - s_j z_j$, is obtained. (Note that the vertices in the balancing class associated with the width $y_j n$ are subtracted.) In this case the error \tilde{e}'_j involved in the approximation satisfies

$$0 \leq \underbrace{\frac{\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j - 1} - W'_j}_{\tilde{e}'_j} \leq \frac{1}{k_j}. \quad (4.6.3)$$

Finally, let e'_j be the error incurred when the balancing colour class with ideal width $y'_j n$ is discretized, *i.e.* $y'_j n = W'_j + e'_j$. Note $W'_j = e'_j = y'_j = 0$ if there is no balancing during the j -th terminating call of Algorithm $\bar{\alpha}(x, k)$.

In order to establish bounds on $D_x^\Delta(K_{k \times n})$, the following growth property of the sequence a_1, \dots, a_ℓ computed in (4.6.1), for which a proof may be found in [23, Lemma 5], is required.

Lemma 4.3 *The sequence $(a_j)_{j=1}^\ell$ generated in (4.6.1) satisfies the recursive relationship $a_{j+1} \geq a_j(a_j + 1)$, for all $j = 1, \dots, \ell - 1$.*

The results above may now be used to establish bounds on $D_x^\Delta(K_{k \times n})$.

Theorem 4.1 $\lceil \alpha(x, k)n \rceil \leq \bar{D}_x^\Delta(K_{k \times n}) \leq \lceil \bar{\alpha}(x, k)n \rceil + 1$.

Proof: The lower bound follows from the fact that $\alpha(x, k)$ is directly proportional to the length of the partite sets, *i.e.* if the length of the partite sets is changed by a factor n , then $\alpha(x, k)$ also changes by a factor n .

Let $\bar{D} = \lceil \bar{\alpha}(x, k)n \rceil + 1$ and let W_j and W'_j be defined as before. To prove that all colour classes have degree at most \bar{D} , it is required to show that $\sum_{j=1}^\ell (W_j + W'_j) \geq n$. However, $\sum_{j=1}^\ell (W_j + W'_j) = \sum_{j=1}^\ell (y_j n - e_j + y'_j n - e'_j) = n - \sum_{j=1}^\ell (e_j + e'_j)$. So, to complete the proof, it need only be shown that $\sum_{j=1}^\ell (e_j + e'_j) \leq 0$.

To achieve this goal, it is first shown that $e_j \leq 1/(a_j + 1) - 1/a_j$. The difference between the ideal width and the ideal width associated with degree \bar{D} , is $y_j n - \bar{D}/a_j = y_j n - (\lceil a_j y_j n \rceil + 1)/a_j \leq y_j n - (a_j y_j n + 1)/a_j = -1/a_j$. If this inequality is added to (4.6.2) the result is $e_j = y_j n - W_j \leq 1/(a_j + 1) - 1/a_j$, as desired.

Next, the inequality $e'_j \leq 1/k_\ell \leq 1/(2(a_\ell + 1))$ is proved. The first inequality follows from (4.6.3). The second inequality follows from the fact that at least three colour classes are required when balancing occurs. (If $x = 2$, the number of partite sets in each colour class will differ by at most 1). Thus, $\sum_{j=1}^\ell (e_j + e'_j) \leq \sum_{j=1}^\ell (1/(a_j + 1) - 1/a_j + 1/(2(a_\ell + 1)))$. For $\ell \geq 2$ it is not difficult to verify that the above expression is negative, as desired. (The identity $1/a_j - 1/(a_j + 1) = 1/(a_j(a_j + 1))$ and Lemma 4.3 may be used.) More care is needed for the case $\ell = 1$. It is sufficient to show that $W_1 + W'_1 \geq n$ for the case where there is only one balancing terminating call, since without balancing, $e_1 \leq 0$ and the theorem holds. In the remainder of the proof, all subscripts are omitted for the sake of convenience, since they are all 1. Note that $W' = (\lfloor (\bar{D} - sn)/(k - s - 1) \rfloor + \bar{D} - sn)/(k - s)$, because the discretization of the last s partite sets is not required. Thus,

$$W + W' = \frac{\lfloor \frac{\bar{D}}{a} \rfloor + \bar{D}}{a + 1} + \frac{\lfloor \frac{\bar{D} - sn}{k - s - 1} \rfloor + \bar{D} - sn}{k - s} \geq n$$

must be shown to hold true. Writing $k - s$ as $(x - 1)(a + 1)$ and multiplying with $(x - 1)(a + 1)$ the inequality

$$\left\lfloor \frac{\overline{D}}{a} \right\rfloor (x - 1) + \left\lfloor \frac{\overline{D} - sn}{k - s - 1} \right\rfloor + \overline{D}x \geq kn \quad (4.6.4)$$

is obtained (after simplification). Because all x colour classes have normalized colour class induced maximum degree $\alpha(x, k)$ in the normalized $\Delta(\overline{d}, x)$ k -partite colouring, $\alpha(x, k)$ may be written as $\alpha(x, k) = (k - y(x - 1) - y')/x$ or $yn(x - 1) + y'n + \alpha(x, k)nx = kn$. Substituting this into (4.6.4), it has to be shown that

$$\left(\left\lfloor \frac{\overline{D}}{a} \right\rfloor - yn \right) (x - 1) + \left\lfloor \frac{\overline{D} - sn}{k - s - 1} \right\rfloor - y'n + (\overline{D} - \alpha(x, k)n)x \geq 0. \quad (4.6.5)$$

As before, $\overline{D}/a - yn \geq 1/a$. Therefore, $\lfloor \overline{D}/a \rfloor - yn > -1$. It is also known that $\alpha(x, k) = y'(k - s - 1) + s = ay$, so that $((ay - s)n)/(k - s - 1) = y'n$. This implies that $(\overline{D} - sn)/(k - s - 1) > y'n$ or $\lfloor (\overline{D} - sn)/(k - s - 1) \rfloor - y'n > -1$. Also, $\overline{D} - \alpha(x, k)n \geq 1$. Substituting these three inequalities into (4.6.5) it follows that

$$\left(\left\lfloor \frac{\overline{D}}{a} \right\rfloor - yn \right) (x - 1) + \left\lfloor \frac{\overline{D} - sn}{k - s - 1} \right\rfloor - y'n + (\overline{D} - \alpha(x, k)n)x \geq (-1)(x - 1) + (-1) + (1)x = 0$$

which completes the proof. \blacksquare

The algorithm presented next (which should be thought of as a follow-up to Algorithm $\overline{\alpha}(x, k)$ of §4.6.1), is capable of determining whether $D_x^\Delta(K_{k \times n}) \leq \lceil \overline{\alpha}(x, k)n \rceil$, or whether the upper bound in Theorem 4.1 is required.

Algorithm 7 Algorithm $\overline{D}_x^\Delta(K_{k \times n})$

Input: The cardinality of each partite set, n , the upper bound, $\overline{\alpha}(x, k)$, the sequences k_1, \dots, k_ℓ and s_1, \dots, s_ℓ determined during the course of Algorithm $\overline{\alpha}(x, k)$, as well as the sequences a_1, \dots, a_ℓ , y_1, \dots, y_ℓ and y'_1, \dots, y'_ℓ .

Output: An upper bound $\overline{D}_x^\Delta(K_{k \times n})$ on $D_x^\Delta(K_{k \times n})$.

```

1:  $\overline{D} \leftarrow \lceil \overline{\alpha}n \rceil$ 
2: if  $y_j n, y'_j n \in \mathbb{N}$  for all  $j = 1, \dots, \ell$  then
3:   stop return  $\overline{D}$ 
4: end if
5: for all  $j = 1, \dots, \ell$  do
6:    $W_j \leftarrow (\lfloor \overline{D}/a_j \rfloor + \overline{D})/(a_j + 1)$ 
7:   if  $\lfloor \overline{D}/a_j \rfloor = 0$  then
8:      $W_j \leftarrow W_j + 1/(a_j + 1)$ 
9:   end if
10:  if  $0 < s_j < a_j$  then
11:     $W'_j \leftarrow (\lfloor (\overline{D} - \lfloor \overline{D}/a_j \rfloor)/(k_j - 1) \rfloor + \overline{D} - s_j \lfloor \overline{D}/a_j \rfloor)/k_j$ 
12:    if  $\lfloor (\overline{D} - \lfloor \overline{D}/a_j \rfloor)/(k_j - 1) \rfloor = 0$  then
13:       $W'_j \leftarrow W'_j + 1/k_j$ 
14:    end if
15:  else
16:     $W'_j \leftarrow 0$ 
17:  end if
18: end for
19: if  $\sum_{j=1}^{\ell} (W_j + W'_j) < n$  then
20:    $\overline{D} \leftarrow \overline{D} + 1$ 
21: end if
22: return  $\overline{D}$ 

```

Algorithm $\overline{D}_x^\Delta(K_{k \times n})$ commences by setting \overline{D} equal to $\lceil \overline{\alpha}(x, k)n \rceil$ and then determines the ideal widths $y_j n$ and $y'_j n$ for all $j = 1, \dots, \ell$. If all these ideal widths are integral as in Example 4.4(a), then no

discretization is required and the algorithm terminates, since a good $\Delta(d, x)$ -colouring of $K_{k \times n}$ has been achieved. On the other hand, if any of the above ideal widths is not integral, as in Example 4.4(b), discretization is required, as is implicitly achieved by computation of the average discretized widths, W_j and W'_j , of the colour classes in Steps 6 and 11 of the algorithm. Note that if $\lfloor \bar{D}/a_j \rfloor = 0$, then t_j should be incremented by one more in order to obtain the same maximum degree \bar{D} as that of the other colour classes, because in this case each of the t_j vertices is adjacent to $\bar{D} - 1$ vertices in the same colour class. Similar arguments hold for t'_j . These are reflected in Steps 8 and 13 of the algorithm. If the sum of all the average discretized widths, $\sum_{j=1}^{\ell} (W_j + W'_j)$, is less than n , then all vertices cannot be coloured in such a way that the maximum degrees of all the colour class induced subgraphs are smaller than or equal to $\lceil \bar{\alpha}(x, k)n \rceil$. However, in this case it follows from Theorem 4.1 that all vertices may indeed be coloured such that all the colour class induced maximum degrees are smaller than or equal to $\lceil \bar{\alpha}(x, k)n \rceil + 1$.

Steps 5–18 of Algorithm $\bar{D}_x^\Delta(K_{k \times n})$ are performed a total of ℓ times. Therefore Algorithm $\bar{D}_x^\Delta(K_{k \times n})$ computes at most 2ℓ average discretized widths W_j and W'_j . Hence, Algorithm $\bar{D}_x^\Delta(K_{k \times n})$ also has an $\mathcal{O}(\log \log x)$ worst-case time complexity, similar to that of Algorithm $\bar{\alpha}(x, k)$.

Example 4.5 (Continuation of Example 4.4(b)) Recall that in Example 4.4(b) a colouring of the graph $K_{11 \times 14}$ in 10 colours was sought. The values of the variables determined before were $\bar{\alpha}(10, 11) = 9/14$, $k_1 = 10$, $k_2 = 10$, $s_1 = 0$, $s_2 = 1$, $a_1 = 1$, $a_2 = 2$, $\bar{\alpha}(10, 11)n = 9$, $y_1n = 9$, $y_2n = 9/2$ and $y'_2n = 1/2$. Since some of the y_jn and y'_jn values are not integral, the discretization in Steps 5–18 of Algorithm $\bar{D}_x^\Delta(K_{k \times n})$ must be performed. The values of the variables in the loop between Steps 5 and 18 of Algorithm $\bar{D}_x^\Delta(K_{k \times n})$ are determined as $W_1 = (9 + 9)/2 = 9$, $W'_1 = 0$, $W_2 = (4 + 9)/3 = 4.33$ and $W'_2 = (0 + 9 - 4)/10 = 0.5$. Since $\sum_{j=1}^2 (W_j + W'_j) = 9 + 4.33 + 0.5 = 13.83 < 14$, Algorithm $\bar{D}_x^\Delta(K_{k \times n})$ terminates with $\bar{D}_{10}^\Delta(K_{11 \times 14}) = 10$. The colour classes of a good $\Delta(10, 10)$ -colouring of $K_{11 \times 14}$ are shown graphically in Figure 4.27. ■

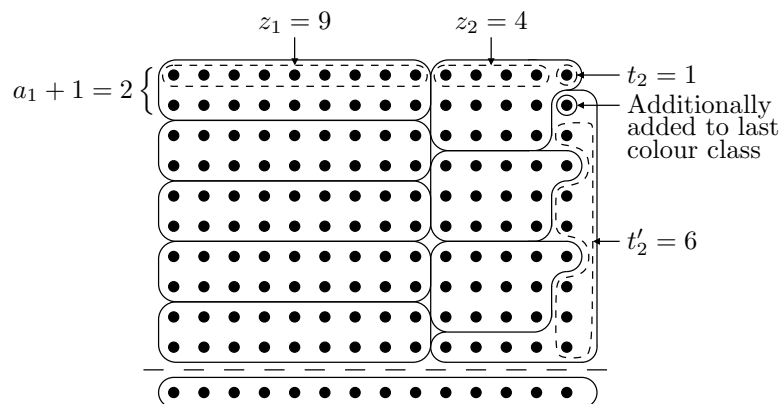


FIGURE 4.27: A graphical representation of the discretization approach achieved in Algorithm $\bar{D}_x^\Delta(K_{k \times n})$ for the graph $K_{11 \times 14}$ considered in Examples 4.4(b) and 4.5. The corresponding ideal $\Delta(9, 10)$ -colouring was represented graphically in Figure 4.25.

4.7 Chapter Summary

In this chapter the maximum degree chromatic number was sought for graphs from various structure classes. More or less in the first part of the chapter, exact values for the maximum degree chromatic number were established for graphs from various simple structure classes. These structure classes include bipartite graphs (§4.1), cycles and wheels (§4.2), complete graphs (§4.3) and products of paths and cycles (§4.4). Some of the exact values obtained in these sections will be used in the next chapter to validate $\Delta(d, x)$ -colouring algorithms.

In the remainder of the chapter either upper and lower bounds for the maximum degree chromatic number or the maximum degree chromatic number for certain values of d only, for graph classes for which the maximum degree chromatic number could not be obtained exactly, or could not be obtained for all values of d , were established. Structure classes included in this part of the chapter are products of complete graphs (§4.4), certain circulants (§4.5) and complete balanced multipartite graphs (§4.6). The results obtained for these structure classes will be used later in this dissertation in search of a characterization of the Δ -chromatic sequence of a graph G .

Chapter 5

$\Delta(d, x)$ –Colouring Algorithms

“Tell me and I’ll forget; show me and I may remember; involve me and I’ll understand.”

Chinese proverb

This chapter contains four new $\Delta(d, x)$ –colouring algorithms. The first two algorithms presented in §5.1 are heuristic methods and therefore only give an *upper bound* on the $\Delta(d)$ –chromatic number of a graph for a given value of d , whereas the subsequent two exact algorithms presented in §5.2 give the exact value of the $\Delta(d)$ –chromatic number. However, the advantage of the heuristic algorithms is that these algorithms are much more efficient in terms of time complexity than any exact algorithm for larger order graphs.

5.1 Heuristic Methods

The first heuristic, called the **colour degree heuristic**, is a greedy algorithm that employs the ideas of Breaz [13] to determine the classical chromatic number of a graph, as discussed in §2.3. Although the second heuristic may in some instances not obtain a $\Delta(d, x)$ –colouring of a graph and therefore not an upper bound on the $\Delta(d)$ –chromatic number, it often results in better upper bounds than the colour degree heuristic. This latter heuristic is a local search technique and employs the well-known tabu search technique, as summarised in Appendix E, and is therefore referred to as the **tabu search $\Delta(d, x)$ –colouring heuristic**.

The colour degree heuristic and the tabu search $\Delta(d, x)$ –colouring heuristic are described in §5.1.1 and §5.1.2 respectively. Both subsections commence with a description of the algorithm, given in pseudo-code, as well as an example to illustrate the implementation of the algorithm. The section is concluded in §5.1.3 with the results of both algorithm’s performance on 196 test graphs.

5.1.1 A Procedure Based on Breaz’s Heuristic for Proper Colourings

As mentioned in §2.3, Breaz [13] noticed that a vertex with more differently coloured vertices adjacent to it, may be harder to colour later than a vertex with the same degree and fewer differently coloured vertices adjacent to it. Recall from §2.3 that the **colour degree** of a vertex v in a graph G , denoted by $\text{coldeg}_G(v)$, is the number of *differently* coloured vertices to which it is adjacent, regardless of how many times a vertex is adjacent to identically coloured vertices. In this sense the colour degree heuristic, given in pseudo-code as Algorithm 8, is a greedy algorithm, because only the number of different colours in the neighbourhood $N(v)$ of a certain vertex v is explored, while the respective colour class induced degrees of the coloured vertices in $N(v)$ are not considered. Furthermore, during each iteration the next vertex to be coloured is selected and coloured according to what seems to be the best vertex to select

and coloured with the best colour at that point of execution of the algorithm without looking ahead to consider whether or not this might be a good choice further on during the implementation.

Algorithm 8 Colour degree heuristic

Input: A graph G of order n and a value d for which the $\Delta(d)$ -chromatic number must be determined.

Output: An upper bound x on $\chi_d^\Delta(G)$ as well as a $\Delta(d, x)$ -colouring of G .

```

1:  $L \leftarrow \{v_i \mid \deg_G(v_i) = \Delta(G)\}$ 
2: Choose  $v \in L$  and let  $\mathcal{C}_1 \leftarrow \mathcal{C}_1 \cup \{v\}$ 
3:  $x \leftarrow 1$ 
4: for all  $i = 2, \dots, n$  do
5:   Determine  $\text{coldeg}_G(v_j) \forall j = 1, \dots, n, v_j \notin \mathcal{C}_k, k = 1, \dots, x$ 
6:    $T \leftarrow \{v_i \mid v_i \notin \mathcal{C}_k, k = 1, \dots, x \text{ and } \text{coldeg}_G(v_i) \geq \text{coldeg}_G(v_j) \forall v_j \notin \mathcal{C}_k, k = 1, \dots, x\}$ 
7:   if  $|T| > 1$  then
8:     Determine  $G'$  such that  $V(G') = V(G) \setminus (\mathcal{C}_1 \cup \mathcal{C}_2 \cup \dots \cup \mathcal{C}_x)$ 
9:      $v \leftarrow$  a vertex in  $T$  such that  $\deg_{G'}(v) \geq \deg_{G'}(v_j) \forall v_j \in T$ 
10:  else  $[|T| = 1]$ 
11:     $v \leftarrow$  the only vertex in  $T$ 
12:  end if
13:   $\mathcal{C}_j \leftarrow \mathcal{C}_j \cup \{v\}$  with  $j$  the smallest possible colour number such that  $\Delta(\langle \mathcal{C}_j \cup \{v\} \rangle) \leq d$ 
14:  if  $j > x$  then
15:     $x = j$ 
16:  end if
17: end for
18: return  $x, s = (\mathcal{C}_1, \dots, \mathcal{C}_x)$ 

```

Algorithm 8 commences in Step 1 by listing all the vertices with maximum degree in a list L , followed in Step 2 by the colouring of any one of these vertices in L using colour 1. The parameter x (in Steps 3 and 15) keeps track of the number of colours used thus far. During the loop spanning Steps 4 to 17 each of the other $n - 1$ vertices are coloured in turn. This is achieved by first determining the colour degree of all the uncoloured vertices in Step 5 of Algorithm 8. All uncoloured vertices with maximum colour degree are listed in a list T in Step 6 of the algorithm. If the list T contains only one vertex (Steps 10 and 11), this vertex is coloured next, otherwise a vertex from T is selected in the if-statement spanning Steps 7–9 of Algorithm 8. To select a vertex from T , the subgraph G' of G induced by the uncoloured vertices of G , is determined in Step 8 of the algorithm. The vertex in the list T with the largest degree in G' amongst all the vertices in T , is then selected to be coloured next in Step 9 of the algorithm. If a tie still occurs, any vertex may be selected. Finally, the selected vertex v receives the smallest possible numbered colour at that point, as allocated in Step 13 of Algorithm 8, and the parameter x is updated in Step 15 if need be. The working of Algorithm 8 is illustrated in Example 5.1.

Example 5.1 Suppose a χ_1^Δ -colouring of the graph G_1 given in Figure 5.1(a) is sought. It is clear from Figure 5.1(a) that $\Delta(G_1) = 6$ and that $L = \{v_4, v_5, v_6\}$ in Step 1 of Algorithm 8. Suppose v_4 is chosen in Step 2 so that $\mathcal{C}_1 = \{v_4\}$. The colour degrees of the uncoloured vertices in G_1 determined in Step 5 of Algorithm 8 for the index $i = 2$, are $\text{coldeg}_{G_1}(v_1) = 1$, $\text{coldeg}_{G_1}(v_2) = 0$, $\text{coldeg}_{G_1}(v_3) = 1$, $\text{coldeg}_{G_1}(v_5) = 1$, $\text{coldeg}_{G_1}(v_6) = 1$, $\text{coldeg}_{G_1}(v_7) = 0$, $\text{coldeg}_{G_1}(v_8) = 1$, $\text{coldeg}_{G_1}(v_9) = 1$ and $\text{coldeg}_{G_1}(v_{10}) = 0$. Hence, $T = \{v_1, v_3, v_5, v_6, v_8, v_9\}$ in Step 6. The subgraph induced by the currently uncoloured vertices, G'_1 , is shown in Figure 5.1(b). For both vertices v_5 and v_6 it follows that $\deg_{G'_1}(v_5) = \deg_{G'_1}(v_6) = \Delta(G'_1)$ and thus, either v_5 or v_6 may be chosen in Step 9, since they are both in T . Suppose v_5 is chosen. Then $\mathcal{C}_1 = \{v_4, v_5\}$ and x remains equal to 1. The values of the parameters during the other iterations of the loop spanning Steps 4 to 17, are summarised in Table 5.1.

The algorithm terminates with $x = 3$, and thus $\chi_1^\Delta(G_1) \leq 3$, while the final colour classes obtained are $\mathcal{C}_1 = \{v_4, v_5, v_7, v_{10}\}$, $\mathcal{C}_2 = \{v_3, v_6\}$ and $\mathcal{C}_3 = \{v_1, v_2, v_8, v_9\}$ as illustrated in Figure 5.2(b). ■

Since a new colour is only introduced if all the colours used thus far are already present in the neighbourhood of the vertex currently to be coloured, Algorithm 8 always results in a $\Delta(d, x)$ -colouring with at most $\Delta(G) + 1$ colours for any graph G , although it seldom achieves this poor upper bound for larger

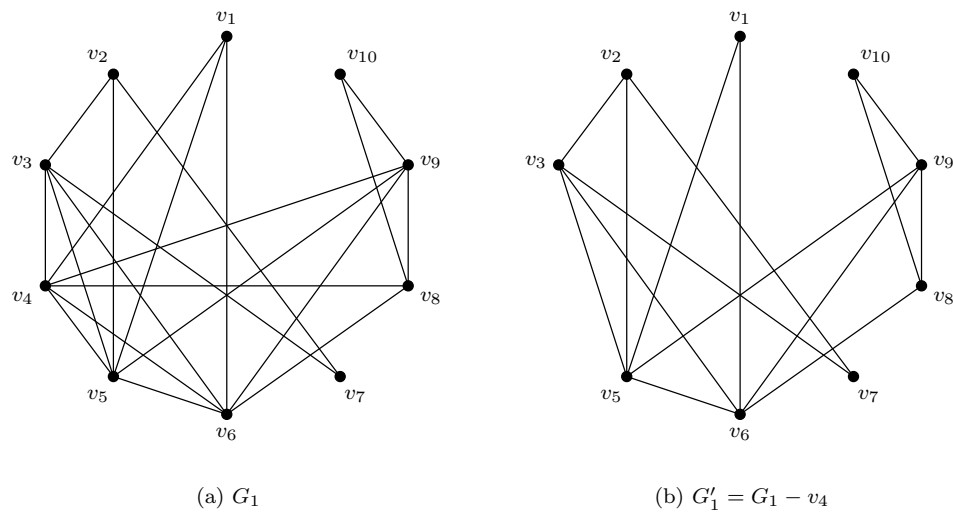


FIGURE 5.1: The graph G_1 in (a) is used to illustrate the working of the colour degree heuristic. (b) The subgraph induced by the currently uncoloured vertices, G'_1 , during the first iteration of the loop spanning Steps 4–17 of Algorithm 8.

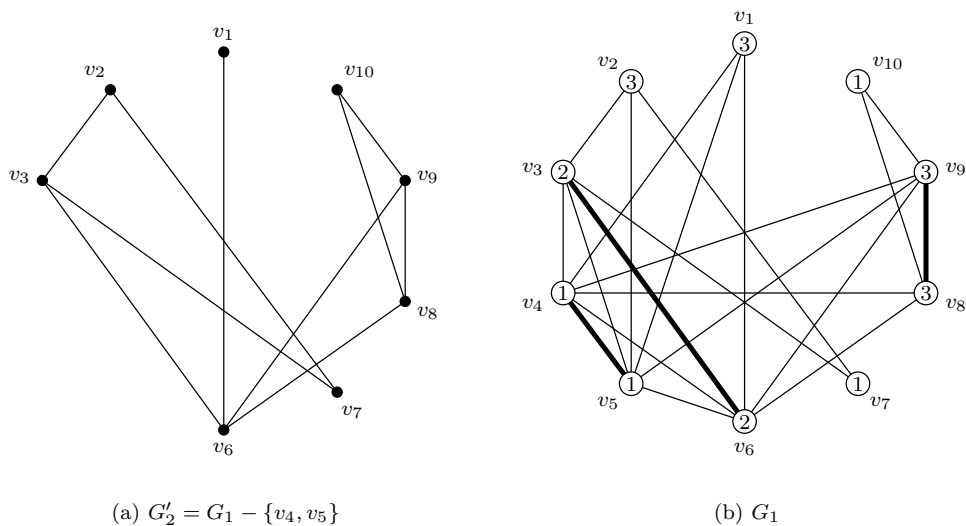


FIGURE 5.2: (a) The subgraph induced by the currently uncoloured vertices, G'_2 , during the second iteration of the loop spanning Steps 4–17 of Algorithm 8. (b) The final $\Delta(1,3)$ -colouring of G_1 in Figure 5.1(a) as obtained via the colour degree heuristic in Example 5.1. The bold edges indicate that the colour class induced maximum degree for all colour classes are 1.

values of d . Furthermore, the loop spanning Steps 4–17 is repeated exactly $n-1$ times and for a given i at most n calculations can be performed at Step 5. All other calculations are independent of n . Therefore, Algorithm 8 has a polynomial time complexity of $\mathcal{O}(n^2)$.

5.1.2 A Tabu Search Approach

Hertz and De Werra [64] implemented the tabu search technique (summarised in Appendix E) to obtain proper (classical) colourings for large graphs. Their ideas are adapted here and implemented in the context of $\Delta(d, x)$ -colourings in the tabu search $\Delta(d, x)$ -colouring heuristic. The tabu search $\Delta(d, x)$ -

$i = 3$ $\text{coldeg}_{G_1}(v_1) = \text{coldeg}_{G_1}(v_2) = \text{coldeg}_{G_1}(v_3) = \text{coldeg}_{G_1}(v_6) = \text{coldeg}_{G_1}(v_8) = \text{coldeg}_{G_1}(v_9) = 1$ and $\text{coldeg}_{G_1}(v_7) = \text{coldeg}_{G_1}(v_{10}) = 0$ $T = \{v_1, v_2, v_3, v_6, v_8, v_9\}$ The subgraph induced by the currently uncoloured vertices, G'_2 , is given in Figure 5.2(a). $\text{deg}_{G'_2}(v_1) = 1, \text{deg}_{G'_2}(v_2) = 2, \text{deg}_{G'_2}(v_3) = \text{deg}_{G'_2}(v_8) = \text{deg}_{G'_2}(v_9) = 3$ and $\text{deg}_{G'_2}(v_6) = 4$ Thus, possible choices in Step 9 : v_6 Colour class adaptation: $\mathcal{C}_2 = \{v_6\}$ $x = 2$
<hr/> $i = 4$ $\text{coldeg}_{G_1}(v_1) = \text{coldeg}_{G_1}(v_3) = \text{coldeg}_{G_1}(v_8) = \text{coldeg}_{G_1}(v_9) = 2, \text{coldeg}_{G_1}(v_2) = 1$ and $\text{coldeg}_{G_1}(v_7) = \text{coldeg}_{G_1}(v_{10}) = 0$ $T = \{v_1, v_3, v_8, v_9\}$ Possible choices in Step 9: v_3, v_8, v_9 Choice made: v_3 Colour class adaptation: $\mathcal{C}_2 = \{v_3, v_6\}$ $x = 2$
<hr/> $i = 5$ $\text{coldeg}_{G_1}(v_1) = \text{coldeg}_{G_1}(v_2) = \text{coldeg}_{G_1}(v_8) = \text{coldeg}_{G_1}(v_9) = 2, \text{coldeg}_{G_1}(v_7) = 1$ and $\text{coldeg}_{G_1}(v_{10}) = 0$ $T = \{v_1, v_2, v_8, v_9\}$ Possible choices in Step 9: v_8, v_9 Choice made: v_8 Colour class adaptation: $\mathcal{C}_3 = \{v_8\}$ $x = 3$
<hr/> $i = 6$ $\text{coldeg}_{G_1}(v_9) = 3, \text{coldeg}_{G_1}(v_1) = \text{coldeg}_{G_1}(v_2) = 2$ and $\text{coldeg}_{G_1}(v_7) = \text{coldeg}_{G_1}(v_{10}) = 1$ $T = \{v_9\}$ Colour class adaptation: $\mathcal{C}_3 = \{v_8, v_9\}$ $x = 3$
<hr/> $i = 7$ $\text{coldeg}_{G_1}(v_1) = \text{coldeg}_{G_1}(v_2) = 2$ and $\text{coldeg}_{G_1}(v_7) = \text{coldeg}_{G_1}(v_{10}) = 1$ $T = \{v_1, v_2\}$ Possible choices in Step 9: v_2 Colour class adaptation: $\mathcal{C}_3 = \{v_2, v_8, v_9\}$ $x = 3$
<hr/> $i = 8$ $\text{coldeg}_{G_1}(v_1) = \text{coldeg}_{G_1}(v_7) = 2$ and $\text{coldeg}_{G_1}(v_{10}) = 1$ $T = \{v_1, v_7\}$ Possible choices in Step 9: v_1, v_7 Choice made: v_1 Colour class adaptation: $\mathcal{C}_3 = \{v_1, v_2, v_8, v_9\}$ $x = 3$
<hr/> $i = 9$ $\text{coldeg}_{G_1}(v_7) = 2$ and $\text{coldeg}_{G_1}(v_{10}) = 1$ $T = \{v_7\}$ Colour class adaptation: $\mathcal{C}_1 = \{v_4, v_5, v_7\}$ $x = 3$
<hr/> $i = 10,$ Colour class adaptation: $\mathcal{C}_1 = \{v_4, v_5, v_7, v_{10}\}$ $x = 3$

TABLE 5.1: The values of the parameters during the iterations of Algorithm 8 applied to the graph G_1 in Example 5.1.

colouring heuristic is given in pseudo-code as Algorithm 9. The main idea behind Algorithm 9 is to attempt finding a valid $\Delta(d, x)$ -colouring of a graph G using a fixed number x of colours, starting out from an initial random x -colouring of G which is most probably an invalid $\Delta(d, x)$ -colouring. The strategy is then to repeatedly decrease x and to attempt finding a valid $\Delta(d, x)$ -colouring of the graph G using this smaller number of colours, until no valid $\Delta(d, x)$ -colouring of the graph G using a smaller number of colours can be obtained. Thus, the upper bound x on the $\Delta(d)$ -chromatic number is reduced until Algorithm 9 can find no valid $\Delta(d, x)$ -colouring for the smaller upper bound.

If the vertex set is partitioned into a fixed number x of subsets (the colour classes), resulting in a colouring with colour classes $s = (\mathcal{C}_1, \dots, \mathcal{C}_x)$, the objective is to minimize the largest colour class induced maximum degree, *i.e.* to

$$\text{minimize } f(s) = \max_{1 \leq j \leq x} \Delta(\mathcal{C}_j). \quad (5.1.1)$$

Clearly, as soon as $f(s) \leq d$ for a given value d for which the $\Delta(d)$ -chromatic number of a graph G must be determined, a valid $\Delta(d, x)$ -colouring of G has been obtained. Hence, this is the first of three stopping criteria implemented in Algorithm 9. The second stopping criterion is when the algorithm has performed a pre-specified maximum number of iterations, *maxit*, during which no solution (colouring), s , with $f(s) \leq d$ was obtained and thus no valid $\Delta(d, x)$ -colouring of G was obtained. The final stopping criterion is when no candidate solutions could be generated (if, for example, the only possibilities are all in the tabu list).

In the tabu search implementation by Hertz and De Werra [64] for proper graph colourings, each candidate solution is generated by choosing from the current solution, s , *any* vertex v with a non-zero degree in that specific colour class induced subgraph, and then recolouring v with a different colour, *randomly* chosen from the other colours used in s . In order to be more true to the original deterministic nature of the tabu search technique, each candidate solution, s^* , in the tabu search $\Delta(d, x)$ -colouring heuristic is generated by selecting from the current solution, s , a colour class i , where the colour class induced maximum degree for colour class i is equal to the largest colour class induced maximum degree for s . From colour class i , a vertex v with colour class induced degree equal to the colour class induced maximum degree for colour class i , is selected. The vertex v is then recoloured with a different colour, j , from the other colours used in s , where j is the colour with smallest colour class induced maximum degree for s . The candidate solutions are generated by sequentially stepping through all colour classes with largest colour class induced maximum degree for s , all vertices v in these colour classes with colour class induced degree equal to the largest colour class induced maximum degree and all colour classes with smallest colour class induced maximum degree to be used as new colour for v , until a total of ℓ candidate solutions are generated and stored in the candidate list. If all possible candidate solutions have been generated, as described above, and the number of candidate solutions is smaller than ℓ , then the above candidate solution generation process is repeated for a vertex u with colour class induced degree equal to the second largest colour class induced maximum degree, where u is also in a colour class with colour class induced maximum degree equal to the largest colour class induced maximum degree for s . Finally, if need be, the above procedure is repeated once more where the colour j with which the selected vertex v is recoloured, is now a colour with the second smallest colour class induced maximum degree for s . If the cardinality of the candidate list is still smaller than ℓ , the generation of candidate solutions terminates. From the candidate list (if it is not empty), the candidate solution, s' , with the smallest objective function value in (5.1.1) is chosen, which may or may not correspond to a smaller objective function value than that of the current solution s , as the next solution to move to. In contrast to the tabu search proper colouring algorithm by Hertz and De Werra [64] where, in case of a tie, a candidate solution, s' , is chosen randomly from amongst the candidate solutions with the minimum value for $f(s')$, if a tie occurs during the execution of the tabu search $\Delta(d, x)$ -colouring heuristic, the one with the smallest number of colour classes with a colour class induced maximum degree greater than d is selected.

Whenever a vertex v in colour class \mathcal{C}_i in the current solution, s , is moved to colour class \mathcal{C}_j in the next solution, s' , the pair (v, i) becomes tabu and is inserted into the tabu list, T . Thus, vertex v cannot return to colour class \mathcal{C}_i for some specified number of iterations of Algorithm 9. If the size of the tabu list T after the insertion of the pair (v, i) into T , is larger than the tabu tenure, t , the oldest entry in T is removed.

Similar to the implementation of Hertz & De Werra [64], an aspiration criterion of a tabu move as a

Algorithm 9 Tabu search $\Delta(d, x)$ -colouring heuristic

Input: A graph G of order n and a value d for which the $\Delta(d)$ -chromatic number must be determined, the tabu tenure, t , the size of the candidate list, ℓ , and the maximum number of iterations, maxit .

Output: An upper bound $\bar{\chi}_d^\Delta$ on $\chi_d^\Delta(G)$ as well as a $\Delta(d, \bar{\chi}_d^\Delta)$ -colouring of G .

```

1:  $x \leftarrow \lceil (\Delta(G) + 1) / (d + 1) \rceil$ 
2:  $\bar{\chi}_d^\Delta \leftarrow -1$ 
3: STOP  $\leftarrow$  false
4: while not STOP do
5:   itcounter  $\leftarrow$  0; EMPTY  $\leftarrow$  false
6:   Generate a random initial colouring  $s = (\mathcal{C}_1, \dots, \mathcal{C}_x)$ 
7:    $f(s) \leftarrow \max_{1 \leq j \leq x} \Delta(\langle \mathcal{C}_j \rangle)$ 
8:    $A(0) \leftarrow 0$ 
9:    $A(z) \leftarrow z - 1 \forall z = 1, \dots, \Delta(G)$ 
10:  while ( $(f(s) > d)$  and (itcounter < maxit) and (not EMPTY)) do
11:    count  $\leftarrow$  1; NOMORE  $\leftarrow$  false
12:    while ((count <  $\ell + 1$ ) and (not NOMORE)) do
13:      Choose a vertex  $v \in \mathcal{C}_i$  with  $\deg_{\langle \mathcal{C}_i \rangle}(v) = \Delta(\langle \mathcal{C}_i \rangle)$  for any  $i$  with the largest colour class induced
      maximum degree
14:      Choose a colour  $j \neq i$  for any  $j$  with the smallest colour class induced maximum degree
15:       $\mathcal{C}'_i \leftarrow \mathcal{C}_i \setminus \{v\}$ ,  $\mathcal{C}'_j \leftarrow \mathcal{C}_j \cup \{v\}$ ,  $\mathcal{C}'_k \leftarrow \mathcal{C}_k \forall k = 1, \dots, x, k \neq i, j$ 
16:       $s^* = (\mathcal{C}'_1, \dots, \mathcal{C}'_x)$ 
17:      if ( $((v, j) \notin T)$  or ( $f(s^*) \leq A(f(s))$ ) if ( $(v, j) \in T$ )) then
18:         $L \leftarrow L \cup \{s^*\}$ 
19:        count  $\leftarrow$  count + 1
20:      end if
21:      if all possibilities in Steps 13 and 14 have been considered then
22:        NOMORE  $\leftarrow$  true
23:      end if
24:    end while
25:    if  $L = \emptyset$  then
26:      EMPTY  $\leftarrow$  true
27:    else
28:       $s' \leftarrow$  a candidate in  $L$  such that  $f(s') \leq f(s_i) \forall s_i \in L$ 
29:      if  $f(s') \leq A(f(s))$  then
30:         $A(f(s)) \leftarrow f(s') - 1$ 
31:      end if
32:       $T \leftarrow T \cup \{(v, i)\}$ , where  $v \in \mathcal{C}_i$  in  $s$  and  $v \notin \mathcal{C}'_i$  in  $s'$ 
33:      if  $|T| > t$  then
34:        Remove oldest entry in  $T$ 
35:      end if
36:       $s \leftarrow s'$ ;  $f(s) \leftarrow f(s')$ 
37:      itcount  $\leftarrow$  itcount + 1
38:    end if
39:  end while
40:  if  $f(s) \leq d$  then
41:     $\bar{\chi}_d^\Delta \leftarrow x$ 
42:     $s_d \leftarrow s$ 
43:     $x \leftarrow x - 1$ 
44:  else [ $f(s) > d$ ]
45:    STOP  $\leftarrow$  true
46:  end if
47: end while
48: return  $\bar{\chi}_d^\Delta$ ,  $s_d = (\mathcal{C}_1, \dots, \mathcal{C}_{\bar{\chi}_d^\Delta})$ 

```


function $A(z)$ is defined as follows: If a tabu pair (v, i) is included in a neighbouring solution, \tilde{s} , of the current solution, s , and the objective function value, $f(\tilde{s})$, of the neighbouring solution \tilde{s} is less than or equal to the aspiration function value, $A(f(s))$, of the objective function value of the current solution s , then the tabu status of (v, i) is overridden and the neighbouring solution \tilde{s} is included in the candidate list. Initially, the aspiration criterion, $A(z)$, for each value of z is set as $z - 1$. Then, whenever a solution s' satisfying $f(s') \leq A(f(s))$ is chosen from the candidate list, $A(f(s))$ is set as $f(s') - 1$.

The details of how the above aspects were implemented in Algorithm 9 are now discussed. In Step 1 of Algorithm 9 an upper bound x on $\chi_d^\Delta(G)$ of a graph G for a given degree d , is determined. The algorithm thus first attempts to determine a valid $\Delta(d, x)$ -colouring of G using this number of colours. As soon as a feasible (valid) $\Delta(d, x)$ -colouring of G for this upper bound is reached, this feasible $\Delta(d, x)$ -colouring is stored (Steps 41 and 42 in Algorithm 9) and the upper bound x on $\chi_d^\Delta(G)$ is decreased by one in Step 43. Another iteration of the while-loop spanning Steps 4–47 is executed using this new (smaller) value of x . These iterations continue until no feasible $\Delta(d, x)$ -colouring can be obtained, in which case the boolean variable STOP in Step 45 is set to **true**. At this point the best upper bound $\bar{\chi}_d^\Delta$ (Step 41) on $\chi_d^\Delta(G)$ obtained by Algorithm 9 is $x + 1$, where x is the first upper bound for which no valid $\Delta(d, x)$ -colouring could be obtained.

Since it is possible that Algorithm 9 may not even be able to determine a feasible $\Delta(d, x)$ -colouring of G for the first upper bound on $\chi_d^\Delta(G)$ using $x = \lceil (\Delta(G) + 1)/(d + 1) \rceil$ colours, the upper bound $\bar{\chi}_d^\Delta$ on $\chi_d^\Delta(G)$ is set to the infeasible value of -1 in Step 2. If, at the end of the execution of the algorithm, the value of $\bar{\chi}_d^\Delta$ is given as -1 , it is clear that the algorithm was not able to find any feasible solution to the $\Delta(d, x)$ -colouring problem.

For a value x of the upper bound on $\chi_d^\Delta(G)$, an initial colouring of the vertices is obtained in Step 6 of Algorithm 9, after which the value of the objective function for this colouring is determined in Step 7. The search for a feasible $\Delta(d, x)$ -colouring of G is performed in the while-loop spanning Steps 10–39 of Algorithm 9. A candidate list of size ℓ is generated in the while-loop spanning Steps 12–24. In Step 28 of Algorithm 9 the next solution is chosen from the candidate list provided that the candidate list is not empty (Step 27), and the aspiration criterion is updated in Step 30, if necessary. Finally, the tabu list is updated in Steps 32–35 of Algorithm 9. On the other hand, if the candidate list is empty (Step 25) then the boolean variable EMPTY in Step 26 is set to **true** and the search for a feasible $\Delta(d, x)$ -colouring of G is concluded.

The working of Algorithm 9 is illustrated in Example 5.2.

Example 5.2 *As in Example 5.1, suppose a χ_1^Δ -colouring of the graph G_1 given in Figure 5.1(a) is sought. Furthermore, suppose the tabu tenure, t , in Algorithm 9 is 3, while the size of the candidate list, ℓ , and the maximum number of iterations, maxit , is 4 and 5 respectively. As before, it is clear from Figure 5.1(a) that $\Delta(G_1) = 6$ and thus $x = 4$ in Step 1 of Algorithm 9. Suppose the random initial $\Delta(1, 4)$ -colouring of G_1 generated in Step 6 of Algorithm 9 is the one given in Figure 5.3(a). It is clear from Figure 5.3(a) that the value of the function $f(s)$ for this colouring is 1, so that the while-loop spanning Steps 10–39 is not executed for $x = 4$. The upper bound $\bar{\chi}_d^\Delta$ on $\chi_d^\Delta(G_1)$ in Step 41 of the algorithm is set to 4, while s_d in Step 42 is $\{\{v_1, v_4\}, \{v_2, v_6, v_9\}, \{v_3, v_{10}\}, \{v_5, v_7, v_8\}\}$. At Step 43 x is decreased to 3 and the while-loop spanning Steps 4–47 needs to be repeated.*

Suppose the random initial colouring of G_1 in 3 colours generated in Step 6 of Algorithm 9 is the one given in Figure 5.3(b). For this colouring, $\Delta(\langle C_1 \rangle) = 3$, $\Delta(\langle C_2 \rangle) = 1$ and $\Delta(\langle C_3 \rangle) = 1$, so that $f(s) = 3$ in Step 7 of the algorithm. The values for the aspiration criterion in Step 9 are $A(0) = 0$, $A(1) = 0$, $A(2) = 1$, $A(3) = 2$, $A(4) = 3$, $A(5) = 4$ and $A(6) = 5$. Since $f(s) = 3 > 1 = d$, the while-loop spanning Steps 10–39 needs to be executed.

The colour class with the largest colour class induced maximum degree is C_1 with $\deg_{\langle C_1 \rangle}(v_4) = \deg_{\langle C_1 \rangle}(v_5) = \Delta(\langle C_1 \rangle)$. Both colour classes C_2 and C_3 yield the smallest colour class induced maximum degree for the initial colouring, s , given in Figure 5.3(b). Therefore, the first candidate solution generated in the while-loop spanning Steps 12–24 of Algorithm 9 is obtained by moving vertex v_4 from colour class C_1 to colour class C_2 , while the second candidate solution is obtained by moving vertex v_4 from colour class C_1 to colour class C_3 . Since the size of the candidate list, ℓ , is 4, the third and final (fourth) candidate solutions for this iteration is obtained by moving vertex v_5 from colour class C_1 to colour class C_2 , C_3

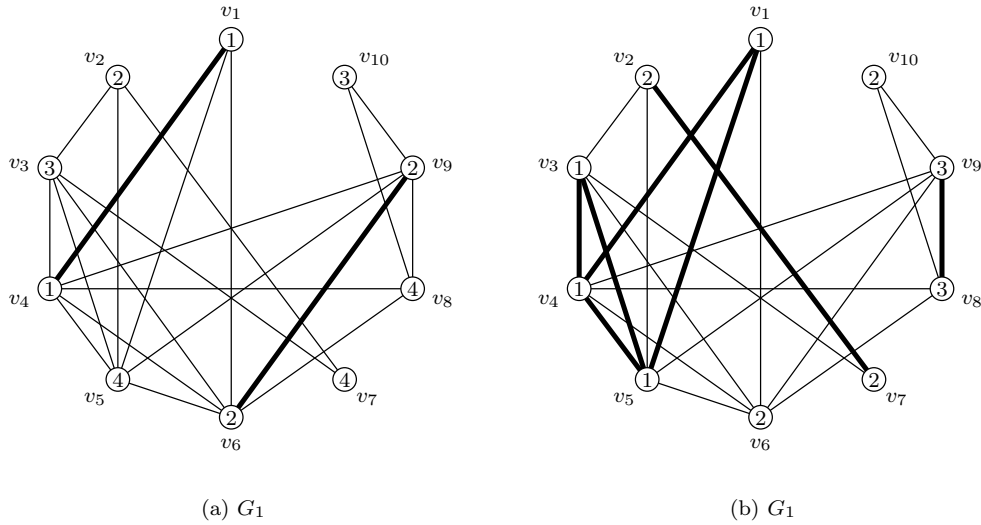


FIGURE 5.3: The graph G_1 in Figure 5.1(a) is used to illustrate the working of the tabu search $\Delta(d, x)$ -colouring heuristic. The bold edges indicate the colour class induced degrees of the vertices in both figures. (a) An initial random colouring in 4 colours as obtained in step 6 of Algorithm 9, which is also a valid $\Delta(1, 4)$ -colouring of G_1 . (b) An initial random colouring in 3 colours as obtained in step 6 of Algorithm 9, which is not a valid $\Delta(1, 3)$ -colouring of G_1 .

respectively. In summary, the candidate solutions are:

1. $s_1^* = \{\{v_1, v_3, v_5\}, \{v_2, v_4, v_6, v_7, v_{10}\}, \{v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 2$, $\Delta(\langle C'_2 \rangle) = 1$ and $\Delta(\langle C'_3 \rangle) = 1$, and thus $f(s_1^*) = 2$,
2. $s_2^* = \{\{v_1, v_3, v_5\}, \{v_2, v_6, v_7, v_{10}\}, \{v_4, v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 2$, $\Delta(\langle C'_2 \rangle) = 1$ and $\Delta(\langle C'_3 \rangle) = 2$, and thus $f(s_2^*) = 2$,
3. $s_3^* = \{\{v_1, v_3, v_4\}, \{v_2, v_5, v_6, v_7, v_{10}\}, \{v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 2$, $\Delta(\langle C'_2 \rangle) = 2$ and $\Delta(\langle C'_3 \rangle) = 1$, and thus $f(s_3^*) = 2$.
4. $s_4^* = \{\{v_1, v_3, v_4\}, \{v_2, v_6, v_7, v_{10}\}, \{v_5, v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 2$, $\Delta(\langle C'_2 \rangle) = 1$ and $\Delta(\langle C'_3 \rangle) = 2$, and thus $f(s_4^*) = 2$.

Since $f(s_i^*) = 2$ for all $i = 1, 2, 3, 4$ (i.e. for all four candidate solutions), candidate solution s_1^* is chosen at Step 28 of the algorithm, because in s_1^* only one colour class has a colour class induced maximum degree of more than $d = 1$, while candidate solutions s_2^* , s_3^* and s_4^* all have two colour class induced maximum degrees of more than $d = 1$. At Step 32 the tabu list is updated to $T = \{(v_4, 1)\}$, while the next colouring of G_1 in Figure 5.1(a) is set to s_1^* in Step 36, with $f(s) = 2$ at Step 36. The current colouring of G_1 is shown in Figure 5.4(a).

The candidate solutions during the next execution of the while-loop spanning Steps 10–39 of Algorithm 9 are given as follows:

1. $s_1^* = \{\{v_1, v_3\}, \{v_2, v_4, v_5, v_6, v_7, v_{10}\}, \{v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 0$, $\Delta(\langle C'_2 \rangle) = 3$ and $\Delta(\langle C'_3 \rangle) = 1$, and thus $f(s_1^*) = 3$,
2. $s_2^* = \{\{v_1, v_3\}, \{v_2, v_4, v_6, v_7, v_{10}\}, \{v_5, v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 0$, $\Delta(\langle C'_2 \rangle) = 1$ and $\Delta(\langle C'_3 \rangle) = 2$, and thus $f(s_2^*) = 2$,
3. $s_3^* = \{\{v_3, v_5\}, \{v_1, v_2, v_4, v_6, v_7, v_{10}\}, \{v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 1$, $\Delta(\langle C'_2 \rangle) = 2$ and $\Delta(\langle C'_3 \rangle) = 1$, and thus $f(s_3^*) = 2$.
4. $s_4^* = \{\{v_3, v_5\}, \{v_2, v_4, v_6, v_7, v_{10}\}, \{v_1, v_8, v_9\}\}$ with $\Delta(\langle C'_1 \rangle) = 1$, $\Delta(\langle C'_2 \rangle) = 1$ and $\Delta(\langle C'_3 \rangle) = 1$, and thus $f(s_4^*) = 1$.

Candidate solution s_4^* is selected since $f(s_4^*) < f(s_i^*)$ for all $i = 1, 2, 3$. The tabu list is updated to $T = \{(v_1, 1), (v_4, 1)\}$ and since $f(s_4^*) = 1 = d$, the algorithm exits the loop spanning Steps 10–39 with s_4^* as a valid $\Delta(1, 3)$ -colouring of G_1 . This valid $\Delta(1, 3)$ -colouring of G_1 is illustrated in Figure 5.4(b).

At Step 43 of Algorithm 9 the variable x is decreased to 2 and the while-loop spanning Steps 4–47 needs to be repeated once more. However, after the maximum number of iterations, 5, still no valid $\Delta(1, 2)$ -colouring of G_1 could be obtained and the algorithm terminates with $\bar{\chi}_1^\Delta = 3$ and $s_d = \{\{v_3, v_5\}, \{v_2, v_4, v_6, v_7, v_{10}\}, \{v_1, v_8, v_9\}\}$. ■

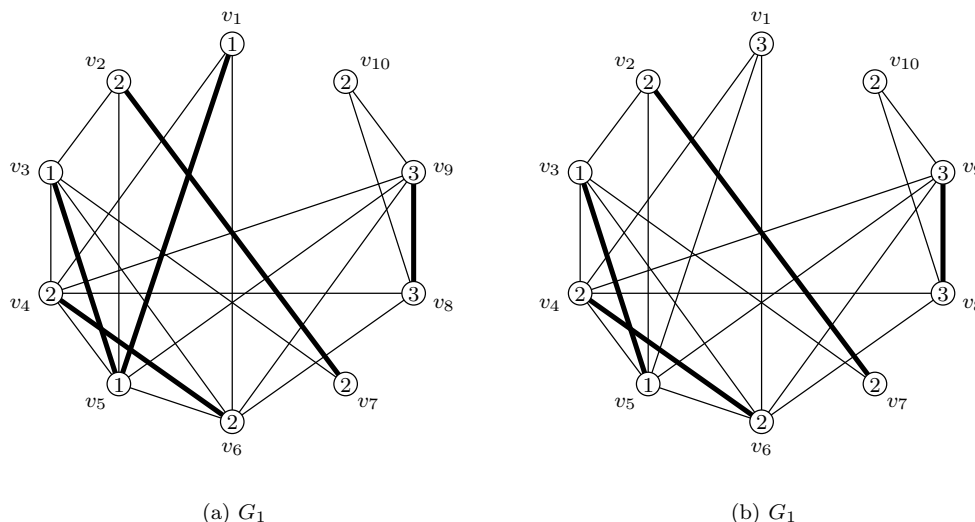


FIGURE 5.4: (a) A colouring of the graph G_1 in Figure 5.1(a) obtained after the first move during the execution of Algorithm 9 was performed on the initial random colouring of G_1 in Figure 5.3(b), and (b) the colouring of G_1 after one more move during the execution of Algorithm 9 was performed on the colouring in (a).

5.1.3 Results Obtained by the $\Delta(d, x)$ -colouring Heuristics

The test graphs used as input to any of the algorithms presented in this chapter to compare the algorithm with the other three $\Delta(d, x)$ -colouring algorithms are all listed in Appendix F and are also included on the CD accompanying this dissertation (see Appendix G). All the algorithms presented in this chapter were coded in MATHEMATICA and executed on a Dell Latitude D620 Notebook with an Intel Core 2 Duo T7200 CPU (2.0GHz with 667MHz FSB), 2 Gig 667MHz dual channel memory and Windows XP Professional SP2 as operating system.

For testing purposes, the colour degree heuristic was adapted to determine the Δ -chromatic sequences of the input graphs and not only the $\Delta(d)$ -chromatic number for a given value of d as in Algorithm 8. Furthermore, once a value of 2 is obtained for x_k in the sequence (x_i) , all values of x_d for $d = k + 1, k + 2, \dots, \Delta - 1$ are set equal to 2 in order to speed up the execution time, since all these values in the sequence (x_i) can only be 2. This adapted version of the colour degree heuristic, coded in MATHEMATICA, is included on the CD accompanying this dissertation (see Appendix G).

The colour degree heuristic was first executed on the small graphs listed in §F.1 and the results are summarised in the first three lines of Table 5.2. In the first column the types of graphs, as grouped together in the tables in Appendix F, are given. In the second, third and fourth columns the number of graphs in the particular group, the average order and the average size of the graphs in the group, respectively, are shown. For example, the first line of Table 5.2 represents the set of all 31 pairwise non-isomorphic connected graphs of order at most 5 as listed in Table F.1, where the average order of these graphs is 4.5. The average running time required to determine an upper bound on the Δ -chromatic sequence of each graph is listed in the fifth column. The sixth column contains the number of times that the colour degree heuristic obtained the correct Δ -chromatic sequence of a graph as listed in the last column of, for example, Table F.1, while the last column of Table 5.2 contains the number of times when more than one value in a particular Δ -chromatic sequence was found to be incorrect.

Graph type	# of graphs	Avg order	Avg size	Avg time per graph (in seconds)	# of sequences correct	# of times more than 2 values incorrect
Order ≤ 5	31	4.5	5.2	0.016	30	0
Regular graphs	20	7.6	14.6	0.017	11	1
Vertex-transitive	6	10.0	17.5	0.026	5	0
Trees	23	8.0	7.0	0.008	23	0
Cycles	6	31.8	31.8	0.404	6	0
Wheels	6	31.8	61.7	0.854	6	0
Complete graphs	6	31.8	579.0	9.457	6	0
Complete bipartite graphs	6	57.0	824.8	1.861	6	0

TABLE 5.2: The results obtained by the colour degree heuristic when applied to small graphs (order at most 10) and the graph structure classes of Chapter 4.

Note that for most graphs in the first two lines of Table 5.2 the computation times were too short to be captured by MATHEMATICA. For this reason, only ten instead of 31 graphs [18 instead of 20], were used to determine the average running time per graph in the case of graphs of order at most 5 [the regular graphs listed in Table F.2]. This was also the case for the tree graph structure class, where only twelve graphs instead of all 23 trees, were used to determine the average running time per tree. In the case of all pairwise non-isomorphic connected graphs of order at most 5, only one value in the Δ -chromatic sequence of the test graph $G48$ was determined to be a higher value than the exact value. In each of the Δ -chromatic sequences of the regular graphs $C2$, $C4$, $C5$, $Q6$, $Q8$, $F2$, $F3$ and $F4$ one value was determined to be larger than the exact value, while two values in the Δ -chromatic sequence of the regular graph $Q7$ were larger than the exact values. Finally, only one value in the Δ -chromatic sequences of one of the vertex-transitive graphs in Table F.3 was determined to be higher than the exact value, namely x_1 in the Δ -chromatic sequence of $Ct6$.

The results of the colour degree heuristic when applied to the graphs from structure classes given in Table F.4–F.6 are summarised in lines four to eight of Table 5.2. For all examples from these structure classes the colour degree heuristic determined the correct Δ -chromatic sequence.

The colour degree heuristic was also applied to a few complete balanced multipartite graphs in order to compare the colour degree heuristic to the heuristic method in §4.6, developed specifically for complete balanced multipartite graphs. In this case the colour degree heuristic performs poorly. For example, the Δ -chromatic sequence of $K_{4 \times 4}$ obtained by the heuristic method of §4.6 is 4 4 4 3 2 2 2 2 2 2 2 2 1, while the colour degree heuristic determines $\bar{\chi}_1^\Delta(K_{4 \times 4}) = 8$. The reason for this is the greediness of Algorithm 8. After the first vertex, say v_k , is coloured, the colour degree of all the vertices in the other partite sets is one, while the vertices in the same partite set as v_k have a colour degree of zero. Hence, a vertex in one of the partite sets not containing v_k , is selected in Step 9 of Algorithm 8 to be coloured second. Since $d = 1$, this vertex will be coloured with the same colour as v_k , but then no other vertex may be coloured with this colour. This phenomenon repeats itself until eight colours are necessary to colour $K_{4 \times 4}$. In contrast, the heuristic method of §4.6 colours all the vertices in the same partite set with the same colour and thus uses only four colours to colour $K_{4 \times 4}$. The upper bounds on the first fifteen values in the Δ -chromatic sequence of the other complete balanced multipartite graphs in Table F.7, as obtained by the colour degree heuristic, are given in Table 5.3. For comparison purposes the upper bounds as obtained by the method of §4.6 are reproduced from Table F.7 as well.

The results of the colour degree heuristic applied to the proper colouring benchmark graphs in §F.3 are summarised in Table 5.4. The first four columns are repeated from Table F.8 and include the order, the size and the (classical) chromatic number of the specific graph listed in the first column. The fifth column contains the average running time required by the colour degree heuristic to determine the Δ -chromatic sequence. The last six columns in Table 5.4 show the upper bounds on χ_0^Δ , χ_1^Δ , χ_2^Δ , χ_4^Δ , χ_8^Δ and χ_{12}^Δ respectively, as obtained by the colour degree heuristic.

Finally, the results obtained by the colour degree heuristic applied to the random graphs in §F.4 are summarised in Table 5.5. In this case the last six columns in Table 5.5 contain the average upper bounds obtained by the colour degree heuristic on χ_0^Δ , χ_1^Δ , χ_2^Δ , χ_4^Δ , χ_8^Δ and χ_{12}^Δ respectively.

Graph	Method	Upper bounds on the Δ -chromatic sequence												
$K_{4 \times 7}$	Colour degree heuristic	4	14	8	6	4	4	4	2	2	2	2	2	2
$K_{4 \times 7}$	Method of §4.6	4	4	4	4	4	4	3	2	2	2	2	2	2
$K_{4 \times 11}$	Colour degree heuristic	4	22	12	8	6	6	4	4	4	4	4	2	2
$K_{4 \times 11}$	Method of §4.6	4	4	4	4	4	4	4	4	4	3	3	2	2
$K_{7 \times 4}$	Colour degree heuristic	7	13	7	7	4	4	4	3	3	3	3	2	2
$K_{7 \times 4}$	Method of §4.6	7	7	7	6	4	4	4	3	3	3	3	2	2
$K_{7 \times 11}$	Colour degree heuristic	7	34	19	13	10	10	7	7	7	7	7	4	4
$K_{7 \times 11}$	Method of §4.6	7	7	7	7	7	7	7	7	6	6	5	4	4
$K_{11 \times 4}$	Colour degree heuristic	11	21	11	11	6	6	6	6	4	4	4	4	3
$K_{11 \times 4}$	Method of §4.6	11	11	11	9	6	6	6	5	4	4	4	4	3

TABLE 5.3: Upper bounds on the values in the Δ -chromatic sequence obtained by the colour degree heuristic, when applied to complete balanced multipartite graphs, are listed in all the odd rows. In the even numbered rows the good upper bounds obtained by the heuristic method of §4.6 are given.

Graph	Order	Size	χ	Running time (s)	$\bar{\chi}_0^\Delta$	$\bar{\chi}_1^\Delta$	$\bar{\chi}_2^\Delta$	$\bar{\chi}_4^\Delta$	$\bar{\chi}_8^\Delta$	$\bar{\chi}_{12}^\Delta$
queen5_5	25	160	5	1.031	5	7	5	3	2	2
queen6_6	36	290	7	3.219	8	8	7	4	3	2
queen7_7	49	476	7	8.704	10	8	7	4	3	2
queen8_8	64	728	9	19.323	11	9	8	5	3	2
myciel4	11	20	4	0.031	4	2	2	2	1	1
myciel5	23	71	5	0.453	5	3	3	2	2	1
myciel6	47	236	6	3.953	6	4	3	3	2	2
myciel7	95	755	7	42.753	7	6	6	4	3	2
huck	74	301	11	16.407	11	6	5	4	2	2
jean	80	254	10	17.751	10	7	5	3	2	2
david	87	406	11	33.158	11	7	6	4	3	2

TABLE 5.4: The results obtained by the colour degree heuristic applied to the proper colouring benchmark graphs.

Order n	Density p	# of graphs	Average						
			time (s)	$\bar{\chi}_0^\Delta$	$\bar{\chi}_1^\Delta$	$\bar{\chi}_2^\Delta$	$\bar{\chi}_4^\Delta$	$\bar{\chi}_8^\Delta$	$\bar{\chi}_{12}^\Delta$
20	0.2	5	0.188	3.6	3.2	2.2	2	1	1
20	0.5	5	0.416	6	5.2	4	3	2	1.8
20	0.8	5	0.710	9.2	8	5.2	4	2	2
35	0.2	5	1.466	4.6	4	3.6	2.6	2	1.6
35	0.5	5	3.107	8.8	7.2	6	4.2	3	2
35	0.8	5	5.236	14.4	12.6	9	5.8	4	3
50	0.2	5	4.522	5.8	4.8	4	3	2	2
50	0.5	5	11.540	10.6	9.4	8	5.4	3.6	3
50	0.8	5	19.656	19	18.2	12.2	8	5	4
65	0.2	5	11.566	6.6	5.6	4.6	3.4	2	2
65	0.5	5	31.597	13.2	11.4	10.2	6.4	4	3
65	0.8	5	53.195	22.6	21.2	15.4	9.8	6	4.6
85	0.2	5	32.593	7.8	6.8	5.8	4	3	2
85	0.5	5	87.536	15.8	14.2	12	8	5	4
85	0.8	5	148.947	28.6	26.6	20.2	12.6	7.8	6

TABLE 5.5: The results obtained by the colour degree heuristic applied to random graphs of the form $G_{n,p}$, where n is the order of the graph and p is the density of the graph.

In order to determine the sensitivity of the initial vertex choice in the colour degree heuristic, Algorithm 8 was adapted to start with every vertex in L (Step 1) rather than to choose any vertex in L as was performed in Step 2. This adapted version of Algorithm 8 was rerun on some of the graphs on which the

original colour degree heuristic was executed. It was mentioned in §2.1 that, if an algorithm is applied to a vertex-transitive graph, any vertex may be used for initialisation of the algorithm without loss of generality or efficiency of the algorithm. The vertex-transitive graphs listed in Table F.3 were therefore excluded from the sensitivity analysis of the colour degree heuristic. Similarly, since cycles, complete graphs and complete balanced multipartite graphs are also vertex-transitive, these graphs were likewise excluded from the sensitivity analysis of the colour degree heuristic. Any graph from the graph structure class wheels has only one vertex with maximum degree, while all vertices with maximum degree in a complete bipartite graph are alike. These graphs, too, were excluded from the sensitivity analysis of the colour degree heuristic. Finally, in the case of the graph structure class trees, once the first vertex has been coloured, the use of the colour degree in the colour degree heuristic will prevent a situation where both vertices adjacent to the next vertex to be coloured, will already have been coloured. Thus, in the case of trees, the colour degree heuristic will always use at most two colours to colour the graph. For this reason, the trees listed in Table F.4 were also excluded when sensitivity analysis was performed on the initial vertex to be coloured in the colour degree heuristic. A sensitivity analysis of the colour degree heuristic was performed on all remaining graphs listed in Appendix F, *i.e.* these graphs included all the pairwise non-isomorphic connected graphs listed in Table F.1, the connected regular graphs listed in Table F.2, the proper colouring benchmark graphs listed in §F.3 as well as all the random graphs generated according to Table F.9.

The results of the sensitivity analysis on the choice of initial vertex to be coloured are summarised in Table 5.6. All graphs not listed in Table 5.6, but on which sensitivity analysis was performed, have only one vertex of maximum degree. In the first three columns of Table 5.6 the results of the sensitivity analysis of the colour degree heuristic on all the pairwise non-isomorphic connected graphs of order at most 5 as well as the regular graphs listed in Table F.2 are shown, where the second column contains the number of times that the correct Δ -chromatic sequence was obtained out of the number of times the colour degree heuristic was run on a particular graph, *i.e.* the number of vertices with maximum degree which exist in that particular graph. In the third column all the values of the parameter d are shown for which the $\Delta(d)$ -chromatic number in the Δ -chromatic sequence of the particular graph was determined incorrectly. In the last three columns of Table 5.6 the results on the sensitivity analysis of the colour degree heuristic on the proper colouring benchmark graphs as well as the random graphs are shown. In this case the fifth column lists the number of times that the smallest upper bounds on the values in the Δ -chromatic sequence were obtained out of the number of times the colour degree heuristic was run on a particular graph.

In the majority of cases (65.6%) the colour degree heuristic resulted in the same sequence, regardless of which vertex with maximum degree was selected first. Of the 22 graphs for which the obtained sequence was different during the various runs of the colour degree heuristic on that particular graph, the difference occurred only at the value $d = 1$ in 14 cases, the difference in the values in the sequences occurred at more than one value of d in 6 cases, including $d = 1$ in all cases, and the difference at a value of $d \neq 1$ in only 2 cases.

The tabu search $\Delta(d, x)$ -colouring heuristic (Algorithm 9) was also adapted to determine the Δ -chromatic sequences of the input graphs — not only computing the $\Delta(d)$ -chromatic number for a given value of d as in Algorithm 9. Although the growth property in Theorem 3.1 may be used to obtain a possibly better upper bound on $\chi_d^\Delta(G)$ of the input graph G once an upper bound on $\chi_{d-1}^\Delta(G)$ has been established, this was not incorporated during the implementation in MATHEMATICA. Instead, the upper bound $\lceil (\Delta(G) + 1) / (d + 1) \rceil$ in Step 1 of Algorithm 9 was used for all values of d in order to determine the same value of the upper bound on $\chi_d^\Delta(G)$ that would have been obtained if the algorithm had been implemented to determine an upper bound $\bar{\chi}_d^\Delta$ on $\chi_d^\Delta(G)$ for a specific value of d only, and not the entire Δ -chromatic sequence of G . As before, the tabu search $\Delta(d, x)$ -colouring heuristic was also adapted so that once a value of 2 was obtained for x_k in the Δ -chromatic sequence (x_i) , all values of x_d for $d = k + 1, k + 2, \dots, \Delta - 1$ were set equal to 2 in order to speed up the execution time. This adapted version of the tabu search $\Delta(d, x)$ -colouring heuristic, coded in MATHEMATICA, is included on the CD accompanying this dissertation (see Appendix G).

Before the tabu search $\Delta(d, x)$ -colouring heuristic could be applied to the small graphs listed in §F.1, the graph G_1 in Figure 5.1(a) used to illustrate the working of the tabu search $\Delta(d, x)$ -colouring heuristic in Example 5.2, was used to determine a set of good values for the parameters t (the tabu tenure), ℓ (the

Graph	# times correct out of # of runs	d values	Graph	# times the smallest sequence out of # of runs	d values
$G3$	2 out of 2	—	queen6_6	1 out of 4	$d = 1$
$G7$	3 out of 3	—	queen8_8	3 out of 4	$d = 7$
$G14$	2 out of 2	—	$G_{20,0.2}$ #2	2 out of 2	—
$G16$	4 out of 4	—	$G_{20,0.5}$ #1	2 out of 2	—
$G17$	2 out of 2	—	$G_{20,0.5}$ #2	4 out of 4	—
$G18$	4 out of 4	—	$G_{20,0.5}$ #4	2 out of 2	—
$G31$	3 out of 3	—	$G_{20,0.5}$ #5	2 out of 2	—
$G35$	2 out of 2	—	$G_{20,0.8}$ #2	3 out of 4	$d = 1$
$G38$	5 out of 5	—	$G_{20,0.8}$ #3	2 out of 2	—
$G41$	3 out of 3	—	$G_{35,0.5}$ #2	2 out of 2	—
$G43$	2 out of 2	—	$G_{35,0.5}$ #4	2 out of 2	—
$G44$	2 out of 2	—	$G_{35,0.8}$ #3	3 out of 5	$d = 0$
$G46$	2 out of 2	—	$G_{35,0.8}$ #4	2 out of 2	—
$G48$	1 out of 4	$d = 1$	$G_{35,0.8}$ #5	1 out of 3	$d = 1$
$G49$	2 out of 2	—	$G_{50,0.2}$ #2	4 out of 4	—
$G51$	3 out of 3	—	$G_{50,0.2}$ #3	3 out of 3	—
$G52$	5 out of 5	—	$G_{50,0.2}$ #4	1 out of 2	$d = 1$
$C2$	4 out of 6	$d = 1$	$G_{50,0.5}$ #2	2 out of 2	—
$C3$	6 out of 6	—	$G_{50,0.5}$ #5	2 out of 3	$d = 1, 3$
$C4$	3 out of 8	$d = 1$	$G_{50,0.8}$ #2	2 out of 2	—
$C5$	2 out of 8	$d = 1$	$G_{65,0.2}$ #5	2 out of 2	—
$C6$	6 out of 8	$d = 1$	$G_{65,0.5}$ #1	2 out of 2	—
$C7$	8 out of 8	—	$G_{65,0.8}$ #1	2 out of 2	—
$C8$	8 out of 8	—	$G_{85,0.2}$ #2	2 out of 2	—
$Q2$	6 out of 6	—	$G_{85,0.5}$ #2	2 out of 2	—
$Q3$	3 out of 7	$d = 1$	$G_{85,0.5}$ #5	3 out of 3	—
$Q4$	4 out of 7	$d = 1$	$G_{85,0.8}$ #3	2 out of 2	—
$Q5$	6 out of 8	$d = 0, 1, 2$			
$Q6$	3 out of 8	$d = 1, 3$			
$Q7$	1 out of 8	$d = 1$			
$Q8$	0 out of 8	$d = 0, 1$			
$Q9$	4 out of 8	$d = 1$			
$Q10$	8 out of 8	—			
$F2$	1 out of 8	$d = 1$			
$F3$	0 out of 8	$d = 0, 1, 2$			
$F4$	3 out of 8	$d = 1, 2$			
$X2$	8 out of 8	—			

TABLE 5.6: The results of a sensitivity analysis on the choice of initial vertex to be coloured in the colour degree heuristic. The second and fifth columns list the number of times that the correct Δ -chromatic sequence or the smallest upper bounds on the values in the Δ -chromatic sequence respectively, was obtained out of the number of times the colour degree heuristic was run on the particular graph. In the third and sixth columns all the values of the parameter d for which different values were obtained in the sequences determined during each run of the particular graph.

size of the candidate list) and maxit (the maximum number of iterations) for graphs of order at most 10. For this purpose the objective was to minimize the sum of the entries in the sequence of upper bounds on the Δ -chromatic sequence of G_1 , since the minimum of each entry is sought. The parameter t [ℓ , maxit, respectively] was varied between the values 5 and 10 [5 and 12, 10 and 28, respectively] when the tabu search $\Delta(d, x)$ -colouring heuristic was applied to G_1 . For each set of parameter values the tabu search $\Delta(d, x)$ -colouring heuristic was executed on G_1 a total of ten times, because of the randomness of the initial colouring. The mean value of the objective function values of the ten runs for each set of parameter values was used to determine a set of good values for the parameters t , ℓ and maxit for

graphs of order at most 10. A selection of the mean objective function values obtained by the procedure described above is shown in Figure 5.5.

It is easy to verify by hand that the Δ -chromatic sequence of G_1 is 4322221 so that the minimum possible objective function value is 16. This minimum value of 16 was obtained for seven different sets of parameter values, where maxit was 28 in all these cases. For each of these seven sets of parameter values, the tabu search $\Delta(d, x)$ -colouring heuristic was applied to G_1 a further 50 times, because the randomness of the initial solution might still have a slight influence on the performance of the algorithm. The results obtained for these runs are summarised in Table 5.7.

Maxit	ℓ	t	# times correct sequence obtained out of 50 runs	avg running time per sequence (s)
28	8	7	46	0.590
28	10	6	49	0.676
28	10	7	50	0.635
28	10	9	48	0.611
28	12	5	49	0.733
28	12	6	48	0.700
28	12	7	50	0.667

TABLE 5.7: Results obtained when the tabu search $\Delta(d, x)$ -colouring heuristic was applied to G_1 a further 50 times for each set of parameter values for which the original objective function value was 16.

It was found that for both sets of parameter values $(\text{maxit}, \ell, t) = (28, 10, 7), (28, 12, 7)$ the same sequence of upper bounds on $\chi_d^\Delta(G_1)$ resulted for all $d \geq 0$, for all executions of Algorithm 9, while for other combinations of parameter values in Table 5.7 at least one sequence contained a different entry than the other sequences obtained. Finally, the set of parameter values $(28, 10, 7)$ was chosen to be implemented when the tabu search $\Delta(d, x)$ -colouring heuristic was applied to the test graphs of order at most ten, since the average running time for this set of parameters was slightly shorter than that for the set of parameter values $(28, 12, 7)$.

Since the initial colouring obtained in Step 6 of Algorithm 9 is random, the algorithm may perform slightly better or worse depending on the quality of the initial random colouring. For this reason the tabu search $\Delta(d, x)$ -colouring heuristic was executed five times for each test graph.

The results of the tabu search $\Delta(d, x)$ -colouring heuristic applied to the small graphs listed in §F.1 as well as the trees listed in Table F.4, are shown in Tables 5.8 and 5.9. The first six columns of Table 5.8 are reproduced from Table 5.2 in order to compare the tabu search $\Delta(d, x)$ -colouring heuristic to the colour degree heuristic. In the next ten columns of Table 5.8 the average running times (in seconds) and the number of correct sequences obtained for the particular group of graphs listed in the first column are shown for each of the five executions of the tabu search $\Delta(d, x)$ -colouring heuristic. In the case of all 31 connected graphs of order at most 5, the tabu search $\Delta(d, x)$ -colouring heuristic determined the correct sequences for all the graphs during all five executions of the algorithm, while the colour degree heuristic obtained one incorrect sequence. In the case of the regular graphs listed in Table F.2 the tabu search $\Delta(d, x)$ -colouring heuristic outperformed the colour degree heuristic during each execution of the algorithm in terms of the number of correct sequences obtained. The tabu search $\Delta(d, x)$ -colouring heuristic applied to the vertex-transitive graphs listed in Table F.3 also obtained either the same number of correct sequences as, or more than, the colour degree heuristic during all five executions of the algorithm. However, in the case of trees, the tabu search $\Delta(d, x)$ -colouring heuristic obtained at least one incorrect sequence, while the colour degree heuristic always obtained the correct sequence of a tree. The reason for this, as mentioned before, is that the specific structure of a tree forces the colour degree heuristic to obtain an optimal colouring, while the tabu search $\Delta(d, x)$ -colouring heuristic does not make use of this structure at all. The average number of iterations performed during each of the five executions of the tabu search $\Delta(d, x)$ -colouring heuristic applied to the small graphs listed in §F.1 as well as the trees listed in Table F.4, are shown in Table 5.9.

The random graph of the form $G_{35,0.5}$ (see §F.4) that was generated second was used to determine a set of values for the parameters t, ℓ and maxit for graphs of orders 11–49. As in the case of the graph G_1 in Figure 5.1(a), the objective was to minimize the sum of the entries in the sequence of upper bounds on

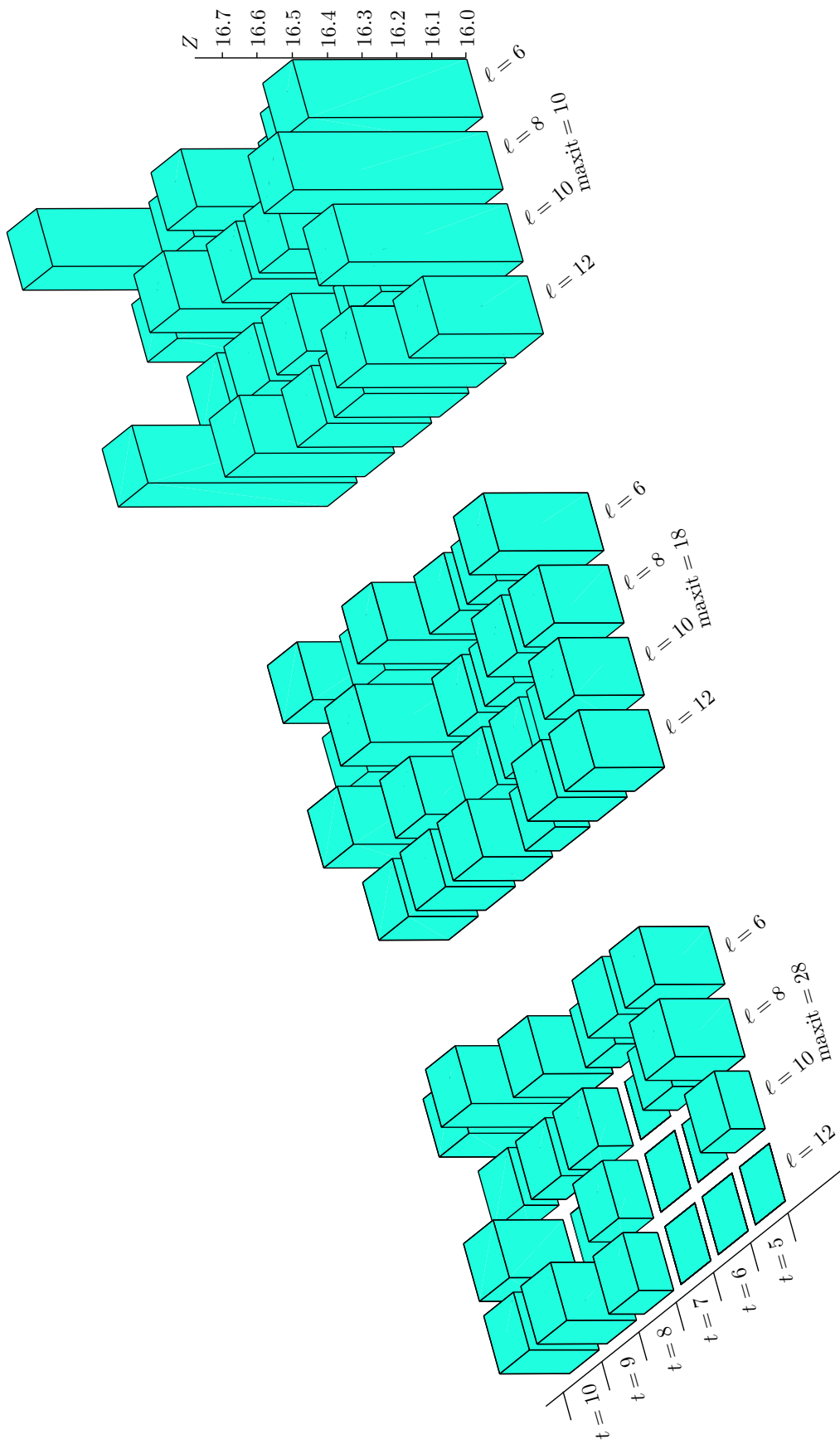


FIGURE 5.5: Mean objective function values, Z , obtained by the tabu search $\Delta(d, x)$ -colouring heuristic when applied to the graph G_1 in Figure 5.1(a) for different sets of tabu parameter values.

Graph type	# of graphs	Tabu search $\Delta(d, x)$ -colouring heuristic														
		Colour degree heuristic					Run 1		Run 2		Run 3		Run 4		Run 5	
		Avg order	Avg size	Avg time	# seq correct	Avg time	Avg time	# seq correct	Avg time	# seq correct	Avg time	# seq correct	Avg time	# seq correct	Avg time	# seq correct
Order at most 5	31	4.5	5.2	0.016	30	0.034	0.031	31	0.035	31	0.033	31	0.036	31		
Regular graphs	20	7.6	14.6	0.017	11	0.208	0.215	16	0.203	16	0.217	15	0.219	14		
Vertex-transitive	6	10.0	17.5	0.026	5	0.365	0.362	5	0.307	6	0.398	6	0.336	6		
Trees	23	8.0	7.0	0.008	23	0.045	0.052	22	0.055	16	0.045	20	0.045	21		

TABLE 5.8: The results obtained by the tabu search $\Delta(d, x)$ -colouring heuristic when applied to small graphs (order at most 10) and trees. In all cases the tabu parameter values were taken as $\text{maxit} = 28$, $\ell = 10$ and $t = 7$. All times are given in seconds.

Graph type	Avg order	Avg size	Average # of iterations during execution of the tabu search $\Delta(d, x)$ -colouring heuristic														
			Run 1			Run 2			Run 3			Run 4			Run 5		
			$d = 0$	$d = 1$	seq.	$d = 0$	$d = 1$	seq.	$d = 0$	$d = 1$	seq.	$d = 0$	$d = 1$	seq.	$d = 0$	$d = 1$	seq.
Order at most 5	4.5	5.2	8.6	1.3	9.9	8.3	1.0	9.4	7.9	1.4	9.3	8.7	1.1	9.8	9.1	1.4	10.5
Regular graphs	7.6	14.6	24.9	9.4	35.1	24.2	10.7	35.6	24.6	8.8	34.4	25.9	9.2	35.9	26.9	9.2	36.8
Vertex-transitive	10.0	17.5	33.2	8.5	42.7	32.3	9.2	42.2	27.0	8.5	35.8	34.2	11.3	46.0	33.0	5.3	39.2
Trees	8.0	7.0	7.6	0.0	7.6	8.9	0.1	9.0	9.5	0.3	9.8	7.7	0.1	7.9	7.8	0.1	8.0

TABLE 5.9: The average number of iterations performed during the execution of the tabu search $\Delta(d, x)$ -colouring heuristic applied to small graphs (order at most 10) and trees. In all cases the tabu parameter values were taken as $\text{maxit} = 28$, $\ell = 10$ and $t = 7$. The average number of iterations performed by the tabu search $\Delta(d, x)$ -colouring heuristic to obtain an upper bound on $\chi_0^\Delta [X_1^\Delta]$, the entire Δ -chromatic sequence, respectively] are shown in the columns labelled $d = 0$ [$d = 1$, seq., respectively] for each of the five runs.

the Δ -chromatic sequence of $G_{35,0.5}$ #2. The parameter t was varied between the values 6 and 15 when the tabu search $\Delta(d, x)$ -colouring heuristic was applied to $G_{35,0.5}$ #2, while the parameter ℓ [maxit, respectively] was varied between 30 and 40 [over the set of values $\{120, 140, 155, 170\}$, respectively]. As before, for each set of parameter values the tabu search $\Delta(d, x)$ -colouring heuristic was executed on $G_{35,0.5}$ #2 a total of ten times and the mean value of the objective function values of the ten runs for each set of parameter values was determined. A selection of these resulting mean objective function values is shown in Figure 5.6.

The minimum objective function value when maxit = 120 is 66.0, while a minimum objective function value of 65.7 was obtained when maxit = 140. However, as maxit increased to 170, the minimum objective function value was still 65.7, while the average running time increased from 327 seconds to 365 seconds. It was therefore decided to choose maxit = 140. For both sets of parameter values (maxit, ℓ , t) = (140, 40, 7), (140, 40, 8) the minimum objective function value of 65.7 was obtained. As before, for each of these two sets of parameter values, the tabu search $\Delta(d, x)$ -colouring heuristic was applied to $G_{35,0.5}$ #2 a further 50 times. During these additional runs, the performance of the algorithm was slightly better for the set of parameter values (140, 40, 8) than for the set of parameter values (140, 40, 7). Thus, the set of parameter values (140, 40, 8) was chosen to be implemented when the tabu search $\Delta(d, x)$ -colouring heuristic was applied to the test graphs of orders 11–49.

It was noted that the mean objective function values obtained for maxit = 170 seems random. A reason for this might be that the execution of the tabu search $\Delta(d, x)$ -colouring heuristic on a graph of order 35 for a total of ten times only is not enough to smooth out the effects of the randomness of the initial solution. However, for most sets of parameter values the running time of the tabu search $\Delta(d, x)$ -colouring heuristic applied to $G_{35,0.5}$ #2 for a total of ten times was more than an hour. For this reason the number of executions of the program on $G_{35,0.5}$ #2 for each set of parameter values was not increased.

The results of the tabu search $\Delta(d, x)$ -colouring heuristic when applied to the graphs of order at most 50 from the structure classes listed in Tables F.5 and F.6 are summarised in Tables 5.10 and 5.11. Note that although another set of parameters was determined for graphs of order at least 50, C_{50} , W_{50} and K_{50} were included in the results listed in Tables 5.10 and 5.11, since these graphs would otherwise be the only cycle, wheel or complete graph listed in Table F.5 to which the tabu search $\Delta(d, x)$ -colouring heuristic was not applied. As in the case of graphs of order at most 10, the first six columns of Table 5.10 are reproduced from Table 5.2 and in the next ten columns of Table 5.10 the average running times (in seconds) and the number of correct sequences obtained for the particular group of graphs listed in the first column are shown for each of the five executions of the tabu search $\Delta(d, x)$ -colouring heuristic.

Similar to the results obtained when the tabu search $\Delta(d, x)$ -colouring heuristic was applied to trees, the algorithm also performed worse than the colour degree heuristic when applied to cycles and wheels. Again the specific structure of a cycle or a wheel forces the colour degree heuristic to obtain an optimal colouring, while the tabu search $\Delta(d, x)$ -colouring heuristic does not make use of this structure at all. Furthermore, in all cases where the tabu search $\Delta(d, x)$ -colouring heuristic did not obtain the correct sequence, the only incorrect entry was an upper bound of 3 instead of $\chi_0^\Delta = 2$ for an even cycle or an upper bound of 4 instead of $\chi_0^\Delta = 3$ for an odd wheel. In the case of complete graphs or complete bipartite graphs the tabu search $\Delta(d, x)$ -colouring heuristic obtained the correct sequence during each run.

The average number of iterations performed during each of the executions of the tabu search $\Delta(d, x)$ -colouring heuristic summarised in Table 5.10 are shown in Table 5.11. The difference between the upper bound $\lceil(\Delta + 1)/(d + 1)\rceil$ and χ_d^Δ for a wheel is relatively large. For example, for $d = 0$ it follows that $\lceil(\Delta(W_{29}) + 1)\rceil = 29$, while $\chi_0^\Delta(W_{29}) = 3$. Therefore the number of colours (x) for which a valid $\Delta(d, x)$ -colouring is sought, should be adjusted several times during the course of the tabu search $\Delta(d, x)$ -colouring heuristic, resulting in a large number of iterations performed by the algorithm. In the case of a complete graph, K_n , a number of iterations of the tabu search $\Delta(d, x)$ -colouring heuristic are performed to obtain a χ_d^Δ -colouring since $\lceil(\Delta(K_n) + 1)/(d + 1)\rceil = \chi_d^\Delta(K_n)$. In most cases the maximum number of iterations, maxit, are then performed to search for a valid $\Delta(d, \chi_d^\Delta(K_n) - 1)$ -colouring for each value of d , resulting once again in a large number of iterations.

The tabu search $\Delta(d, x)$ -colouring heuristic was also applied to a few complete balanced multipartite graphs of order at most 50 in order to compare the algorithm to the heuristic method in §4.6. The results

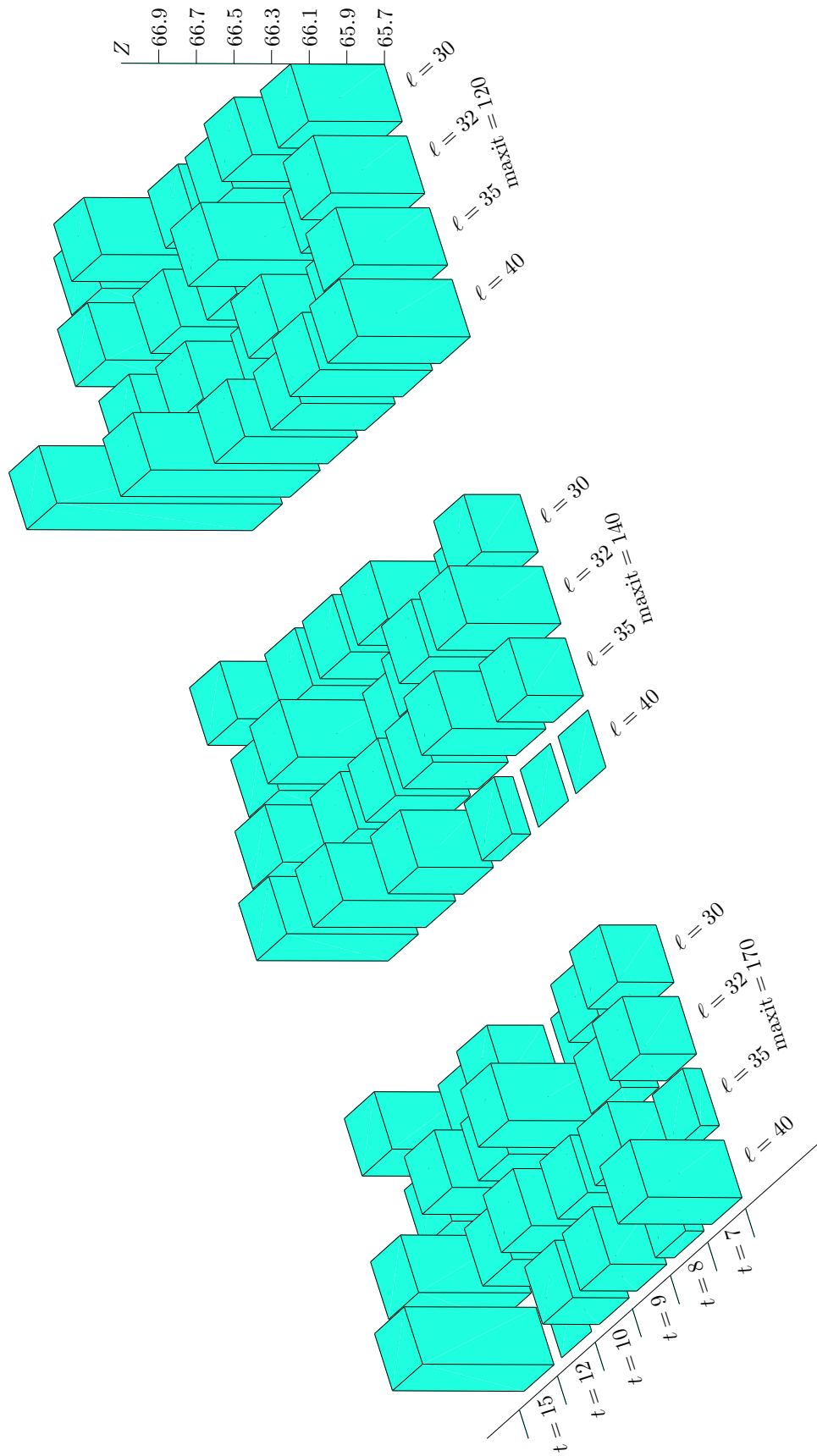


FIGURE 5.6: Mean objective function values, Z , obtained by the tabu search $\Delta(d, x)$ -colouring heuristic when applied to the random graph $G_{35,0.5}$ #2 for different sets of tabu parameter values.

obtained by the tabu search $\Delta(d, x)$ -colouring heuristic applied to these complete balanced multipartite graphs are shown in Table 5.12, together with the upper bounds on the entries in the Δ -chromatic sequence as obtained by the method described in §4.6 and those obtained by the colour degree heuristic. Similarly to the colour degree heuristic, the tabu search $\Delta(d, x)$ -colouring heuristic performs poorly in comparison with the method described in §4.6. Furthermore, a valid $\Delta(d, x)$ -colouring of a complete balanced multipartite graph for the first upper bound in Step 1 of Algorithm 9 was often not obtained during the execution of the algorithm.

The tabu search $\Delta(d, x)$ -colouring heuristic which commences by generating a random initial colouring and then attempts to improve it, does not perform well on graph structure classes where the specific structure of the graph limits the possible ways in which an optimal or even good colouring may be obtained. Examples of such structures are trees, cycles, wheels and complete balanced multipartite graphs. Although the number of possible ways to obtain an optimal colouring of a complete bipartite graph are also limited, the tabu search $\Delta(d, x)$ -colouring heuristic does obtain the optimal values in the Δ -chromatic sequence. To see why, suppose the partite sets of the bipartite graph are V_1 and V_2 . As soon as the tabu search $\Delta(d, x)$ -colouring heuristic attempts to find a valid $\Delta(0, 2)$ -colouring, a vertex v with maximum colour class induced degree is moved from the colour class with the larger colour class induced maximum degree, say C_1 , to the remaining colour class, say C_2 . Suppose $v \in V_2$. During each iteration of the tabu search $\Delta(d, x)$ -colouring heuristic from here onwards, by choosing a vertex with maximum colour class induced degree from the colour class with the larger colour class induced maximum degree, either a vertex in V_1 is moved from C_2 to C_1 or a vertex in V_2 is moved from C_1 to C_2 . Thus, when the tabu search $\Delta(d, x)$ -colouring heuristic terminates, $C_1 = V_1$ and $C_2 = V_2$.

The results of the tabu search $\Delta(d, x)$ -colouring heuristic when applied to the six graphs of order at most 49 from the benchmark graphs listed in Table F.8 are summarised in Tables 5.13 and 5.14. With the exceptions of $\bar{\chi}_0^\Delta$ (queen5.5) obtained during runs 1 and 3 of the tabu search $\Delta(d, x)$ -colouring heuristic and some of the upper bounds on the entries in the Δ -chromatic sequence of myciel6, the tabu search $\Delta(d, x)$ -colouring heuristic obtained either the same value as or a better upper bound for each entry in the Δ -chromatic sequences of the graphs listed in the first column of Table 5.13 than the colour degree heuristic. It is interesting, however, to note how much more execution time is necessary to obtain a better upper bound. For example, the colour degree heuristic obtained upper bounds on the entries in the Δ -chromatic sequences of queen7.7 within less than 10 seconds, while each execution of the tabu search $\Delta(d, x)$ -colouring heuristic applied to queen7.7 lasted more than 15 minutes.

The tabu search $\Delta(d, x)$ -colouring heuristic with parameter values $(\text{maxit}, \ell, t) = (140, 40, 8)$ was also applied to the generated random graphs of orders 20 and 35. The subsequent results are given in Table 5.15. Similarly to the benchmark graphs, with a few exceptions Algorithm 9 obtained either the same value as or a better upper bound than the colour degree heuristic for each entry in the Δ -chromatic sequences of the graphs listed in the first column of Table 5.15. Furthermore, the five different executions of Algorithm 9 applied to the same graph often resulted in similar sequences of values.

Finally, the random graph $G_{65,0.5}$ (see §F.4) generated first was used to determine a set of values for the parameters t , ℓ and maxit for the remaining graphs, *i.e.* the test graphs of orders at least 50, where the largest order of a test graph in Appendix F is 95 (myciel7 listed in Table F.8). Again, the objective was to minimize the sum of the entries in the sequence of upper bounds on the Δ -chromatic sequence of $G_{65,0.5} \#1$. The parameter t was varied between the values 6 and 15 when applying Algorithm 9 to $G_{65,0.5} \#1$, while the parameter ℓ [maxit , respectively] was varied over the set of values $\{30, 40, 50, 60, 75, 100\}$ [$\{140, 150, 165, 180, 200\}$, respectively]. As before, for each set of parameter values Algorithm 9 was executed on $G_{65,0.5} \#1$ a total of ten times and the mean value of the objective function values obtained over the ten runs for each set of parameter values was determined. However, the execution time of Algorithm 9 increases rapidly as the order of a test graph, the parameter ℓ , or the parameter maxit increases. For this reason the runs to determine a set of parameter values for the test graphs of order at least 50, were not executed sequentially on the Dell Latitude D620 Notebook as was done in the other two cases reported earlier in this section. Instead, the runs were executed in a computer laboratory on twenty-four 3 GB Pentium 4 desktop computers with 1 GB RAM each.

In order to provide the reader with an indication of the duration of the runs to determine a set of parameter values for the test graphs of order at least 50, some of the execution times of these runs on the 3 GB Pentium 4 desktop computers are listed in Table 5.16.

Graph type	# of graphs	Avg order	Avg size	Tabu search $\Delta(d, x)$ -colouring heuristic											
				Colour degree heuristic		Run 1		Run 2		Run 3		Run 4		Run 5	
				Avg time	# seq correct	Avg time	# seq correct	Avg time	# seq correct	Avg time	# seq correct	Avg time	# seq correct	Avg time	# seq correct
Cycles	6	31.8	31.8	0.404	6	4	34	3	32	3	35	3	31	4	
Wheels	6	31.8	61.7	0.854	6	4	111	4	100	4	96	3	100	4	
Complete graphs	6	31.8	579.0	9.457	6	6	947	6	947	6	947	6	952	6	
Complete bipartite	2	42.0	196.0	0.117	2	2	15	2	16	2	16	2	14	2	

TABLE 5.10: The results obtained by the tabu search $\Delta(d, x)$ -colouring heuristic when applied to the graphs of order at most 50 from various structure classes. In all cases the tabu parameter values were taken as $\text{maxit} = 140$, $\ell = 40$ and $t = 8$. All times are given in seconds.

Graph type	Avg order	Avg size	Average # of iterations during execution of the tabu search $\Delta(d, x)$ -colouring heuristic													
			Run 1		Run 2		Run 3		Run 4		Run 5					
			χ_0^Δ	seq.	χ_0^Δ	seq.	χ_1^Δ	seq.	χ_0^Δ	seq.	χ_1^Δ	seq.				
Cycles	31.8	31.8	135	4	140	154	6	159	134	6	140	132	6	137	6	119
Wheels	31.8	61.7	286	158	474	276	185	494	269	164	465	273	182	485	278	472
Complete graphs	31.8	579.0	151	148	2185	152	147	2183	151	148	2182	151	148	2180	152	2191
Complete bipartite	42.0	196.0	104	0	104	101	0	101	113	0	113	110	0	110	104	104

TABLE 5.11: The average number of iterations performed during the execution of the tabu search $\Delta(d, x)$ -colouring heuristic applied to graphs of order at most 50 from structure classes. In all cases the tabu parameter values were taken as $\text{maxit} = 140$, $\ell = 40$ and $t = 8$. The average number of iterations performed by the tabu search $\Delta(d, x)$ -colouring heuristic to obtain an upper bound on χ_0^Δ [χ_1^Δ , the entire Δ -chromatic sequence, respectively] are shown in the columns labelled χ_0^Δ [χ_1^Δ , seq., respectively] for each of the five runs.

Graph	Method of §4.6	Colour degree	Upper bounds on the first eight entries in the Δ -chromatic sequence									
			Run 1		Run 2		Run 3		Run 4		Run 5	
			χ_0^Δ	seq.	χ_0^Δ	seq.	χ_1^Δ	seq.	χ_0^Δ	seq.	χ_1^Δ	seq.
$K_{4 \times 4}$	44432222	48442222	55433322	55433322	55433322	55433322	55433322	55433322	55433322	55433322	55433322	
$K_{4 \times 7}$	4444432	414864442	811855443	87854433	99855443	99855443	8785543-	8785543-	8785543-	8785543-	8785543-	
$K_{4 \times 11}$	4444444	4221286644	1091177655	7977-55	109117-55	109117-55	107107-5	107107-5	107107-5	107107-5	107107-5	
$K_{7 \times 4}$	77764443	713774444	811774544	98875544	8996-543	8996-543	8996-543	8996-543	8996-543	8996-543	8996-543	
$K_{11 \times 4}$	111196665	11211116666	1620-977-6	16211398-66	17-1297--6	17-1297--6	20-14986-6	20-14986-6	20-14986-6	20-14986-6	20-14986-6	

TABLE 5.12: Upper bounds on the values in the Δ -chromatic sequence of complete balanced multipartite graphs obtained by the heuristic method of §4.6 listed in column 2, obtained by the colour degree heuristic listed in column 3 and obtained by the tabu search $\Delta(d, x)$ -colouring heuristic listed in columns 4-8 for the five runs. A “-” in the last columns indicates that no valid $\Delta(d, x)$ -colouring during the execution of the tabu search $\Delta(d, x)$ -colouring heuristic could be found. In all cases the tabu parameter values were taken as $\text{maxit} = 140$, $\ell = 40$ and $t = 8$.

Values of and upper bounds on the first eight entries in the Δ -chromatic sequence

Graph	Order	Size	χ	Tabu search $\Delta(d, x)$ -colouring heuristic							
				Colour degree							
				Run 1	Run 2	Run 3	Run 4	Run 5			
queen5_5	25	160	5	65433332	56433332	65433332	55433332	55433332	55433332		
queen6_6	36	290	7	88754333	76544333	86544333	86544333	86544333	86544333		
queen7_7	49	476	7	108754433	97654433	97654433	98654433	97654433	97654433		
myciel4	11	20	4	422221	422221	422221	422221	422221	422221		
myciel5	23	71	5	53332222	53332222	53332222	53332222	53332222	53332222		
myciel6	47	236	6	64333333	64333333	64333333	64333333	64333333	64333333		

TABLE 5.13: Values of and upper bounds on the values in the Δ -chromatic sequence of the benchmark graphs of order at most 49 obtained by the colour degree heuristic listed in column 5 and obtained by the tabu search $\Delta(d, x)$ -colouring heuristic listed in columns 6–10 for the five runs. In all cases the tabu parameter values were taken as $\text{maxit} = 140$, $\ell = 40$ and $t = 8$.

Average # of iterations during execution of Algorithm 9

Graph	Order	Size	Time: CDH	Time: Algorithm 9																								
				Run 1					Run 2					Run 3					Run 4					Run 5				
				χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.	χ_0^Δ	seq.					
queen5_5	25	160	1.03	113.08	121.92	113.10	128.76	124.28	283	944	391	998	306	948	437	1106	366	1047										
queen6_6	36	290	3.22	371.34	427.21	374.45	359.31	355.09	366	1394	486	1574	397	1458	366	1355	379	1332										
queen7_7	49	476	8.70	939.42	947.98	911.57	938.53	909.24	577	1881	529	1895	552	1806	589	1869	554	1814										
myciel4	11	20	0.03	1.44	1.56	1.53	1.48	1.55	149	151	148	154	151	156	145	154	149	156										
myciel5	23	71	0.45	17.58	19.19	17.64	15.77	20.89	202	283	194	270	189	264	183	258	195	312										
myciel6	47	236	3.95	376.98	361.82	389.16	373.00	365.34	446	1069	411	1019	433	1107	410	1033	448	1004										

TABLE 5.14: The execution times (in seconds) to obtain upper bounds on the entire Δ -chromatic sequences of the benchmark graphs of order at most 49 via the colour degree heuristic in comparison with the relevant execution times of each of the five runs of the tabu search $\Delta(d, x)$ -colouring heuristic. The average number of iterations performed by the tabu search $\Delta(d, x)$ -colouring heuristic to obtain an upper bound on χ_0^Δ [the entire Δ -chromatic sequence, respectively] for each of the five runs are shown in the columns labelled χ_0^Δ [seq., respectively]. In all cases the tabu parameter values were taken as $\text{maxit} = 140$, $\ell = 40$ and $t = 8$.

Graph	Colour degree heuristic					Tabu search $\Delta(d, x)$ -colouring heuristic					Running time (s)							
						Run 1	Run 2	Run 3	Run 4	Run 5	CDH	Tabu						
						Run 1	Run 2	Run 3	Run 4	Run 5	CDH	Tabu						
$G_{20,0.2}$ #1	3	3	2	2	2	3	3	2	2	2	3	3	2	2	2	2	0.17	5.54
$G_{20,0.2}$ #2	4	4	3	2	2	4	4	3	2	2	4	4	3	2	2	2	0.23	13.80
$G_{20,0.2}$ #3	4	3	2	2	2	3	3	2	2	2	3	3	2	2	2	1	0.19	11.81
$G_{20,0.2}$ #4	3	3	2	2	2	3	3	2	2	2	3	3	2	2	2	2	0.16	6.58
$G_{20,0.2}$ #5	4	3	2	2	2	4	3	2	2	2	4	3	2	2	2	2	0.19	12.73
$G_{20,0.5}$ #1	6	5	4	3	3	5	4	3	3	2	5	4	3	3	2	2	0.47	35.28
$G_{20,0.5}$ #2	6	5	4	3	3	6	4	3	3	2	6	4	3	3	2	2	0.39	34.37
$G_{20,0.5}$ #3	6	5	4	3	3	6	4	3	3	2	6	4	3	3	2	2	0.41	42.80
$G_{20,0.5}$ #4	6	5	4	3	3	6	4	3	3	2	6	4	3	3	2	2	0.41	34.73
$G_{20,0.5}$ #5	6	5	4	3	3	6	4	3	3	2	6	4	3	3	2	2	0.41	37.15
$G_{20,0.8}$ #1	9	9	5	4	3	9	7	5	4	3	9	7	5	4	3	3	0.72	99.71
$G_{20,0.8}$ #2	9	7	5	4	3	9	6	5	4	3	9	6	5	4	3	3	0.69	67.77
$G_{20,0.8}$ #3	10	8	5	4	3	10	6	5	4	3	10	6	5	4	3	3	0.70	78.92
$G_{20,0.8}$ #4	9	8	6	4	3	9	7	5	4	3	9	7	5	4	3	3	0.72	88.27
$G_{20,0.8}$ #5	9	8	5	4	3	9	7	5	4	3	9	6	5	4	3	3	0.72	85.67
$G_{35,0.2}$ #1	5	4	3	2	2	5	4	3	3	2	5	4	3	3	2	2	1.28	119.82
$G_{35,0.2}$ #2	4	4	3	2	2	4	4	3	3	2	4	4	3	3	2	2	1.25	128.24
$G_{35,0.2}$ #3	5	4	3	3	2	5	4	3	3	2	5	4	3	3	2	2	1.52	124.05
$G_{35,0.2}$ #4	4	4	3	3	2	4	4	3	3	2	4	4	3	3	2	2	1.80	118.77
$G_{35,0.2}$ #5	5	4	3	3	2	5	4	3	3	2	5	4	3	3	2	2	1.48	98.93
$G_{35,0.5}$ #1	10	7	6	5	4	10	7	6	5	4	10	7	6	5	4	3	3.31	375.87
$G_{35,0.5}$ #2	9	7	6	5	4	9	6	5	4	3	9	6	5	4	3	3	2.99	364.78
$G_{35,0.5}$ #3	9	7	6	5	3	9	6	5	4	3	9	6	5	4	3	3	3.30	365.74
$G_{35,0.5}$ #4	8	8	6	5	4	8	6	5	4	3	8	6	5	4	3	3	3.00	371.81
$G_{35,0.5}$ #5	8	7	6	5	4	8	6	5	4	3	8	6	5	4	3	3	2.94	349.46
$G_{35,0.8}$ #1	16	13	9	7	6	16	10	8	7	6	16	10	8	7	6	5	5.61	746.82
$G_{35,0.8}$ #2	13	11	10	7	6	13	10	8	7	6	13	10	8	7	6	5	5.19	699.20
$G_{35,0.8}$ #3	13	13	8	7	5	13	9	8	7	5	13	9	8	7	5	4	5.09	646.05
$G_{35,0.8}$ #4	15	13	9	7	6	15	10	8	7	6	15	10	8	7	6	5	5.16	670.01
$G_{35,0.8}$ #5	15	13	9	7	6	15	10	8	7	6	15	10	8	7	6	5	5.13	658.65

TABLE 5.15: Values of and upper bounds on the values in the Δ -chromatic sequence of the generated random graphs of orders 20 and 35 obtained by the colour degree heuristic listed in column 2 and obtained by the tabu search $\Delta(d, x)$ -colouring heuristic listed in columns 3–7 for the five runs. In all cases $(\max\{\ell, t\}) = (140, 40, 8)$. In the last two columns the execution times [average execution times over the five runs, respectively] (in seconds) to obtain upper bounds on the entire Δ -chromatic sequences of each graph via the colour degree heuristic [the tabu search $\Delta(d, x)$ -colouring heuristic, respectively] are shown.

	$\ell = 30$	$\ell = 40$	$\ell = 50$	$\ell = 60$	$\ell = 75$	$\ell = 100$
Maxit = 140	1.13 hours	1.37 hours	1.59 hours	1.78 hours	2.13 hours	2.48 hours
Maxit = 165	1.25 hours	1.53 hours	1.81 hours	2.00 hours	2.30 hours	2.75 hours
Maxit = 200	1.44 hours	1.74 hours	1.99 hours	2.17 hours	2.63 hours	3.13 hours

TABLE 5.16: Average execution times of the runs to determine a set of parameter values to be used when the tabu search $\Delta(d, x)$ -colouring heuristic is applied to the test graphs of order at least 50. Each time value in the table is the mean of the durations of the ten runs for each set of parameter values, where in each case the value of t was 6.

As is evident from Table 5.16, the execution times when the tabu search $\Delta(d, x)$ -colouring heuristic with maxit = 200 was applied to $G_{65,0.5} \#1$, were longer than one and a half hours and some even lasted up to 3 hours ($\ell = 100$). Therefore, sets of parameter values with maxit greater than 200 were not considered when the tabu search $\Delta(d, x)$ -colouring heuristic was to be applied to the test graphs of order at least 50. A selection of the mean objective function values for sets of parameter values that were considered is shown in Figure 5.7.

The mean objective function values for maxit = 200 varied between 141.9 and 144.5 with eight values larger than 144.0, while the mean objective function values for maxit = 180 varied between 142.7 and 145.6 with only one value larger than 144.9, indicating only a slight improvement on the objective function if maxit increases from 180 to 200. However, the execution time increased with up to 10.1% if maxit was increased from 180 to 200. It was therefore decided to select the value maxit = 180. For the two smaller order groups of graphs the final set of parameter values was selected by choosing the smallest mean objective function value and executing the tabu search $\Delta(d, x)$ -colouring heuristic a further 50 times for each set of parameter values for which this smallest mean objective function value was obtained. Since the effects of the randomness of the initial solution when the tabu search $\Delta(d, x)$ -colouring heuristic is applied to a graph of order 65 a total of ten times only may be more significant than for the smaller order graphs, it does not seem sufficient enough to execute the tabu search $\Delta(d, x)$ -colouring heuristic a further 50 times only for those sets of parameters for which a smallest mean objective function value was obtained after the initial 10 runs. It was therefore decided to employ trendlines to determine appropriate values for t and ℓ . Trendlines from MICROSOFT EXCEL was used to fit a parabola to the data sets (t and Z) for each value of ℓ and maxit = 180. Four of these trendlines, namely for $\ell = 30$, $\ell = 40$, $\ell = 50$ and $\ell = 60$, are given in Figure 5.8. For comparison purposes the trendlines for $\ell = 40$, $\ell = 60$ and maxit = 200 are also included in Figure 5.8.

All four trendlines for maxit = 180 in Figure 5.8 achieve a minimum around $t = 8$ and thus the value of t was chosen as 8. The minimum of a trendline amongst the four trendlines in (a)–(d) appears to be the smallest for $\ell = 40$ and $\ell = 50$. This smallest minimum value of a trendline is just above 143, while the minimum value of the trendline for maxit = 200 and $\ell = 40$ [$\ell = 60$, respectively] in Figure 5.8(e) [Figure 5.8(f), respectively] is approximately 142.3 [143.4, respectively], so that on average only one value in the sequence of upper bounds on the Δ -chromatic sequence obtained by the tabu search $\Delta(d, x)$ -colouring heuristic for maxit = 200 is one less than those for maxit = 180. This observation reiterates the above remark that if maxit is increased from 180 to 200, only a slight improvement on the mean objective function value is obtained. Finally, since the execution time of the tabu search $\Delta(d, x)$ -colouring heuristic increases as the parameter ℓ increases, the value of ℓ was chosen as 40. To summarise, the set of parameter values (maxit, ℓ , t) = (180, 40, 8) was chosen to be implemented when the tabu search $\Delta(d, x)$ -colouring heuristic was applied to the test graphs of order at least 50.

The tabu search $\Delta(d, x)$ -colouring heuristic with parameter values (maxit, ℓ , t) = (180, 40, 8) was first applied to the remaining proper colouring benchmark graphs. The results thus obtained are listed in Table 5.18. As before, the five different executions of the tabu search $\Delta(d, x)$ -colouring heuristic applied to the same graph often resulted in similar sequences of values and, with a few exceptions for relatively small values of d , the tabu search $\Delta(d, x)$ -colouring heuristic obtained either the same value as or a better upper bound than the colour degree heuristic for each entry in the Δ -chromatic sequences of the graphs listed in the first column of Table 5.18. This result was expected, because of the fact that a higher level of computation is invested in the tabu search metaheuristic than in the greedy colour degree heuristic.

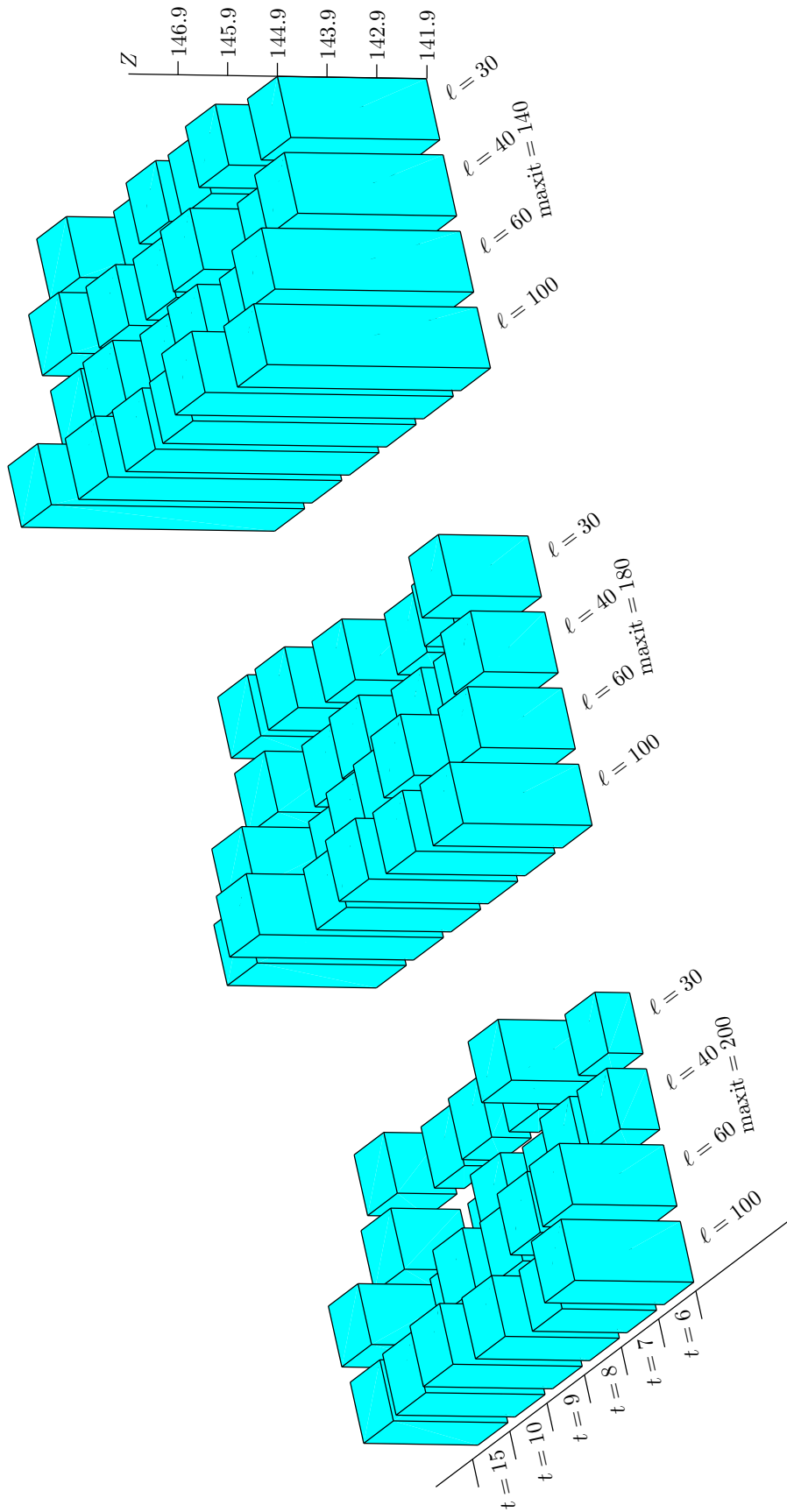


FIGURE 5.7: Mean objective function values, Z , obtained by the tabu search $\Delta(d, x)$ -colouring heuristic when applied to the random graph $G_{65,0.5}$ #1 for different sets of tabu parameter values.

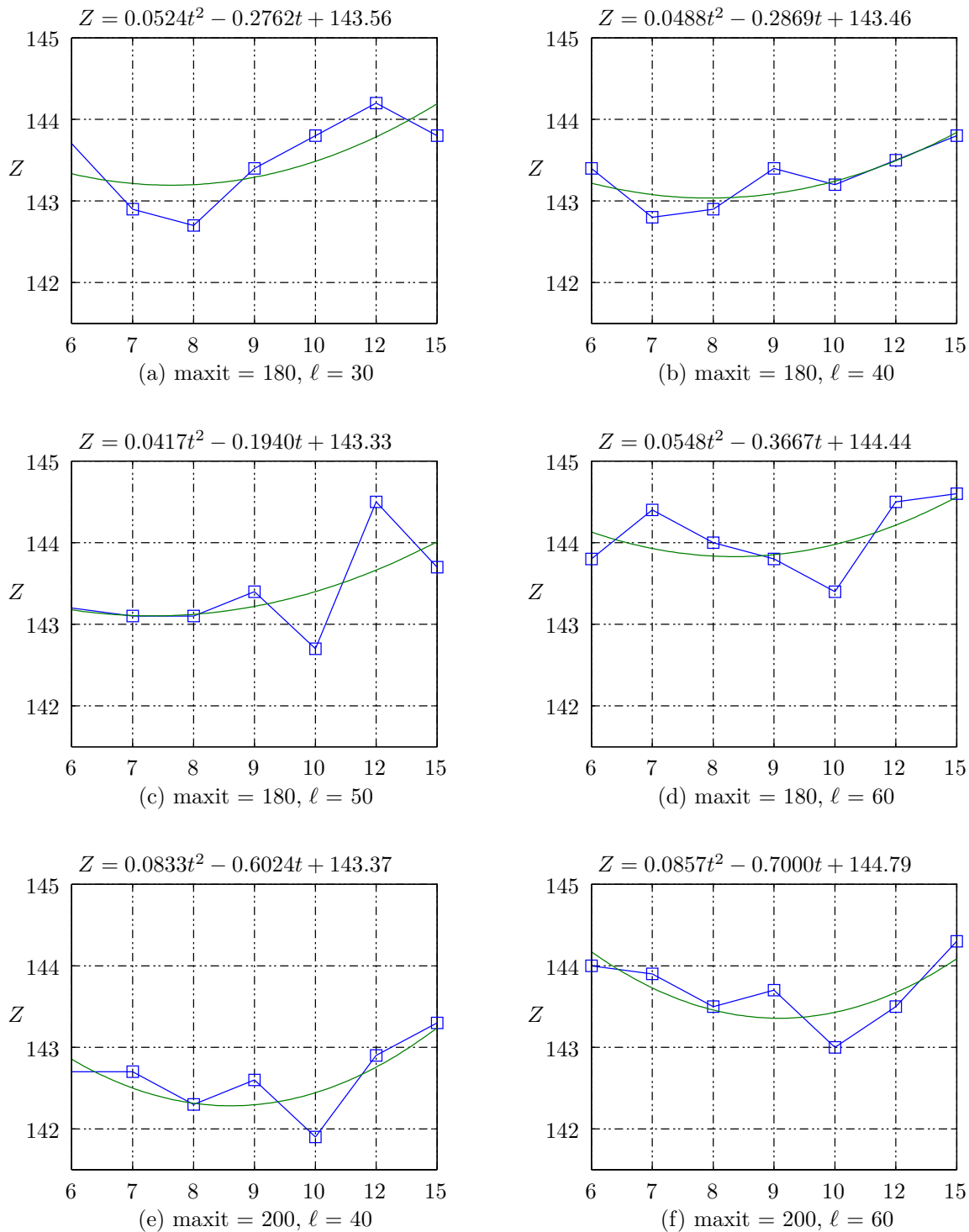


FIGURE 5.8: Trendlines for some of the data sets generated to determine values for the tabu parameters t and ℓ for graphs of order at least 50 after maxit was selected to be 180. In (a)–(d) the trendlines for $\ell = 30, 40, 50$ and 60 for maxit = 180 are given, while the two trendlines in (e) and (f) are only given for comparison purposes.

It was found, however, for the three graphs myciel7, jean and david and for values of d around 6, 7 or 8 when the number of colours were small (2, 3 or 4), that the tabu search $\Delta(d, x)$ -colouring heuristic was outperformed by the colour degree heuristic for a number of entries until the first 2 in the sequence

was obtained by the tabu search $\Delta(d, x)$ -colouring heuristic. As the number of colours decreases, the number of possible partitions of the vertices leading to a valid $\Delta(d, x)$ -colouring possibly decreases as well. Furthermore, the difference between the order of these graphs and the graph $G_{65,0.5} \#1$ is at least 15. It may be that the chosen value of maxit is too small in these cases to find a valid $\Delta(d, x)$ -colouring in only a few colours.

The tabu search $\Delta(d, x)$ -colouring heuristic with parameter values $(\text{maxit}, \ell, t) = (180, 40, 8)$ was also applied to the random graphs of orders 50, 65 and 85. The subsequent results are given in Tables 5.19 and 5.20. In the case of random graphs of order 50, the tabu search $\Delta(d, x)$ -colouring heuristic often improved quite significantly upon the colour degree heuristic. For example, all five runs of the tabu search $\Delta(d, x)$ -colouring heuristic applied to the graph $G_{50,0.8} \#5$ resulted in better upper bounds on $\chi_d^\Delta(G_{50,0.8} \#5)$ for $d = 1, \dots, 5$ than those obtained by the colour degree heuristic. In particular, $\overline{\chi}_0^\Delta(G_{50,0.8} \#5)$ obtained by the colour degree heuristic is 20, while 18, 17, 17, 18 and 18 were obtained during the five runs of the tabu search $\Delta(d, x)$ -colouring heuristic. Also, for $d = 2, \dots, 5$ the sequence $\overline{17} \overline{12} \overline{9} \overline{8}$ of upper bounds was obtained by the colour degree heuristic, while the five runs of the tabu search $\Delta(d, x)$ -colouring heuristic resulted in the sequences $\overline{12} \overline{11} \overline{9} \overline{7}$, $\overline{13} \overline{11} \overline{8} \overline{7}$, $\overline{13} \overline{10} \overline{8} \overline{7}$, $\overline{13} \overline{10} \overline{8} \overline{7}$ and $\overline{13} \overline{10} \overline{8} \overline{7}$. For $d \geq 6$ in these cases the tabu search $\Delta(d, x)$ -colouring heuristic either resulted in the same upper bound or a slightly better one than that obtained by the colour degree heuristic. The tabu search $\Delta(d, x)$ -colouring heuristic also often obtained a valid $\Delta(d, x)$ -colouring in more than one colour fewer than the colour degree heuristic for the same value of $d \geq 1$ for the random graphs of orders 65 and 85, especially those with density 0.8. However, the tabu search $\Delta(d, x)$ -colouring heuristic was frequently outperformed by the colour degree heuristic for $d = 0$, especially for the random graphs of order 85. Similar to the case where a valid $\Delta(d, x)$ -colouring in only a few colours are sought, there may be fewer possible ways of obtaining a valid $\Delta(0, x)$ -colouring than when $d \geq 1$. Better upper bounds on χ_0^Δ for the random graphs of orders 50, 65 and 85 may be obtained when maxit is increased beyond 180 for this value of d . The only other values in Tables 5.19 and 5.20 where the colour degree heuristic obtained a better upper bound than the tabu search $\Delta(d, x)$ -colouring heuristic were $\overline{\chi}_5^\Delta(G_{50,0.2} \#4)$ during run 3, $\overline{\chi}_7^\Delta(G_{65,0.2} \#2)$ during run 2, $\overline{\chi}_7^\Delta(G_{65,0.2} \#3)$ during runs 1, 3 and 5, in which case a valid colouring in only two colours was sought, as well as $\overline{\chi}_6^\Delta(G_{50,0.8} \#3)$ during run 5 and seven more values in the sequences of upper bounds for the random graphs of order 85 and density 0.2.

The software implementation of the tabu search $\Delta(d, x)$ -colouring heuristic was adapted to abort execution when upper bounds on the values in the Δ -chromatic sequence of a particular graph could not be obtained within a time limit of three hours. In the case of the random graphs of order 85 with a density of 0.8, a complete sequence (*i.e.* up to the first 1 in the sequence) of upper bounds on the values in the Δ -chromatic sequence of a graph could not be obtained within the time limit of three hours for any of the five graphs during any of the runs of the tabu search $\Delta(d, x)$ -colouring heuristic. In each case, the time limit of three hours was reached as indicated in Table 5.17.

Graph	Run 1	Run 2	Run 3	Run 4	Run 5
$G_{85,0.8} \#1$	$d = 8$	$d = 8$	$d = 6$	$d = 9$	$d = 9$
$G_{85,0.8} \#2$	$d = 9$	$d = 7$	$d = 8$	$d = 8$	$d = 10$
$G_{85,0.8} \#3$	$d = 8$	$d = 7$	$d = 10$	$d = 10$	$d = 7$
$G_{85,0.8} \#4$	$d = 11$	$d = 8$	$d = 7$	$d = 9$	$d = 8$
$G_{85,0.8} \#5$	$d = 10$	$d = 9$	$d = 8$	$d = 12$	$d = 10$

TABLE 5.17: *The position (i.e. the particular value of d) in the sequence of upper bounds on the values in the Δ -chromatic sequence of the random graphs of order 85 and density 0.8 when the time out condition of three hours was reached during execution of the tabu search $\Delta(d, x)$ -colouring heuristic.*

The tabu search $\Delta(d, x)$ -colouring heuristic applied to the random graphs of orders 50 and 65, and of order 85 with densities 0.2 and 0.5, obtained a value of 3 and 2 in the sequence of upper bounds on the values in the Δ -chromatic sequence of a graph around the same values for d as the colour degree heuristic. Therefore, it seems that the set of parameter values $(\text{maxit}, \ell, t) = (180, 40, 8)$ are good enough to obtain a valid $\Delta(d, x)$ -colouring in only a few colours for these graphs. However, it is possible that the random graphs of order 85 with density 0.8 may obtain a value of 3 or 2 in the sequence of upper bounds on the values in the Δ -chromatic sequence of a graph at a higher value for d than the colour degree heuristic, as in the case of the three benchmark graphs `myciel_7`, `jean` and `dauid`.

Graph	p	Size	χ	Values of and upper bounds on the first seven entries in the Δ -chromatic sequence										Running time (min)	
				Tabu search $\Delta(d, x)$ -colouring heuristic										CDH	Tabu
				Colour degree heuristic	Run 1	Run 2	Run 3	Run 4	Run 5	Run 5	Run 5	Run 5	Run 5		
queen8.8	64	728	9	11 9 8 6 5 5 4	11 8 7 5 5 4 4	10 8 6 5 5 4 4	11 9 7 5 5 4 4	11 8 7 6 5 4 4	11 8 7 5 5 4 4	10 8 7 5 5 4 4	10 8 7 5 5 4 4	0.32	41.1		
myciel7	95	755	7	7 6 6 5 4 4 3	7 6 5 4 4 4 4	7 5 5 4 4 4 4	7 6 5 4 4 4 4	7 6 5 4 4 4 4	7 6 5 4 4 4 4	7 6 5 4 4 4 4	0.71	91.6			
huck	74	301	11	11 6 5 4 4 3 3	11 6 5 4 3 3 3	11 6 4 4 3 3 3	11 6 5 4 4 3 3	12 6 4 4 4 3 3	11 6 4 4 4 3 3	11 6 4 4 4 3 3	0.27	25.3			
jean	80	254	10	10 7 5 4 3 3 3	10 6 4 4 3 3 3	10 5 4 4 3 3 3	10 6 5 4 3 3 3	10 6 5 4 4 3 3	10 6 5 4 4 3 3	11 6 4 4 3 3 3	0.30	25.9			
david	87	406	11	11 7 6 5 4 3 3	11 7 5 5 4 4 4	11 7 5 4 5 4 4	11 7 5 5 4 4 4	11 7 5 4 4 4 4	11 7 5 5 4 4 4	12 7 5 5 4 4 4	0.55	53.6			

TABLE 5.18: Values of and upper bounds on the values in the Δ -chromatic sequence of the remaining benchmark graphs (of order at least 50) obtained by the colour degree heuristic listed in column 5 and those obtained by the tabu search $\Delta(d, x)$ -colouring heuristic listed in columns 6–10 for the five runs. In all cases the tabu parameter values were taken as $\text{maxit} = 180$, $\ell = 40$ and $t = 8$. In the last two columns the execution times [average execution times over the five runs, respectively] (in minutes) to obtain upper bounds on the entire Δ -chromatic sequences of each graph via the colour degree heuristic [the tabu search $\Delta(d, x)$ -colouring heuristic, respectively] are shown.

Graph	Colour degree heuristic	Values of and upper bounds on the first eight entries in the Δ -chromatic sequence										Running time (s)	
		Tabu search $\Delta(d, x)$ -colouring heuristic										CDH	Tabu
		Run 1	Run 2	Run 3	Run 4	Run 5	Run 5	Run 5	Run 5	Run 5	Run 5		
$G_{50,0.2}$ #1	6 5 4 4 3 3 2 2	6 5 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	4.67	535
$G_{50,0.2}$ #2	6 5 4 4 3 3 2 2	6 4 4 3 3 2 2 2	6 5 4 3 3 2 2 2	6 5 3 3 3 2 2 2	6 5 3 3 3 2 2 2	6 4 4 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	4.69	554
$G_{50,0.2}$ #3	6 5 4 4 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	4.69	590
$G_{50,0.2}$ #4	5 4 4 3 3 2 2 2	6 4 3 3 3 2 2 2	6 4 3 3 3 2 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	3.92	405
$G_{50,0.2}$ #5	6 5 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 3 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 4 4 3 3 3 2 2	6 5 4 3 3 3 2 2	6 5 4 3 3 3 2 2	4.64	499
$G_{50,0.5}$ #1	11 9 8 6 6 5 4 4	10 8 6 6 5 5 4 4	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 6 5 5 4 4 4	10 8 6 5 5 4 4 4	11.06	1519
$G_{50,0.5}$ #2	10 10 8 7 6 5 4 4	11 8 7 6 5 4 4 4	11 8 7 6 5 4 4 4	11 8 7 6 5 4 4 4	11 8 7 6 5 4 4 4	11 8 7 6 5 4 4 4	11 8 7 6 5 4 4 4	11 8 7 6 5 4 4 4	11 8 7 6 5 4 4 4	10 9 7 5 5 4 4 4	10 9 7 5 5 4 4 4	11.89	1576
$G_{50,0.5}$ #3	10 9 8 6 5 5 4 4	10 9 6 5 5 4 4 3	11 8 7 5 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	11.80	1529
$G_{50,0.5}$ #4	11 10 8 6 5 5 4 4	10 8 7 6 5 4 4 4	10 8 7 5 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 4	10 8 7 6 5 4 4 3	11 8 7 6 5 4 4 3	11 8 7 6 5 4 4 3	11.93	1554
$G_{50,0.5}$ #5	11 9 8 7 5 5 4 4	10 8 6 5 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 7 6 5 4 4 3	10 8 6 6 5 4 4 3	10 8 6 6 5 4 4 3	11.01	1509
$G_{50,0.8}$ #1	20 20 13 10 8 7 6 6	21 14 11 9 8 7 6 5	21 15 10 9 8 7 6 5	19 14 10 9 7 6 5	19 14 10 9 7 6 5	19 14 10 9 7 6 5	19 14 10 9 7 6 5	19 14 10 9 7 6 5	19 14 10 9 7 6 5	21 14 11 9 8 7 6 5	21 14 11 9 8 7 6 5	20.39	2931
$G_{50,0.8}$ #2	17 19 12 9 8 6 6 5	18 12 10 8 7 6 6 5	18 14 10 9 7 6 5	18 13 10 8 7 6 5	18 13 10 8 7 6 5	18 13 10 8 7 6 5	18 13 10 8 7 6 5	18 13 10 8 7 6 5	18 13 10 8 7 6 5	19 13 10 9 7 6 5	19 13 10 9 7 6 5	19.27	2990
$G_{50,0.8}$ #3	20 18 12 9 8 7 6 5	19 13 10 8 8 7 6 5	21 13 10 9 7 6 5	18 13 10 9 7 6 5	18 13 10 8 8 7 6 5	18 13 10 8 8 7 6 5	18 13 10 8 8 7 6 5	18 13 10 8 8 7 6 5	18 13 10 8 8 7 6 5	19 13 10 8 7 7 5	19 13 10 8 7 7 5	20.12	2914
$G_{50,0.8}$ #4	18 17 12 10 8 7 6 5	19 13 10 9 8 7 6 5	19 13 10 9 8 7 6 5	18 13 10 8 8 7 6 5	19 13 11 9 8 7 6 5	19 13 11 9 8 7 6 5	18 13 10 9 7 6 5	18 13 10 9 7 6 5	18 13 10 9 7 6 5	19 13 10 9 7 6 5	19 13 10 9 7 6 5	19.33	2988
$G_{50,0.8}$ #5	20 17 12 9 8 7 6 5	18 12 11 9 7 7 6 5	18 12 11 9 7 7 6 5	17 13 11 8 7 6 5	17 13 11 8 7 6 5	17 13 11 8 7 6 5	17 13 11 8 7 6 5	17 13 11 8 7 6 5	17 13 11 8 7 6 5	18 13 10 8 7 6 5	18 13 10 8 7 6 5	19.17	2924

TABLE 5.19: Values of and upper bounds on the values in the Δ -chromatic sequence of the generated random graphs of order 50 obtained by the colour degree heuristic listed in column 2 and those obtained by the tabu search $\Delta(d, x)$ -colouring heuristic listed in columns 3–7 for the five runs. In all cases (maxit, ℓ, t) = (180, 40, 8). In the last two columns the execution times [average execution times over the five runs, respectively] (in seconds) to obtain upper bounds on the entire Δ -chromatic sequences of each graph via the colour degree heuristic [the tabu search $\Delta(d, x)$ -colouring heuristic, respectively] are shown.

Graph	Values of and upper bounds on the first six entries in the Δ -chromatic sequence						Running time (min)
	Tabu search $\Delta(d, x)$ -colouring heuristic						
	Run 1	Run 2	Run 3	Run 4	Run 5	CDH	Tabu
$G_{65,0.2} \#1$	7 5 4 4 3 3	7 5 4 3 3 3	8 5 4 4 3 3	7 5 4 4 3 3	8 6 5 4 3 3	0.21	23
$G_{65,0.2} \#2$	6 5 4 4 3 3	7 5 4 3 3 3	7 6 4 3 3 3	7 5 4 3 3 3	7 5 4 3 3 3	0.18	19
$G_{65,0.2} \#3$	7 6 5 4 3 3	7 5 4 4 3 3	7 5 4 3 3 3	7 6 4 4 3 3	7 6 4 3 3 3	0.19	23
$G_{65,0.2} \#4$	7 6 5 4 4 3	7 5 4 4 3 3	7 5 4 4 3 3	7 6 4 4 3 3	7 5 4 3 3 3	0.21	23
$G_{65,0.2} \#5$	6 6 4 4 4 3	7 5 4 3 3 3	7 6 4 3 3 3	8 5 4 4 3 3	7 5 4 4 3 3	0.18	23
$G_{65,0.5} \#1$	14 12 11 8 7 6	13 11 8 7 6 5	14 10 9 7 6 5	14 10 8 7 7 6	14 10 9 8 6 6	0.53	71
$G_{65,0.5} \#2$	13 11 10 7 6 6	13 10 8 7 6 5	13 10 8 7 6 5	12 10 8 7 6 5	13 10 8 7 6 5	0.52	65
$G_{65,0.5} \#3$	13 11 11 8 6 6	13 10 8 7 6 5	13 10 8 7 6 5	12 10 8 7 6 5	13 10 8 7 6 5	0.52	68
$G_{65,0.5} \#4$	12 11 10 8 6 6	13 10 8 7 6 5	13 10 8 7 6 5	13 9 8 7 6 5	13 10 8 7 6 5	0.52	62
$G_{65,0.5} \#5$	14 12 9 8 7 6	13 10 8 7 6 5	13 10 8 7 6 5	13 10 8 7 6 5	13 10 8 7 6 5	0.55	70
$G_{65,0.8} \#1$	22 22 16 12 10 9	24 16 13 11 9 8	24 16 13 12 9 8	24 17 14 11 10 8	28 16 13 10 9 8	0.89	120
$G_{65,0.8} \#2$	23 21 16 12 10 9	24 17 13 11 9 9	24 17 13 11 9 8	24 17 13 11 10 8	24 18 13 11 9 8	0.89	119
$G_{65,0.8} \#3$	23 21 15 11 9 8	29 17 13 10 9 8	23 16 13 10 9 8	24 17 13 11 9 8	24 16 13 11 9 8	0.86	116
$G_{65,0.8} \#4$	22 21 15 13 10 9	24 17 13 10 10 8	22 17 13 11 9 8	31 17 13 11 10 9	23 17 13 11 9 8	0.89	120
$G_{65,0.8} \#5$	23 21 15 12 10 9	23 16 12 10 9 8	22 16 13 10 9 8	22 17 13 10 9 8	22 15 12 11 9 8	0.89	121
$G_{85,0.2} \#1$	8 7 6 5 4 3	9 7 6 4 4 3	9 7 6 4 4 3	9 7 5 4 4 3	9 7 5 4 4 3	0.54	64
$G_{85,0.2} \#2$	8 6 6 5 4 4	8 7 5 4 4 3	8 6 5 4 4 3	9 7 5 4 4 4	8 6 5 4 4 3	0.49	64
$G_{85,0.2} \#3$	7 7 5 5 4 4	9 7 6 4 4 3	9 6 6 4 4 3	9 7 5 4 4 3	9 8 6 4 4 3	0.55	59
$G_{85,0.2} \#4$	8 7 6 5 4 4	9 7 6 4 4 3	8 7 5 4 4 3	9 7 5 4 4 3	9 7 5 4 4 3	0.60	64
$G_{85,0.2} \#5$	8 7 6 5 4 4	8 7 5 4 4 3	8 7 5 4 4 3	8 6 5 4 4 3	9 7 5 4 4 3	0.54	60
$G_{85,0.5} \#1$	15 14 12 10 8 7	16 13 10 9 8 7	21 13 10 9 7 7	17 14 10 9 7 7	17 12 10 8 8 7	1.53	171
$G_{85,0.5} \#2$	17 15 12 9 8 7	16 12 10 9 7 7	18 13 10 9 7 6	17 13 10 9 7 7	17 12 10 9 7 7	1.47	171
$G_{85,0.5} \#3$	16 14 12 10 8 7	17 14 10 9 7 6	17 13 11 8 7 7	17 13 10 8 7 7	18 12 10 8 8 7	1.41	169
$G_{85,0.5} \#4$	16 14 12 10 8 7	16 13 10 9 8 7	18 12 10 9 7 7	17 12 10 9 7 7	16 13 11 9 8 7	1.47	169
$G_{85,0.5} \#5$	15 14 12 10 8 7	16 13 10 8 8 7	18 13 10 8 7 7	17 13 10 8 7 7	17 13 10 8 7 7	1.41	166
$G_{85,0.8} \#1$	27 26 19 15 13 11	32 23 17 13 11 10	30 22 16 13 11 10	31 22 18 13 12 10	35 22 17 13 12 10	2.45	TO
$G_{85,0.8} \#2$	27 26 20 15 12 11	32 22 17 14 11 10	31 21 17 13 11 10	30 21 16 13 12 10	36 21 16 14 12 10	2.39	TO
$G_{85,0.8} \#3$	29 27 21 15 13 11	31 22 17 13 11 10	36 22 17 13 11 10	33 21 18 13 12 10	31 21 16 13 11 10	2.46	TO
$G_{85,0.8} \#4$	29 27 20 15 13 11	36 22 18 14 12 10	31 22 17 14 12 10	32 22 17 14 12 10	32 22 17 14 12 10	2.54	TO
$G_{85,0.8} \#5$	31 27 21 15 12 11	36 22 17 14 12 10	32 23 17 14 12 10	34 23 18 15 12 10	35 23 17 15 12 11	2.56	TO

TABLE 5.20: Values of and upper bounds on the values in the Δ -chromatic sequences of the generated random graphs of orders 65 and 85 obtained by the colour degree heuristic listed in column 2 and those obtained by the tabu search $\Delta(d, x)$ -colouring heuristic listed in columns 3–7 for the five runs. In all cases $(\maxit, \ell, t) = (180, 40, 8)$. As before, the execution times (in minutes) to obtain upper bounds on the entire Δ -chromatic sequences of each graph are shown in the last two columns. A “TO” in the last column indicates that the time limit of three hours was reached during those particular runs.

In an attempt to speed up the execution of the tabu search $\Delta(d, x)$ -colouring heuristic, the upper bound x in Step 1 of Algorithm 9 was adapted to choose the minimum of the upper bound in Step 1 ($\lceil (\Delta(G) + 1)/(d + 1) \rceil$) and the upper bound that was obtained when the colour degree heuristic was applied to a graph G . In order to compare the adapted tabu search $\Delta(d, x)$ -colouring heuristic to Algorithm 9, the adapted version of the algorithm was also executed five times for each test graph in order to determine what possible improvement the adaptation to the algorithm might have.

Since the tabu search $\Delta(d, x)$ -colouring heuristic applied to the small graphs listed in §F.1 already obtained good results in relatively short execution times, the adapted version of the tabu search $\Delta(d, x)$ -colouring heuristic was not applied to these graphs. In the case of the trees listed in Table F.4, as well as the bipartite graphs listed in Table F.6, the colour degree heuristic obtained the exact value of χ_d^Δ for all values of d , which is 2 for all $0 \leq d \leq \Delta - 1$. Thus, the adapted tabu search $\Delta(d, x)$ -colouring heuristic terminated after the colour degree heuristic was executed and the adapted tabu search $\Delta(d, x)$ -colouring heuristic was therefore also not applied to these graphs. Finally, the adapted version of Algorithm 9 was also not applied to any complete graph, since for a complete graph, K_n , the upper bound in Step 1 of the algorithm is the exact value for $\chi_d^\Delta(K_n)$ that was also obtained by the colour degree heuristic, which implies that the suggested improvement will have no effect on the performance of Algorithm 9 applied to K_n . The results of the adapted tabu search $\Delta(d, x)$ -colouring heuristic applied to the remaining graphs from structure classes listed in §F.2 are given in Table 5.21.

Graph	Algorithm 9				Adapted version of Algorithm 9			
	Average time (s)	Average # of iterations			Average time (s)	Average # of iterations		
		χ_0^Δ	χ_1^Δ	seq.		χ_0^Δ	χ_1^Δ	seq.
C_{17}	3.55	96.2	3.4	99.6	2.90	67.8	2.8	70.6
C_{18}	3.45	70.8	2.2	73.0	2.07	43.6	0.0	43.6
C_{28}	17.40	142.0	4.4	146.4	12.89	103.8	0.0	103.8
C_{29}	17.75	152.2	5.6	157.8	19.73	155.0	5.4	160.4
C_{49}	74.56	167.2	9.0	176.2	74.32	165.6	8.2	173.8
C_{50}	80.18	172.2	8.4	180.6	65.53	140.0	0.0	140.0
W_{17}	11.79	210.8	101.6	322.4	9.16	162.2	119.2	285.8
W_{18}	12.11	167.2	129.0	306.0	7.54	80.8	122.0	207.6
W_{28}	45.75	241.0	173.6	435.8	29.75	155.4	124.8	289.2
W_{29}	51.53	240.4	171.6	438.0	27.37	159.4	151.4	320.6
W_{49}	237.61	395.8	221.4	674.8	97.49	140.0	167.4	326.4
W_{50}	247.10	404.0	226.2	690.6	115.23	181.8	163.8	364.4
$K_{4 \times 4}$	34.79	210.6	182.6	799.2	28.32	140.0	182.0	645.4
$K_{4 \times 7}$	342.99	336.2	192.8	1 900.0	236.16	140.0	234.6	1 289.8
$K_{4 \times 11}$	1 808.32	687.8	440.2	3 495.2	1 122.02	140.0	404.8	2 097.8
$K_{7 \times 4}$	428.34	446.8	294.0	2 204.8	354.47	140.0	286.4	1 849.4
$K_{11 \times 4}$	2 016.78	669.0	200.0	3 812.8	1 687.38	140.0	222.4	3 153.0

TABLE 5.21: The improvements obtained by the adapted tabu search $\Delta(d, x)$ -colouring heuristic when applied to graphs from structure classes. In all cases the tabu parameter values were taken as $\text{maxit} = 140$, $\ell = 40$ and $t = 8$. The average number of iterations performed by the original and by the adapted tabu search $\Delta(d, x)$ -colouring heuristic to obtain an upper bound on $\chi_0^\Delta(G)$ [$\chi_1^\Delta(G)$, the entire Δ -chromatic sequence, respectively] for the graph G listed in the first column, are shown in the columns labelled χ_0^Δ [χ_1^Δ , seq., respectively]

The improvement on the average number of iterations required by the tabu search $\Delta(d, x)$ -colouring heuristic when applied to wheels, which is almost half of what was required by the original algorithm for the larger wheels (W_{49} and W_{50}), is due to the fact that the number of different colours for which a valid $\Delta(d, x)$ -colouring is sought is much smaller than before. In the case of complete balanced multipartite graphs, the colour degree heuristic obtained the correct value of χ_0^Δ and then the adapted tabu search $\Delta(d, x)$ -colouring heuristic searched for a valid $\Delta(d, x)$ -colouring in one colour fewer, which of course does not exist, until the maximum number of iterations (maxit) was reached. The adapted tabu search $\Delta(d, x)$ -colouring heuristic was also much less often unable to obtain a valid $\Delta(d, x)$ -colouring than the original algorithm was (see Table 5.12).

The results of the adapted tabu search $\Delta(d, x)$ -colouring heuristic applied to the proper colouring benchmark graphs listed in Table F.8 are given in Table 5.22. As evident from Table 5.22, the number of iterations required by the tabu search $\Delta(d, x)$ -colouring heuristic is reduced by the adapted version in some cases to more than half that of the original implementation of the algorithm — see for example myciel7, huck, jean and david. In the case of myciel7, jean and david, this large reduction in the number of iterations is partly due to another advantage of the adapted tabu search $\Delta(d, x)$ -colouring heuristic. For these graphs the original implementation of the tabu search $\Delta(d, x)$ -colouring heuristic struggled to find a valid $\Delta(d, x)$ -colouring in only a few colours, resulting in additional iterations for some values of d greater than the value of d for which the colour degree could obtain a valid $\Delta(d, 2)$ -colouring. In the adapted tabu search $\Delta(d, x)$ -colouring heuristic a valid $\Delta(d, x)$ -colouring in only a few colours for these particular graphs was obtained via the colour degree heuristic at an earlier value for d than before, without a significant increase in the parameter maxit which would have resulted in more iterations for small values of d .

Graph	Algorithm 9				Adapted version of Algorithm 9			
	Average time (min)	Average # of iterations			Average time (min)	Average # of iterations		
		χ_0^Δ	χ_1^Δ	seq.		χ_0^Δ	χ_1^Δ	seq.
myciel4	0.03	148	6	154	0.02	143	3	146
myciel5	0.30	193	49	277	0.22	151	34	214
myciel6	6.22	430	314	1 046	3.49	183	211	627
queen5_5	2.00	357	254	1 009	1.47	184	208	760
queen6_6	6.29	399	251	1 423	5.36	197	264	1 218
queen7_7	15.49	560	343	1 853	11.82	283	237	1 394
myciel7	91.62	1 320	726	4 117	31.62	187	248	1 612
queen8_8	41.08	792	448	2 822	25.90	251	279	1 786
huck	25.34	583	348	1 873	12.39	166	183	952
jean	25.87	505	357	1 558	12.90	181	223	773
david	53.60	877	443	3 185	22.00	170	217	1 347

TABLE 5.22: *The improvements obtained by the adapted tabu search $\Delta(d, x)$ -colouring heuristic when applied to the proper colouring benchmark graphs. For the graphs above the horizontal line the tabu parameter values were taken as $\text{maxit} = 140$, $\ell = 40$ and $t = 8$, while the tabu parameter values $\text{maxit} = 180$, $\ell = 40$ and $t = 8$ were used for the graphs below the horizontal line. The average number of iterations performed by the original and by the adapted tabu search $\Delta(d, x)$ -colouring heuristic to obtain an upper bound on $\chi_0^\Delta(G)$ [$\chi_1^\Delta(G)$, the entire Δ -chromatic sequence, respectively] for the graph G listed in the first column, are shown in the columns labelled χ_0^Δ [χ_1^Δ , seq., respectively]*

Finally, the results of the adapted tabu search $\Delta(d, x)$ -colouring heuristic applied to the random graphs are summarised in Table 5.23.

5.2 Exact Methods

The two heuristic algorithms discussed in §5.1 only give an upper bound on the $\Delta(d)$ -chromatic number. Therefore two algorithms that determine the exact value of the $\Delta(d)$ -chromatic number were also developed. The first of these two exact methods, called the **irredundant χ_d^Δ -colouring algorithm**, as indicated by its name, uses the idea of avoiding redundancy from the paper by JR Brown [21] on one of the first exact algorithms to determine the classical chromatic number. The second algorithm attempts to find a critical $\Delta(d, x)$ -chromatic subgraph that is easier to colour than the original graph; hence the name the **critical χ_d^Δ -colouring algorithm**.

The irredundant χ_d^Δ -colouring algorithm is described in §5.2.1, while the critical χ_d^Δ -colouring algorithm is described in §5.2.2. In both cases the algorithm is given in pseudo-code and described in more detail, followed by an illustrative example of the algorithm implementation. The section is concluded in §5.2.3 with the results of the performance by both algorithms on several test graphs, as well as comparisons between the two heuristic methods and the two exact methods.

Graph	Algorithm 9				Adapted version of Algorithm 9			
	Average time (min)	Average # of iterations			Average time (min)	Average # of iterations		
		χ_0^Δ	χ_1^Δ	seq.		χ_0^Δ	χ_1^Δ	seq.
$G_{20,0.2}$	0.17	132	42	182	0.13	105	36	148
$G_{20,0.5}$	0.61	220	197	568	0.52	172	186	494
$G_{20,0.8}$	1.40	250	222	1 108	1.27	179	189	1 007
$G_{35,0.2}$	1.97	293	179	568	1.36	166	144	392
$G_{35,0.5}$	6.09	413	285	1 484	5.14	254	247	1 257
$G_{35,0.8}$	11.40	488	309	2 360	9.07	210	269	1 874
$G_{50,0.2}$	8.61	456	338	1 117	4.76	219	218	698
$G_{50,0.5}$	25.63	786	449	2 918	14.66	231	275	1 834
$G_{50,0.8}$	49.16	946	554	4 757	14.48	241	275	1 822
$G_{65,0.2}$	22.38	604	408	1 700	11.47	198	229	970
$G_{65,0.5}$	67.12	1 175	642	4 493	36.31	245	288	2 592
$G_{65,0.8}$	119.34	1 383	815	6 752	68.97	164	462	4 073
$G_{85,0.2}$	61.97	915	492	2 713	27.48	161	212	1 343
$G_{85,0.5}$	169.25	1 690	867	6 674	80.82	151	351	3 405
$G_{85,0.8}$	TO	1 941	1 084	—	167.72	140	535	5 538

TABLE 5.23: The improvements obtained by the adapted tabu search $\Delta(d, x)$ -colouring heuristic when applied to the random graphs. For the graphs above the horizontal line $(\text{maxit}, \ell, t) = (140, 40, 8)$, while for the graphs below the horizontal line $(\text{maxit}, \ell, t) = (180, 40, 8)$. The average number of iterations performed by the original and by the adapted tabu search $\Delta(d, x)$ -colouring heuristic to obtain an upper bound on $\chi_0^\Delta(G)$ [$\chi_1^\Delta(G)$, the entire Δ -chromatic sequence, respectively] for the type of graph G listed in the first column, are shown in the columns labelled χ_0^Δ [χ_1^Δ , seq., respectively], where the averages were taken over all five executions of the algorithm for all five generated graphs of that particular order and density.

5.2.1 The Irredundant χ_d^Δ -colouring Algorithm

The irredundant χ_d^Δ -colouring algorithm uses similar “screening techniques” to those introduced by Brown [21] in his algorithm, which prevents the generation of a large number of redundant non-optimal colourings as explained in §2.3. In this regard, the irredundant χ_d^Δ -colouring algorithm is an adaptation to maximum degree colourings of the original proper colouring algorithm by Brown [21] as described in §2.3. Recall from §2.3 that Breaz [13] suggested some improvements on the algorithm by Brown, which in their corrected form, due to Peemöller [92], were incorporated into Brown’s modified colouring algorithm, given in pseudo-code as Algorithm 4 in §2.3. However, these improvements could not all be incorporated into the irredundant χ_d^Δ -colouring algorithm. Firstly, recall from §2.3 that all the colours in the range colour 1, ..., colour n , where n is the order of the graph to be coloured, are available to colour the vertices when Brown’s algorithm commences, while Brown’s modified colouring algorithm (Algorithm 4) takes as input the number of colours x obtained by a heuristic method and immediately attempts to find a proper colouring utilising only $x - 1$ colours. This improvement was incorporated into the irredundant χ_d^Δ -colouring algorithm, while the second improvement on the algorithm by Brown, as suggested by Breaz, could not entirely be used as good strategy in the irredundant χ_d^Δ -colouring algorithm. In this regard, recall from §2.3 that Breaz suggested that the algorithm commences by colouring a clique and then further on during execution the algorithm only needs to backtrack until one of these initial clique vertices has to be recoloured, instead of backtracking all the way in an attempt to recolor the first vertex that was coloured, as in the case of Brown’s algorithm. The irredundant χ_d^Δ -colouring algorithm may use this improvement for $d = 0$ only, since there is more than one way to partition the clique vertices into a minimum number of colour classes when $d > 0$. Furthermore, an optimal colouring does not necessarily have to employ the minimum number of colours possible to colour the clique vertices only. Thus, when $d > 0$ the irredundant χ_d^Δ -colouring algorithm has to backtrack all the way to the first vertex that was coloured in an attempt to colour the vertices with one fewer colour than before. The irredundant χ_d^Δ -colouring algorithm, given in pseudo-code as Algorithm 10, is now discussed in more detail.

Algorithm 10 Irredundant χ_d^Δ -colouring algorithm**Input:** A graph G of order n , a value d for which $\chi_d^\Delta(G)$ must be determined, and $\omega(G)$.**Output:** $\chi_d^\Delta(G)$ as well as a χ_d^Δ -colouring of G .

```

1: Let  $x$  be the upper bound on  $\chi_d^\Delta(G)$  determined by the colour degree heuristic
2:  $s^* \leftarrow$  colouring obtained by the colour degree heuristic in Step 1
3: if  $x = \lceil \omega(G)/(d+1) \rceil$  then
4:    $\chi_d^\Delta(G) \leftarrow x, s_d \leftarrow s^*$ 
5: else [ $x > \lceil \omega(G)/(d+1) \rceil$ ]
6:   Determine colouring order  $v_1, \dots, v_n$ 
7:   Colour clique vertices and determine  $\mathcal{X}(v_i)$  for all  $i = 1, \dots, \omega$ 
8:   Label vertex  $v_1$ 
9:   STOP  $\leftarrow$  false, BACK  $\leftarrow$  false,  $k \leftarrow \omega + 1$ 
10:  while not STOP do
11:    if not BACK then
12:       $\tilde{x}_k \leftarrow$  number of colours used in the partial colouring of level  $k - 1$ 
13:       $\mathcal{X}(v_k) \leftarrow$  set of colours from  $\{1, \dots, \min(\tilde{x}_k + 1, x - 1)\}$  which may be used to colour  $v_k$ 
14:    else [BACK = true]
15:       $c \leftarrow$  colour of  $v_k$ 
16:       $\mathcal{X}(v_k) \leftarrow \mathcal{X}(v_k) - \{c\}$ ; remove label from  $v_k$  if there are any
17:    end if
18:    if  $\mathcal{X}(v_k) \neq \emptyset$  then
19:       $j \leftarrow$  minimal colour in  $\mathcal{X}(v_k)$ 
20:      Remove  $v_k$  from  $\mathcal{C}_\ell$  if  $v_k \in \mathcal{C}_\ell$  for some  $\ell \neq j$ 
21:       $\mathcal{C}_j \leftarrow \mathcal{C}_j \cup \{v_k\}$ 
22:       $k \leftarrow k + 1$ 
23:    if  $k > n$  then
24:      Let  $x^*$  be the number of colours used in this colouring  $s^* = s$ 
25:      if  $x^* = \lceil \omega(G)/(d+1) \rceil$  then
26:        STOP  $\leftarrow$  true,  $\chi_d^\Delta(G) \leftarrow x^*, s_d = s^*$ 
27:      else [ $x^* > \lceil \omega(G)/(d+1) \rceil$ ]
28:         $\mathcal{X}(v_i) \leftarrow \mathcal{X}(v_i) - \{x^*, \dots, x\}$  for all vertices  $v_i, 1 < i \leq n$ 
29:         $x \leftarrow x^*$ 
30:         $k \leftarrow$  smallest  $i$  for which  $v_i \in \mathcal{C}_x$ 
31:        Remove all labels from  $v_k, \dots, v_n$ 
32:        BACK  $\leftarrow$  true
33:      end if
34:    else [ $k \leq n$ ]
35:      BACK  $\leftarrow$  false
36:    end if
37:    else [ $\mathcal{X}(v_k) = \emptyset$ ]
38:      BACK  $\leftarrow$  true
39:    end if
40:    if BACK then
41:      Label all unlabeled vertices  $v_i$  such that (i)  $i < k$ , (ii) smallest  $i = j$  such that  $v_j \in \mathcal{C}_\ell$  and
42:       $v_j \in N(v_k)$  for all  $\ell = 1, \dots, x$ 
43:       $k \leftarrow$  maximum  $i$  such that  $v_i$  is labeled
44:      Clear  $\mathcal{X}(v_i)$  and remove  $v_i$  from its current colour class for all  $i > k$ 
45:      if  $k = 1$  or ( $k \leq \omega$  and  $d = 0$ ) then
46:        STOP  $\leftarrow$  true,  $\chi_d^\Delta(G) \leftarrow x^*, s_d = s^*$ 
47:      end if
48:    end if
49:  end while

```

The irredundant χ_d^Δ -colouring algorithm commences in Step 1 by determining an upper bound, x , via some $\Delta(d, x)$ -colouring heuristic and in particular, the colour degree heuristic (described in §5.1.1) was implemented here. In (3.2.1) the lower bound, $\lceil \omega(G)/(d+1) \rceil$, on the $\Delta(d)$ -chromatic number of a graph G was given. Therefore, if x equals this lower bound the optimum value of $\chi_d^\Delta(G)$ has already been obtained and this value x is thus assigned to $\chi_d^\Delta(G)$ in Step 4 of Algorithm 10, as well as the colouring obtained by the colour degree heuristic. Otherwise, the search for an optimal colouring begins at Step 6 by determining an order in which the vertices should be coloured. The clique vertices are ordered first in non-decreasing order of degree in an attempt to reduce the number of backtracks in most cases. The reason for this is that the last colour class after only the clique vertices have been coloured, may contain fewer vertices than the other colour classes initially containing only clique vertices, leaving this colour available to colour vertices adjacent to the clique vertices with the highest degree. The remaining vertices are ordered in such a way that vertex v_j , $j \in \{1, \dots, n\}$, is adjacent to more of the vertices in $\{v_1, \dots, v_{j-1}\}$ than to any of the vertices in $\{v_{j+1}, \dots, v_n\}$ as was done in the case of Brown's [21] original algorithm.

Let $\alpha = \lceil \omega(G)/(d+1) \rceil$. At Step 7 of Algorithm 10 the clique vertices are distributed among the colour classes $\mathcal{C}_1, \dots, \mathcal{C}_\alpha$, such that the first $\alpha - 1$ colour classes receive $d + 1$ clique vertices each and colour class \mathcal{C}_α the remainder, *i.e.* $v_{(i-1)(d+1)+1}, \dots, v_{i(d+1)} \in \mathcal{C}_i$ for all $i = 1, \dots, \alpha - 1$ and $v_j \in \mathcal{C}_\alpha$ for all $j = (\alpha - 1)(d + 1) + 1, \dots, \omega$. Colour class \mathcal{C}_α may contain fewer clique vertices than the colour classes $\mathcal{C}_1, \dots, \mathcal{C}_{\alpha-1}$ since $\omega(G)$ is not necessarily divisible by $d+1$. As before, let $\mathcal{X}(v_k)$ be the final set of colours that may be assigned to v_k after the screening process. Colouring the clique vertices first is merely an attempt to reduce the number of backtracks during the execution of Algorithm 10, but as mentioned above, for $d > 0$ backtracking all the way back to the first vertex that has to be recoloured might occur. Therefore, the set $\mathcal{X}(v_i)$ for all the clique vertices needs to be determined, except for v_1 , in which case $\mathcal{X}(v_1)$ is empty. Because of the greediness with which the clique vertices are coloured the set $\mathcal{X}(v_i)$ for all the clique vertices v_2, \dots, v_ω may contain at most two colours, namely colour i and colour $(i + 1)$ in order to avoid redundancy. If $x = \alpha + 1$, then the set $\mathcal{X}(v_j)$ may contain only one colour, namely colour α for all the clique vertices v_j for all $j = (\alpha - 1)(d + 1) + 1, \dots, \omega$.

The labelling in Steps 8 and 40 of Algorithm 10 is used to guide the backtracking during the (implicit) tree traversal. Initially, only vertex v_1 is labelled. Initialization is performed in Step 9, where among others, the vertex counter, k , is set to $w + 1$, since v_{w+1} is the first vertex that has to be coloured after the clique vertices have been coloured in Step 7. The final set of colours, $\mathcal{X}(v_k)$, that may be assigned to v_k after the screening process has been completed, is determined in Steps 12 and 13. During the screening process all the colours in the range $\{1, \dots, \min(\tilde{x}_k + 1, x - 1)\}$ are initially considered, where \tilde{x}_k is the number of colours used in the partial colouring thus far, as determined in Step 12 of the algorithm. This range may contain one more colour than was utilized before, since this might be the position in the colouring process where a new colour has to be introduced. However, since a colouring in x colours has already been obtained, no more than $x - 1$ colours may be used in an attempt at finding a valid χ_d^Δ -colouring. During the screening process the colours from this range with which vertex v_k may be coloured without violating the allowable colour class induced maximum degree, are determined. If $\mathcal{X}(v_k)$ is not empty, then v_k is coloured with the minimum colour in $\mathcal{X}(v_k)$ in Steps 19–21, and the vertex counter is incremented in Step 22 so that the next vertex to be coloured is considered. In Step 20, v_k is removed from its previous colour class if v_k has to be recoloured during backtracking. Steps 12 and 13, and Steps 18–22 are repeated as long as the vertex counter, k , is smaller than the order of the graph n , and as long as the set of possible colours for each vertex is not empty.

As soon as all the vertices have been coloured ($k > n$ in Step 23 of Algorithm 10), a colouring in fewer colours than before has been obtained. Again, this new upper bound, x^* in Step 24, is compared to $\lceil \omega(G)/(d+1) \rceil$ in Step 25. As before, if the two values are equal, then the value of $\chi_d^\Delta(G)$ has been found and the algorithm terminates at Step 26. Otherwise, this colouring is saved as a new possible optimal colouring, and all the relevant variables in Steps 28–32 are reset in preparation for the search for a colouring with even fewer colours than the one just obtained. In this regard, the excessive colours are deducted from the sets of possible colours for each vertex in Step 28, while in Step 29 x is set to the new upper bound on $\chi_d^\Delta(G)$. In Step 30 the vertex counter is set to the first vertex that was coloured with colour x^* (since this vertex is the first one that has to be recoloured in order to obtain a colouring in fewer than x^* colours) and in Step 32 the boolean variable, BACK, is set to **true** to indicate that backtracking is necessary.

Whenever the set of colours $\mathcal{X}(v_k)$ is empty when v_k is the vertex to be coloured next, backtracking is also necessary. Backtracking in this case is also initiated by the boolean variable BACK. BACK is set to true at Step 38 in this case. When backtracking is necessary, vertices are labelled in Step 41. Vertices that may have to be recoloured in order either to recolour the current vertex if a new colouring in fewer colours than before is sought, or to be able to colour the current vertex for which the set of possible colours is empty, are all labelled to indicate that the algorithm may need to backtrack to these vertices. One would like to avoid backtracking too far back; therefore the vertex counter in Step 42 is set to the vertex closest to the current vertex in the colouring order that has to be recoloured in order possibly to colour the current vertex. Once the vertex counter, k , is set to a value in Step 42, backtracking commences in Steps 15 and 16 where the current colour with which v_k is coloured is removed from the set of possible colours with which v_k may be coloured. Vertex v_k may now be recoloured following the same steps as before if $\mathcal{X}(v_k)$ is still not empty, otherwise another backtracking is performed via the boolean variable BACK in Step 38. Before the vertex v_k is recoloured at Step 21, the current label of v_k is removed in Step 16, while v_k is removed from its current colour class in Step 20. If the vertex counter, k , is set to 1 during backtracking, no more backtracking may be performed in order to possibly obtain a colouring in fewer colours and the algorithm terminates at Step 45 with the previously obtained best colouring that was saved as optimal colouring. However, when $d = 0$ the graph cannot be coloured with fewer colours than the maximum order of a clique, in which case the algorithm already terminates at Step 45 when the vertex counter, k , is set to one of the clique vertices, reducing the number of backtrackings.

The working of Algorithm 10 is illustrated in Example 5.3.

Example 5.3 *As in Examples 5.1 and 5.2, suppose a χ_1^Δ -colouring of the graph G_1 given in Figure 5.1(a) is sought. Recall from Example 5.1 that a $\Delta(1, 3)$ -colouring of G_1 was obtained via the colour degree heuristic. Therefore, x is set to 3 in Step 1 of Algorithm 10. Since $\lceil \omega(G_1)/(d+1) \rceil = 2 < 3 = x$ in Step 3, Algorithm 10 will attempt to find a χ_1^Δ -colouring in 2 colours. There are several cliques of the maximum order of 4 present in G_1 . Suppose the clique v_1, v_4, v_5, v_6 is selected to start the colouring process in Algorithm 10, then these vertices are arranged in non-decreasing order of degree in the order v_1, v_5, v_4 and v_6 . Both v_3 and v_9 are adjacent to three vertices in the clique and adjacent to only two vertices outside the clique. If v_3 is selected from v_3 and v_9 to be inserted into the colouring order next, the adjacency test indicates that v_9 is to be inserted into the colouring order after v_3 . At this point in the ordering process, v_8 is adjacent to v_4, v_6 and v_9 already ordered and adjacent to only one vertex as yet to be ordered, namely v_{10} . Continuing in this fashion, the remaining three vertices are ordered in the sequence v_2, v_7, v_{10} . Thus, the final colouring order is $v_1, v_5, v_4, v_6, v_3, v_9, v_8, v_2, v_7, v_{10}$. The vertices are renumbered in order to reflect this colouring order and the resulting graph is shown in Figure 5.9(a). The steps of Algorithm 10 applied to G_1 from Step 7 onwards are summarised in Table 5.24.*

The algorithm terminates without finding a χ_1^Δ -colouring in 2 colours. Therefore, the upper bound obtained by the colour degree heuristic is also the optimal value, i.e. $\chi_1^\Delta = 3$, and the χ_1^Δ -colouring obtained by the the colour degree heuristic with colour classes $\mathcal{C}_1 = \{v_4, v_5, v_7, v_{10}\}$, $\mathcal{C}_2 = \{v_3, v_6\}$ and $\mathcal{C}_3 = \{v_1, v_2, v_8, v_9\}$, are the final colouring output of Algorithm 10 applied to G_1 in Example 5.3. This colouring was illustrated in Figure 5.2(b). As demonstration, the χ_1^Δ -colouring in 3 colours that would have been obtained by Algorithm 10 applied to G_1 if the original upper bound x in Step 1 of Algorithm 10 was 4, is shown in Figure 5.9(b). ■

5.2.2 The Critical χ_d^Δ -colouring Algorithm

The critical χ_d^Δ -colouring algorithm is merely an adaptation to maximum degree colourings of the original Herrmann–Hertz proper colouring algorithm (Algorithm 5) by Herrmann and Hertz [63] as described in §2.3. In this regard, the critical χ_d^Δ -colouring algorithm, given in pseudo-code as Algorithm 11, attempts to find the smallest possible critical $\Delta(d, x)$ -chromatic subgraph H of the graph G for which the $\Delta(d)$ -chromatic number needs to be determined, instead of the smallest possible critical chromatic subgraph (*i.e.* in the classical sense) as in the case of the Herrmann–Hertz algorithm. Similarly to the Herrmann–Hertz algorithm, the critical χ_d^Δ -colouring algorithm also employs another exact χ_d^Δ -colouring algorithm, and in particular the irredundant χ_d^Δ -colouring algorithm, as well as a heuristic $\Delta(d, x)$ -colouring algorithm, called HEURISTIC in Algorithm 11.

Step	Action	BACK	Vertices labelled	Vertices labelled thus far
7	$v_1, v_2 \in \mathcal{C}_1, v_3, v_4 \in \mathcal{C}_2$ $\mathcal{X}(v_1) = \emptyset$ $\mathcal{X}(v_2) = \{\text{colour 1, colour 2}\}$ $\mathcal{X}(v_3) = \mathcal{X}(v_4) = \{\text{colour 2}\}$			
8			v_1	v_1
9	$k = 5$	false		v_1
13	$\mathcal{X}(v_5) = \emptyset$			v_1
18 & 38		true		v_1
41 & 42	$k = 3$		v_2, v_3	v_1, v_2, v_3
15 & 16	$c = \text{colour 2}, \mathcal{X}(v_3) = \emptyset$		remove v_3 label	v_1, v_2
18 & 38		true		v_1, v_2
41 & 42	$k = 2$		–	v_1, v_2
15 & 16	$c = \text{colour 1}, \mathcal{X}(v_2) = \{\text{colour 2}\}$		remove v_2 label	v_1
19–22	$v_2 \in \mathcal{C}_2, k = 3$			v_1
34		false		v_1
13	$\mathcal{X}(v_3) = \{\text{colour 1, colour 2}\}$			v_1
19–22	$v_3 \in \mathcal{C}_1, k = 4$			v_1
13	$\mathcal{X}(v_4) = \{\text{colour 2}\}$			v_1
19–22	$v_4 \in \mathcal{C}_2, k = 5$			v_1
13	$\mathcal{X}(v_5) = \emptyset$			v_1
18 & 38		true		v_1
41 & 42	$k = 3$		v_2, v_3	v_1, v_2, v_3
15 & 16	$c = \text{colour 1}, \mathcal{X}(v_3) = \{\text{colour 2}\}$		remove v_3 label	v_1, v_2
19–22	$v_3 \in \mathcal{C}_2, k = 4$			v_1, v_2
34		false		v_1, v_2
13	$\mathcal{X}(v_4) = \{\text{colour 1}\}$			v_1, v_2
19–22	$v_4 \in \mathcal{C}_1, k = 5$			v_1, v_2
13	$\mathcal{X}(v_5) = \emptyset$			v_1, v_2
18 & 38		true		v_1, v_2
41 & 42	$k = 4$		v_4	v_1, v_2, v_4
15 & 16	$c = \text{colour 1}, \mathcal{X}(v_4) = \emptyset$		remove v_4 label	v_1, v_2
18 & 38		true		v_1, v_2
41 & 42	$k = 3$		v_3	v_1, v_2, v_3
15 & 16	$c = \text{colour 2}, \mathcal{X}(v_3) = \emptyset$		remove v_3 label	v_1, v_2
18 & 38		true		v_1, v_2
41 & 42	$k = 2$		–	v_1, v_2
15 & 16	$c = \text{colour 2}, \mathcal{X}(v_2) = \emptyset$		remove v_2 label	v_1
18 & 38		true		v_1
41 & 42	$k = 1$		–	v_1
43	Algorithm terminates			

TABLE 5.24: The values of the parameters during the execution of the irredundant χ_d^Δ -colouring algorithm applied to the graph G_1 given in Figure 5.1(a), considered in Example 5.3.

The critical χ_d^Δ -colouring algorithm commences with an initialization in Step 1 and the determination of an upper bound x on the $\Delta(d)$ -chromatic number of the input graph G with the aid of HEURISTIC in Step 2. In Steps 4–10 of Algorithm 11 a vertex v is repeatedly removed from a subgraph H of G , originally equal to the graph G itself (Step 3), in order to obtain a new subgraph H of lesser order, as long as the upper bound on the $\Delta(d)$ -chromatic number of H obtained by HEURISTIC is equal to the original upper bound of x on $\chi_d^\Delta(G)$. During this attempt to find a critical $\Delta(d, x)$ -chromatic subgraph H of G , the list P is used to keep track of which vertices have already been tested to determine whether or not the vertex may be removed, and which ones have not yet been tested — initially P contains all the vertices of G (Step 3) and as a vertex is tested, whether or not it is removed from subgraph H , the vertex is removed from P (Step 5). The order in which the vertices should be tested is determined as the algorithm proceeds by selecting from amongst the untested vertices in P , one with the smallest degree

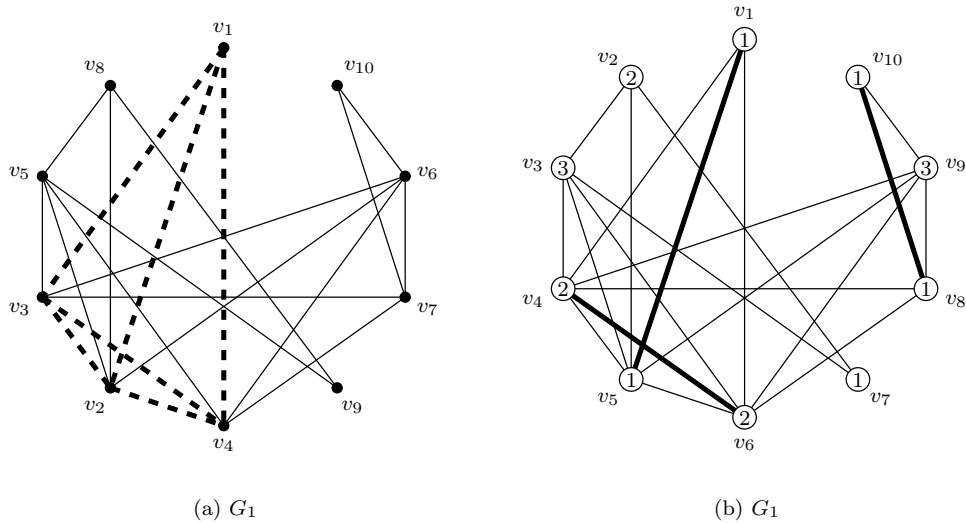


FIGURE 5.9: (a) A renumbering of the vertices of the graph G_1 in Figure 5.1(a) according to the colouring order determined in the irredundant χ_d^Δ -colouring algorithm. The dashed bold edges represent the maximum clique that will be coloured first during execution of the irredundant χ_d^Δ -colouring algorithm. (b) A final χ_1^Δ -colouring of G_1 in Figure 5.1(a) in 3 colours as obtained via the irredundant χ_d^Δ -colouring algorithm in Example 5.3 if the original upper bound x in Step 1 of Algorithm 10 was 4. The bold edges indicate that the colour class induced maximum degrees for all colour classes are at most 1.

in the current subgraph H . After all the vertices in $V(G)$ have been tested and removed (if applicable), the exact $\Delta(d)$ -chromatic number, k' , of the final subgraph H is determined in Step 11 of Algorithm 11. If $k' = x$ or $H = G$ the algorithm terminates at Step 13; otherwise the parameter k' in the rest of the algorithm serves as a lower bound on the $\Delta(d)$ -chromatic number of G and is adapted as the algorithm proceeds from Step 15 onwards.

During the while-loop spanning Steps 15–42 of Algorithm 11 vertices are added back into the subgraph H until either $\chi_d^\Delta(H) = x$ (Step 38) and thus $\chi_d^\Delta(G) = x$, or $H = G$ (Step 35). At the beginning of each iteration of the while-loop spanning Steps 15 and 42 the list T contains all the vertices in $V(G)$ that are not contained in the vertex set of the current subgraph H , as assigned in Step 17 of the algorithm. For each iteration of the for-loop spanning Steps 18 and 27, a vertex v in T is selected and removed from T (Step 19). Using HEURISTIC, an upper bound m on $\chi_d^\Delta(H + v)$ is determined in Step 20. If this upper bound m is strictly greater than the $\Delta(d)$ -chromatic number of H , the exact value m' of the $\Delta(d)$ -chromatic number of $H + v$ is determined in Step 22 of the algorithm with the aid of the irredundant χ_d^Δ -colouring algorithm. If $\chi_d^\Delta(H + v)$ is one more than $\chi_d^\Delta(H)$ (only one vertex was added, thus only one more colour could be required), this vertex v is stored in the list L in Step 24. Thus, the list L contains all vertices that increase the $\Delta(d)$ -chromatic number if they are added to subgraph H . At the completion of the for-loop, if L is not empty, a vertex v such that $\deg_{H+v}(v) \geq \deg_{H+u}(u)$ for any vertex u in L is selected from L (Step 29) and added to subgraph H , and the $\Delta(d)$ -chromatic number of the current subgraph H (parameter k') is updated in Step 30 of the algorithm. If L is empty, a vertex v in G that is not in H , again with largest number of vertices adjacent to it in H , is selected and added to H in Step 33 of the algorithm. Vertices are added to the current subgraph H in this way as long as either all the vertices of G that were not originally in H are added to H in which case the algorithm terminates at Step 36, or the $\Delta(d)$ -chromatic number of the current subgraph H is equal to the upper bound x on the $\Delta(d)$ -chromatic number of G , in which case the algorithm terminates at Step 39.

The working of Algorithm 11 is illustrated in Example 5.4.

Example 5.4 As in Examples 5.1–5.3, suppose a χ_1^Δ -colouring of the graph G_1 given in Figure 5.1(a) is sought. Also, suppose that the colour degree heuristic is used as HEURISTIC in Algorithm 11. Recall from Example 5.1 that a $\Delta(1, 3)$ -colouring of G_1 was obtained via the colour degree heuristic. Therefore, x is

Algorithm 11 Critical χ_d^Δ -colouring algorithm**Input:** A graph G of order n and a value d for which the $\Delta(d)$ -chromatic number must be determined.**Output:** $\chi_d^\Delta(G)$

```

1: STOP  $\leftarrow$  false
2: Determine an upper bound  $x$  on  $\chi_d^\Delta(G)$  by means of HEURISTIC
3:  $H \leftarrow G$ ;  $P \leftarrow V(G)$ 
4: while  $P \neq \emptyset$  do
5:   Choose  $v \in P$  such that  $\deg_H(v) \leq \deg_H(u)$  for all  $u \in P$ ;  $P \leftarrow P \setminus \{v\}$ 
6:   Determine an upper bound  $k$  on  $\chi_d^\Delta(H - v)$  by means of HEURISTIC
7:   if  $k = x$  then
8:      $H \leftarrow H - v$ 
9:   end if
10: end while
11: Determine  $k' = \chi_d^\Delta(H)$  by means of IRREDUNDANT  $\chi_d^\Delta$ -COLOURING ALGORITHM
12: if  $k' = x$  or  $H = G$  then
13:   STOP  $\leftarrow$  true,  $\chi_d^\Delta \leftarrow k'$ 
14: end if
15: while not STOP do
16:    $L \leftarrow \emptyset$ 
17:    $T \leftarrow V(G) \setminus V(H)$ 
18:   for all  $i = 1, \dots, |T|$  do
19:     Choose  $v \in T$ ;  $T \leftarrow T \setminus \{v\}$ 
20:     Determine an upper bound  $m$  on  $\chi_d^\Delta(H + v)$  by means of HEURISTIC
21:     if  $m > k'$  then
22:       Determine  $m' = \chi_d^\Delta(H + v)$  by means of IRREDUNDANT  $\chi_d^\Delta$ -COLOURING ALGORITHM
23:       if  $m' = k' + 1$  then
24:          $L \leftarrow L \cup \{v\}$ 
25:       end if
26:     end if
27:   end for
28:   if  $L \neq \emptyset$  then
29:     Choose  $v \in L$  such that  $|N_{H+v}(v)| \geq |N_{H+u}(u)|$  for all  $u \in L$ 
30:      $H \leftarrow H + v$ ;  $k' \leftarrow k' + 1$ 
31:   else [ $L = \emptyset$ ]
32:     Choose  $v \in V(G) \setminus V(H)$  such that  $|N_{H+v}(v)| \geq |N_{H+u}(u)|$  for all  $u \in V(G) \setminus V(H)$ 
33:      $H \leftarrow H + v$ 
34:   end if
35:   if  $H = G$  then
36:     STOP  $\leftarrow$  true,  $\chi_d^\Delta \leftarrow k'$ 
37:   else [ $H \subset G$ ]
38:     if  $k' = x$  then
39:       STOP  $\leftarrow$  true,  $\chi_d^\Delta \leftarrow k'$ 
40:     end if
41:   end if
42: end while

```

set to 3 in Step 2 of Algorithm 11. It is clear from Figure 5.1(a) that $\deg_{G_1}(v_7) = \deg_{G_1}(v_{10}) = \delta(G_1) = 2$ and any one of the vertices v_7 or v_{10} may be selected first to determine whether this vertex may be removed from G_1 in search of a critical $\Delta(1, x)$ -chromatic subgraph H of G_1 . If v_{10} is selected, the upper bound k on $\chi_1^\Delta(G_1 - \{v_{10}\})$ obtained via the colour degree heuristic in Step 6 of Algorithm 11 is 3 and the resulting $\Delta(1, 3)$ -colouring of $G_1 - \{v_{10}\}$ is given in Figure 5.10(a). Therefore, v_{10} is removed from G_1 in Step 8 to obtain the current subgraph $H = G_1 - \{v_{10}\}$. The actions taken and values of the parameters during the other iterations of the while-loop spanning Steps 4–10, are summarised in Table 5.25.

Thus, the critical $\Delta(1, x)$ -chromatic subgraph H of G_1 obtained at the end of the while-loop spanning

Step	Degrees in H /Action	v	k	H	P
5	$\deg_H(v_4) = \deg_H(v_5) = \deg_H(v_6) = 6,$ $\deg_H(v_3) = 5, \deg_H(v_9) = 4,$ $\deg_H(v_1) = \deg_H(v_2) = \deg_H(v_8) = 3,$ $\deg_H(v_7) = \delta(H) = 2$	v_7		$(G_1 - \{v_{10}\})$	$\{v_1, v_2, v_3, v_4,$ $v_5, v_6, v_8, v_9\}$
6–8	(see Figure 5.10(b))		3	$G_1 - \{v_7, v_{10}\}$	
5	$\deg_H(v_4) = \deg_H(v_5) = \deg_H(v_6) = 6,$ $\deg_H(v_3) = \deg_H(v_9) = 4, \deg_H(v_1) =$ $\deg_H(v_8) = 3, \deg_H(v_2) = \delta(H) = 2$	v_2			$\{v_1, v_3, v_4, v_5,$ $v_6, v_8, v_9\}$
6–8	(see Figure 5.10(c))		3	$G_1 - \{v_2, v_7, v_{10}\}$	
5	$\deg_H(v_4) = \deg_H(v_6) = 6, \deg_H(v_5) =$ $5, \deg_H(v_9) = 4, \deg_H(v_1) = \deg_H(v_3)$ $= \deg_H(v_8) = \delta(H) = 3$	v_8			$\{v_1, v_3, v_4, v_5,$ $v_6, v_9\}$
6–8	(see Figure 5.10(d))		3	$G_1 - \{v_2, v_7, v_8, v_{10}\}$	
5	$\deg_H(v_4) = \deg_H(v_5) = \deg_H(v_6) = 5,$ $\deg_H(v_1) = \deg_H(v_3) =$ $\deg_H(v_9) = \delta(H) = 3$	v_9			$\{v_1, v_3, v_4, v_5,$ $v_6\}$
6–8	(see Figure 5.10(e))		3	$G_1 - \{v_2, v_7, v_8, v_9, v_{10}\}$	
5	$\deg_H(v_4) = \deg_H(v_5) = \deg_H(v_6) = 4,$ $\deg_H(v_1) = \deg_H(v_3) = \delta(H) = 3$	v_3			$\{v_1, v_4, v_5, v_6\}$
6–8	(see Figure 5.10(f))		2		
5&6	$\deg_H(v_4) = \deg_H(v_5) = \deg_H(v_6) = 4,$ $\deg_H(v_1) = \delta(H) = 3$	v_1	2		$\{v_4, v_5, v_6\}$
5&6	$\deg_H(v_4) = \deg_H(v_5) = \deg_H(v_6) = 4$	v_6	2		$\{v_4, v_5\}$
5&6	$\deg_H(v_4) = \deg_H(v_5) = 4$	v_5	2		$\{v_4\}$
5&6	$\deg_H(v_4) = 4$	v_4	2		\emptyset
4	While-loop terminates				

TABLE 5.25: The values of the parameters during execution of the while-loop spanning Steps 4–10 of the critical χ_d^Δ -colouring algorithm applied to the graph G_1 in Figure 5.1(a), considered in Example 5.4.

Steps 4–10 of Algorithm 11 is the graph $H = G_1 - \{v_2, v_7, v_8, v_9, v_{10}\}$ shown in Figure 5.10(e). The exact value of $\chi_1^\Delta(H)$ determined by the irredundant χ_d^Δ -colouring algorithm in Step 11 of Algorithm 11 is $k' = 3$, so that the algorithm terminates at Step 13 with $\chi_1^\Delta(G_1) = 3$. ■

5.2.3 Results Obtained by the χ_d^Δ -colouring Algorithms

As before, the irredundant χ_d^Δ -colouring algorithm was adapted to determine the Δ -chromatic sequences of the input graphs and not only the $\Delta(d)$ -chromatic number for a given value of d as in Algorithm 10. Likewise, once a value of 2 was obtained for x_k in the Δ -chromatic sequence (x_i) , all values of x_d for $d = k + 1, k + 2, \dots, \Delta - 1$ were set equal to 2 in order to speed up the execution time. This adapted version of the irredundant χ_d^Δ -colouring algorithm, coded in MATHEMATICA, is included on the CD accompanying this dissertation (see Appendix G).

The irredundant χ_d^Δ -colouring algorithm was first executed on the small graphs listed in §F.1 and the results are summarised in Table 5.26. For a specific value of d , the irredundant χ_d^Δ -colouring algorithm will only perform backtracks in search of a valid χ_d^Δ -colouring of a graph G in fewer colours than that obtained by the colour degree heuristic if the upper bound obtained by the colour degree heuristic is strictly larger than the lower bound $\lceil \omega/(d+1) \rceil$ on $\chi_d^\Delta(G)$. For this reason, only graphs for which backtracks during the execution of the irredundant χ_d^Δ -colouring algorithm were necessary, are listed in Table 5.26. Note that in Algorithm 10 the clique number of the particular graph is given as input, but in the MATHEMATICA implementation of Algorithm 10 the clique number is determined at the beginning of the program. However, since determining the clique number is a known **NP-complete** problem (see Table 2.2), all time measurements commence *after* the clique number was determined.

In the first four columns of Table 5.26 the graph name, the order, the size and the clique number of the

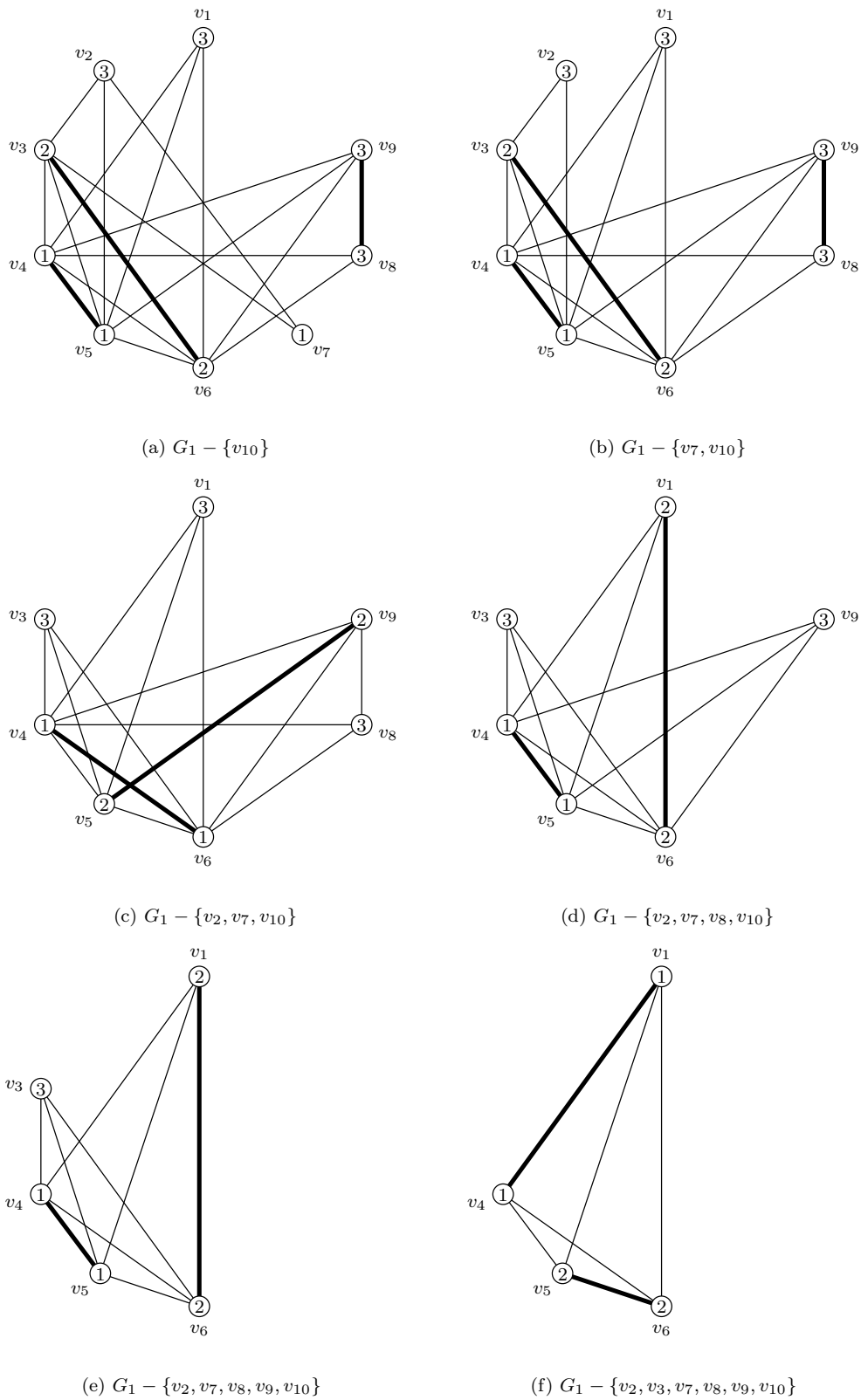


FIGURE 5.10: $\Delta(1, x)$ -colourings of the subgraphs of G_1 obtained during execution of Algorithm 11.

G	$p(G)$	$q(G)$	$\omega(G)$	First 3 values of			# of backtracks			running time (in sec.)
				exact	u. bound	l. bound	$d = 0$	$d = 1$	(x_i)	
				seq. (x_i)	by CDH	$\lceil \omega/(d+1) \rceil$				
$G38$	5	5	2	3 2 1	3 2 1	2 1 1	3	2	5	0.016
$G48$	5	7	3	3 2 2	3 3 2	3 2 1	0	3	3	–
$G50$	5	8	3	3 3 2	3 3 2	3 2 1	0	8	10	0.016
$G51$	5	9	4	4 3 2	4 3 2	4 2 2	0	7	7	0.016
$C5$	8	12	3	3 2 2	3 3 2	3 2 1	0	1	1	0.016
$C7$	8	12	2	3 2 2	3 2 2	2 1 1	4	2	6	0.016
$Q2$	6	12	3	3 3 2	3 3 2	3 2 1	0	8	10	0.016
$Q3$	7	12	3	4 3 2	4 3 2	3 2 1	4	12	18	0.031
$Q5$	8	16	3	3 3 2	3 3 2	3 2 1	0	11	13	0.031
$Q6$	8	16	3	4 3 2	5 3 2	3 2 1	7	9	19	0.031
$Q7$	8	16	3	3 2 2	3 3 3	3 2 1	0	3	3	0.031
$Q8$	8	16	3	4 2 2	5 2 2	3 2 1	7	0	7	0.016
$F2$	8	20	4	4 2 2	4 4 2	4 2 2	0	3	3	0.031
$F3$	8	20	3	4 3 2	4 4 2	3 2 1	4	7	13	0.031
$F4$	8	20	4	4 3 2	4 4 2	4 2 2	0	4	4	0.031
$X2$	8	24	4	4 4 2	4 4 2	4 2 2	0	48	48	0.047
$Ct6$	10	15	2	3 2 2	3 3 2	2 1 1	5	8	13	0.047
$Ct8$	10	15	2	3 2 2	3 2 2	2 1 1	3	1	4	0.031
$Qt10$	10	20	3	4 3 2	4 3 2	3 2 1	7	7	16	0.047
$Qt12$	10	20	2	3 3 2	3 3 2	2 1 1	4	13	19	0.047
Average							2.4	7.9	11.1	0.0288

TABLE 5.26: The results obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the small graphs (order at most 10) as listed in §F.1.

selected graphs are listed. In the subsequent three columns the first three values of the Δ -chromatic sequence, the upper bounds on the values in the Δ -chromatic sequence obtained by the colour degree heuristic, and the lower bounds on the values in the Δ -chromatic sequence obtained by the formula $\lceil \omega/(d+1) \rceil$, are listed. The number of backtracks performed during execution of the irredundant χ_d^Δ -colouring algorithm when $d = 0$, when $d = 1$ and to determine the values of the entire Δ -chromatic sequence, are shown in columns 8–10. It is easy to see that if the upper bound on $\chi_d^\Delta(G)$ obtained by the colour degree heuristic and the lower bound on $\chi_d^\Delta(G)$ obtained by the formula $\lceil \omega(G)/(d+1) \rceil$ for a graph G are equal, then during the execution of the irredundant χ_d^Δ -colouring algorithm, no backtracks were necessary for that particular value of d , since the exact value has already been established by the two bounds. On the other hand, if the upper bound on $\chi_d^\Delta(G)$ obtained by the colour degree heuristic was strictly larger than the lower bound on $\chi_d^\Delta(G)$ obtained by the formula $\lceil \omega/(d+1) \rceil$ for a graph G , then either a colouring in fewer colours than the upper bound had to be found, or it had to be shown that no colouring in fewer colours than the upper bound exists. Consequently, in these cases backtracks were performed during the execution of the irredundant χ_d^Δ -colouring algorithm.

The running times required to determine the Δ -chromatic sequence of each graph listed in the first column are shown in the last column of Table 5.26. Note that, as before, the computation time for the graph $G48$ was too short to be captured by MATHEMATICA and was omitted from any calculations. As evident from the running times in Table 5.26 the irredundant χ_d^Δ -colouring algorithm, in general, performs well on graphs of order at most 10 and the exact values in the Δ -chromatic sequence of a particular graph were obtained. For this reason, the irredundant χ_d^Δ -colouring algorithm is preferable to the two heuristics described in §5.1 for graphs of order at most 10.

In the case of trees, the lower bound $\omega(T)$ on the $\Delta(0)$ -chromatic number of a tree T is 2 for all trees. Furthermore, since the colour degree heuristic determines the upper bound on the $\Delta(0)$ -chromatic number of a tree T as 2 for all trees, the execution of the irredundant χ_d^Δ -colouring algorithm when applied to a tree T will result in no backtracking when $\chi_0^\Delta(T)$ is determined. Since $\chi_0^\Delta(T) = 2$ for any tree T , $\chi_d^\Delta(T)$ for all $0 < d < \Delta$ is also set equal to 2, resulting in no backtracking at all when the Δ -chromatic sequence of a tree is determined. Also, the running time of the irredundant χ_d^Δ -

colouring algorithm applied to a tree will approximately be that of the running time when the colour degree heuristic is applied to that particular tree. Thus, the irredundant χ_d^Δ -colouring algorithm was not executed on the trees in Table F.4.

For graphs from the remaining structure classes listed in §F.2, the irredundant χ_d^Δ -colouring algorithm only needs to be executed on a selection of these graphs as well. Since the clique number of a cycle is 2 and, as in the case of trees, the colour degree heuristic determines the correct value of $\chi_0^\Delta(C_n)$ for an even cycle of order n . However, if n is odd, the lower bound on $\chi_0^\Delta(C_n)$ [$\chi_1^\Delta(C_n)$] determined by the formula $\lceil \omega(C_n)/(d+1) \rceil$ is still 2 [1 respectively], but $\chi_0^\Delta(C_n) = 3$ and $\chi_1^\Delta(C_n) = 2$. In this case some backtracking will take place during the execution of the irredundant χ_d^Δ -colouring algorithm in order to show that no valid χ_0^Δ -colouring [χ_1^Δ -colouring] in fewer than 3 [2 respectively] colours is possible. The results of the irredundant χ_d^Δ -colouring algorithm when applied to the three odd cycles in Table F.5 are shown in the first three lines of Table 5.27.

G	$p(G)$	$q(G)$	$\omega(G)$	First 4 values of		# of backtracks			running time (in sec.)
				exact seq. (x_i)	l. bound $\lceil \omega/(d+1) \rceil$	$d=0$	$d=1$	(x_i)	
C_{17}	17	17	2	3 2 1 1	2 1 1 1	15	2	17	0.094
C_{29}	29	29	2	3 2 1 1	2 1 1 1	27	2	29	0.344
C_{49}	49	49	2	3 2 1 1	2 1 1 1	47	2	49	1.391
W_{17}	17	32	3	3 3 2 2	3 2 1 1	0	12	14	0.141
W_{18}	18	34	3	4 3 2 2	3 2 1 1	15	12	29	0.172
W_{28}	28	54	3	4 3 2 2	3 2 1 1	25	12	39	0.500
W_{29}	29	56	3	3 3 2 2	3 2 1 1	0	12	14	0.500
W_{49}	49	96	3	3 3 2 2	3 2 1 1	0	12	14	2.001
W_{50}	50	98	3	4 3 2 2	3 2 1 1	47	12	61	2.298
$K_{4 \times 4}$	16	96	4	4 4 4 3	4 2 2 1	0	529	1772	1.840
Average						17.6	60.7	203.8	0.928

TABLE 5.27: The results obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the graphs from structure classes as listed in §F.2.

The results obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the wheels listed in Table F.5, are shown in lines 4–9 of Table 5.27. It follows from Proposition 4.4 that the lower bound $\lceil \omega(K_n)/(d+1) \rceil$ is exactly the $\Delta(d)$ -chromatic number of the complete graph K_n for all values of d . These values are also the values determined by the colour degree heuristic, so that execution of the irredundant χ_d^Δ -colouring algorithm in the case of complete graphs is needless as well. Similar arguments as before render the execution of the irredundant χ_d^Δ -colouring algorithm on the bipartite graphs listed in Table F.6 redundant.

Recall from Table 5.3 that the colour degree heuristic performed poorly on the complete balanced multipartite graphs listed in Table F.7. Consequently, the irredundant χ_d^Δ -colouring algorithm when applied to a complete balanced multipartite graph has to search for a valid $\Delta(d, x)$ -colouring in fewer colours a number of times for certain values of d . However, as the difference between the upper bound on $\chi_d^\Delta(K_{k \times n})$ as determined by the colour degree heuristic and $\chi_d^\Delta(K_{k \times n})$ itself, is relatively large initially, a valid $\Delta(d, x)$ -colouring of $K_{k \times n}$ may be obtained easily at first, but as this difference decreases, the search for a valid $\Delta(d, x)$ -colouring in fewer colours requires more and more backtracking during the execution of the irredundant χ_d^Δ -colouring algorithm. Although the number of times the irredundant χ_d^Δ -colouring algorithm, when applied to a particular complete balanced multipartite graph, has to search for a valid $\Delta(d, x)$ -colouring in fewer colours is the largest for $d=1$, the running time required to determine $\chi_d^\Delta(K_{k \times n})$ increases as d increases. Therefore, the number of times a valid $\Delta(d, x)$ -colouring in fewer colours is sought is not necessarily the largest contribution towards the running time when the irredundant χ_d^Δ -colouring algorithm is applied to a complete balanced multipartite graph.

In order to demonstrate why, when the irredundant χ_d^Δ -colouring algorithm is applied to a complete balanced multipartite graph, the running time increases as d increases, recall from §2.3 that during the course of Brown's modified colouring algorithm a tree is implicitly constructed and traversed. Similarly, during the course of the irredundant χ_d^Δ -colouring algorithm a tree is also implicitly constructed and

traversed. During the execution of the irredundant χ_d^Δ -colouring algorithm this implicitly constructed tree grows rapidly as d increases. For example, consider the graph G_1 in Figure 5.1(a) used to illustrate the working of the irredundant χ_d^Δ -colouring algorithm in Example 5.3. The colouring order was shown in Figure 5.9(a), where the initial clique was v_1, v_5, v_4 and v_6 . Suppose a valid $\Delta(d, x)$ -colouring in four colours is sought. For $d = 0$ all the clique vertices must have different colours and the colouring of the clique vertices in the tree implicitly constructed during the execution of the irredundant χ_d^Δ -colouring algorithm consists of only one node, namely $v_1 \in \mathcal{C}_1, v_5 \in \mathcal{C}_2, v_4 \in \mathcal{C}_3$ and $v_6 \in \mathcal{C}_4$. However, when $d \geq 1$ only v_1 is coloured uniquely. The partial trees implicitly constructed during the execution of the irredundant χ_d^Δ -colouring algorithm for colouring the clique vertices when $d = 1$ and when $d = 2$ are shown in Figures 5.11(a) and (b), respectively. In both cases vertex v_1 is coloured uniquely and forms the root of the tree (level 0), while vertex v_5 may be coloured with two possible colours. Thus, both trees have two nodes at level 1. However, the tree implicitly constructed during the execution of the irredundant χ_d^Δ -colouring algorithm when $d = 1$ has four nodes at level 2 and ten nodes at level 3, while the tree in the case of $d = 2$ has five nodes at level 2 and 14 nodes at level 3. In the case of $d = 0$, the first vertex to be coloured after the clique vertices, v_3 , may only be coloured with colour 1 and thus the tree constructed for $d = 0$ has one node on level 1. The possibilities to colour v_3 result in 25 nodes on level 4 when $d = 1$ and a staggering 45 nodes on level 4 when $d = 2$.

The program implementing the irredundant χ_d^Δ -colouring algorithm was adapted to abort execution when the Δ -chromatic sequence of a particular graph could not be obtained within a time limit of three hours. When the irredundant χ_d^Δ -colouring algorithm was applied to the complete balanced multipartite graphs listed in Table F.7, only the Δ -chromatic sequence of $K_{4 \times 4}$ could be determined within the three hour time limit, and its results are listed in the last line of Table 5.27. In Table 5.28 the results of the irredundant χ_d^Δ -colouring algorithm applied to the remaining complete balanced multipartite graphs before the time limit was reached, are shown.

G	$p(G)$	$q(G)$	$\omega(G)$	$d = 0$			$d = 1$			$d = 2$		
				χ_0^Δ	cdh	time (s)	χ_1^Δ	cdh	time (s)	χ_2^Δ	cdh	time (s)
$K_{4 \times 7}$	28	294	4	4	4	1.468	4	14	1.51	4	8	176.19
$K_{4 \times 11}$	44	726	4	4	4	1.002	4	22	26.85	4	12	631.10
$K_{7 \times 4}$	28	336	7	7	7	0.008	7	13	2354.17	?	7	TO
$K_{7 \times 11}$	77	2541	7	?	7	TO		34			19	
$K_{11 \times 4}$	44	880	11	?	11	TO		21			11	

TABLE 5.28: The results obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the complete balanced multipartite graphs listed in Table F.7. A “TO” in a time column indicates that the time limit of three hours was reached at that particular value of d .

The first four columns in Table 5.28 are the same as before. The last nine columns of Table 5.28 are divided into three groups of three columns each, where each group of three columns represents the results for $d = 0, d = 1$ and $d = 2$, respectively. The first column in each group contains $\chi_d^\Delta(K_{k \times n})$, while the second column in each group shows the upper bound on $\chi_d^\Delta(K_{k \times n})$ when the colour degree heuristic was applied to $K_{k \times n}$. Finally, the last column in each group contains the running time required to determine $\chi_d^\Delta(K_{k \times n})$. A “TO” in the final column of each group indicates that $\chi_d^\Delta(K_{k \times n})$ could not be obtained via the irredundant χ_d^Δ -colouring algorithm within the remaining time of the time limit of three hours set to determine the Δ -chromatic sequence.

The results obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the proper colouring benchmark instances listed in Table F.8 are summarised in Tables 5.29 and 5.30 for those graphs for which the Δ -chromatic sequence could be obtained within the time limit, and for those graphs for which the Δ -chromatic sequence could *not* be obtained within the time limit, respectively.

The sequence of upper bounds on the entries in the Δ -chromatic sequence of the graph myciel4 determined by the colour degree heuristic was 4 2 2 2 2 1, which are also the exact values. However, the lower bound on $\chi_0^\Delta(\text{myciel4})$ determined by $\omega(\text{myciel4})$ is 2, so that a total of 48 backtracks were performed during the execution of the irredundant χ_d^Δ -colouring algorithm when $d = 0$ to show that no valid $\Delta(0, 3)$ -colouring of myciel4 exists.

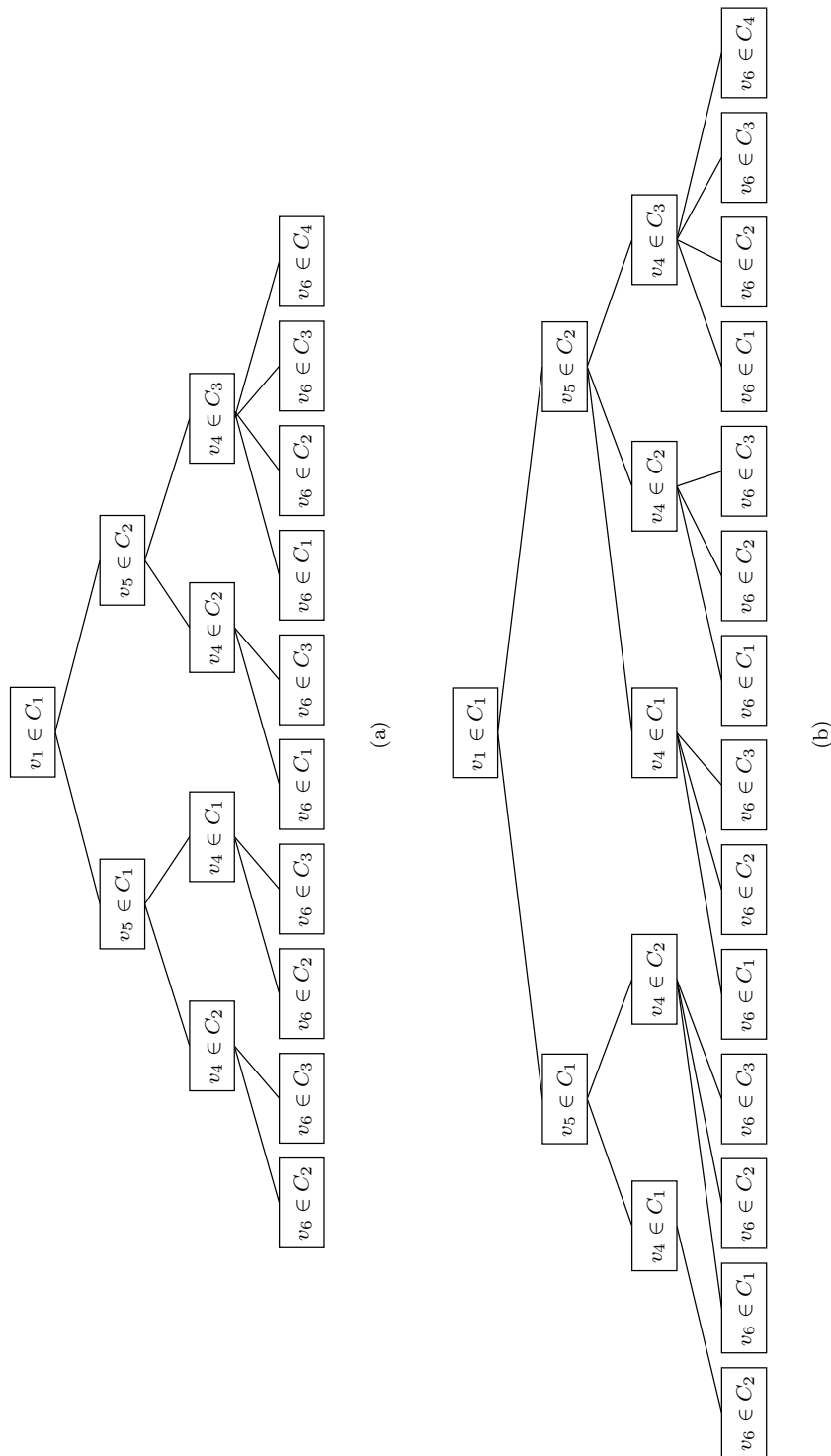


FIGURE 5.11: Partial trees that are implicitly constructed for the graph G_1 in Figure 5.1(a) during the course of the irredundant χ_d^Δ -colouring algorithm when the colouring order in Figure 5.9(a) is used. These partial trees are implicitly constructed during the colouring of the initial clique vertices of G_1 in the irredundant χ_d^Δ -colouring algorithm when (a) $d = 1$ and (b) $d = 2$.

G	p	q	ω	Δ -chromatic sequence	# of backtracks			running time
					$d = 0$	$d = 1$	seq.	
myciel4	11	20	2	4 2 2 2 2 1	48	1	49	0.047
myciel5	23	71	2	5 3 3 2 2 2 2 2 2 2 1	125 549	10	125 975	228.999
queen55	25	160	5	5 5 4 3 3 3 2 2 2 2 2 2 2 2 1	0	16 526	58 068	93.484

TABLE 5.29: The Δ -chromatic sequences obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the proper colouring benchmark instances listed in Table F.8 for which the sequence could be obtained within the time limit.

G	$p(G)$	$q(G)$	$\omega(G)$	$d = 0$			$d = 1$			$d = 2$		
				χ_0^Δ	cdh	time (s)	χ_1^Δ	cdh	time (s)	χ_2^Δ	cdh	time (s)
myciel6	47	236	2	6	6	TO		4			3	
myciel7	95	755	2	7	7	TO		6			6	
queen66	36	290	6	7	8	84.859	6	8	2 459.984		7	TO
queen77	49	476	7	7	10	227.608		8	TO		7	
queen88	64	728	8	9	11	TO		9			8	
huck	74	301	11	11	11	16.719	6	6	0.001		5	TO
jean	80	254	10	10	10	18.062	6	7	1 078.665	4	5	10.422
david	87	406	11	11	11	33.734		7	TO		6	

TABLE 5.30: The results obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the proper colouring benchmark instances listed in Table F.8 for which the sequence could not be obtained within the time limit of three hours. A “TO” in a time column indicates that the time limit of three hours was reached at that particular value of d .

The same phenomenon occurred if $d = 1$, but from then onwards $\chi_d^\Delta(\text{myciel4})$ was set to 2 for all $2 < d < \Delta$, resulting in no more backtracks during the execution of the irredundant χ_d^Δ -colouring algorithm. Similarly, since the upper bounds on the entries in the Δ -chromatic sequence of the graph myciel5 determined by the colour degree heuristic were 5 3 3 3 2 2 2 2 2 2 2 1, a total of 125 549 [10 respectively] backtracks were performed during the execution of the irredundant χ_d^Δ -colouring algorithm to show that no valid $\Delta(0, 4)$ -colouring of myciel5 [no valid $\Delta(1, 2)$ -colouring of myciel5, respectively] exists. The remaining number of backtracks (416) performed during the execution of the irredundant χ_d^Δ -colouring algorithm when applied to myciel5, were performed to show that no valid $\Delta(2, 2)$ -colouring of myciel5 exists and to find a χ_3^Δ -colouring in 2 colours. In the case of the graph queen55 the upper bounds on the entries in the Δ -chromatic sequence determined by the colour degree heuristic were 5 5 4 3 3 3 2 2 2 2 2 2 2 2 1. Here the upper bound on $\chi_0^\Delta(\text{queen55})$ determined by the colour degree heuristic and the lower bound $\omega(\text{queen55})$ are both equal to 5, resulting in no backtracking during the execution of the irredundant χ_d^Δ -colouring algorithm when $d = 0$. However, a number of backtracks were necessary when the irredundant χ_d^Δ -colouring algorithm was applied to queen55 for other values of d for the same reasons as those discussed above.

Finally, the irredundant χ_d^Δ -colouring algorithm was also applied to the random graphs of the form $G_{n,p}$ (where n and p are the order and the density of the graph respectively) as described in §F.4. The Δ -chromatic sequence could only be obtained within a three hour time limit for random graphs of order 20 and random graphs of order 35 with density 0.2. These Δ -chromatic sequences are shown in Table 5.31.

A summary of the results of the irredundant χ_d^Δ -colouring algorithm when applied to the remaining random graphs of order 35, *i.e.* those graphs for which the Δ -chromatic sequence could not be obtained within a three hour time limit, are given in Table 5.32. No results within a three hour time limit could be obtained when the irredundant χ_d^Δ -colouring algorithm was applied to the random graphs of orders 50, 65 and 85 and are therefore excluded from this section.

The critical χ_d^Δ -colouring algorithm was also adapted to determine the Δ -chromatic sequences of the input graphs and not only their $\Delta(d)$ -chromatic numbers for a given value of d as in Algorithm 11. Again, once a value of 2 was obtained for x_k in the Δ -chromatic sequence (x_i) , all values of x_d for $d = k+1, k+2, \dots, \Delta-1$ were set equal to 2 in order to speed up the execution time. This adapted version of the critical χ_d^Δ -colouring algorithm, coded in MATHEMATICA, is included on the CD accompanying this dissertation (see Appendix G).

G	ω	Δ -chromatic sequence	# of backtracks	running time (s)
$G_{20,0.2}$ #1	3	3 3 2 2 2 2 2 1	25	0.219
$G_{20,0.2}$ #2	3	4 3 2 2 2 2 2 1	97	0.359
$G_{20,0.2}$ #3	3	3 3 2 2 2 2 2 1	41	0.234
$G_{20,0.2}$ #4	3	3 3 2 2 2 2 2 1	41	0.219
$G_{20,0.2}$ #5	3	4 3 2 2 2 2 2 1	121	0.328
		Average	65	0.272
$G_{20,0.5}$ #1	5	5 4 3 3 3 2 2 2 2 2 2 1	2918	4.422
$G_{20,0.5}$ #2	5	6 4 3 3 3 2 2 2 2 2 1	1327	2.656
$G_{20,0.5}$ #3	5	6 4 3 3 3 2 2 2 2 2 2 2 1	12773	16.750
$G_{20,0.5}$ #4	5	6 4 3 3 3 2 2 2 2 2 2 2 1	2837	4.547
$G_{20,0.5}$ #5	6	6 4 3 3 3 2 2 2 2 2 2 2 1	2152	2.953
		Average	4401	6.266
$G_{20,0.8}$ #1	9	9 7 5 4 4 3 3 3 2 2 2 2 2 2 2 2 1	1038475	1327.679
$G_{20,0.8}$ #2	9	9 6 5 4 3 3 3 2 2 2 2 2 2 2 2 1	88641	101.734
$G_{20,0.8}$ #3	10	10 6 5 4 3 3 3 3 2 2 2 2 2 2 2 2 1	73184	81.171
$G_{20,0.8}$ #4	9	9 6 5 4 3 3 3 3 2 2 2 2 2 2 2 2 1	90685	96.281
$G_{20,0.8}$ #5	9	9 6 5 4 3 3 3 3 2 2 2 2 2 2 2 2 1	75157	88.296
		Average	273228	339.032
$G_{35,0.2}$ #1	4	4 3 3 3 2 2 2 2 2 2 2 1	1511	5.328
$G_{35,0.2}$ #2	4	4 3 3 2 2 2 2 2 2 2 2 2 1	15492	28.812
$G_{35,0.2}$ #3	4	5 4 3 3 2 2 2 2 2 2 2 2 1	45055	100.249
$G_{35,0.2}$ #4	4	4 4 3 3 2 2 2 2 2 2 2 2 1	161195	401.794
$G_{35,0.2}$ #5	4	4 3 3 2 2 2 2 2 2 2 2 1	3743	13.484
		Average	45399	109.934

TABLE 5.31: The Δ -chromatic sequences obtained within the time limit by the irredundant χ_d^Δ -colouring algorithm when applied to some random graphs created according to the parameters listed in Table F.9.

G	$\omega(G)$	$d = 0$		$d = 1$		$d = 2$		$d = 3$			
		χ_0^Δ	back	time	χ_1^Δ	back	time	χ_2^Δ	time	χ_3^Δ	time
$G_{35,0.5}$ #1	7	8	40599	102	6	4594007	751	5	2292		TO
$G_{35,0.5}$ #2	6	8	643752	1239	6	4983840	2278	4	3727	4	550
$G_{35,0.5}$ #3	7	8	22182	40	6	6255728	2010	5	7775		TO
$G_{35,0.5}$ #4	7	8	10956	23	6	5118958	856	5	3686	4	4000
$G_{35,0.5}$ #5	7	7	6140	15	6	5512467	1068	5	7125	4	669
$G_{35,0.8}$ #1	12	14	115376	233			TO				
$G_{35,0.8}$ #2	13	13	0	5			TO				
$G_{35,0.8}$ #3	11	12	12469	32			TO				
$G_{35,0.8}$ #4	13	14	14051	35			TO				
$G_{35,0.8}$ #5	12	14	213984	426			TO				

TABLE 5.32: The results obtained by the irredundant χ_d^Δ -colouring algorithm when applied to the random graphs described in §F.4, for which the sequence could not be obtained within a three hour time limit. The two “back” columns represent the number of backtracks performed during the execution of the irredundant χ_d^Δ -colouring algorithm when applied to the particular random graph of order 35 when $d = 0$ and $d = 1$, respectively. All times in the table are given in seconds. A “TO” in a time column indicates that the three hour time limit was exceeded for that particular value of d .

During application of the critical χ_d^Δ -colouring algorithm to the small graphs listed in §F.1 the colour degree heuristic was chosen as HEURISTIC in Algorithm 11. An extract of the results thus obtained are summarised in Table 5.33, where the graphs are listed in the first column, followed by the order of each graph and upper bounds on the entries in the $\Delta(d)$ -chromatic sequence [exact values, respectively] obtained via the colour degree heuristic [the irredundant χ_d^Δ -colouring algorithm, respectively] in column 3 [column 4, respectively] of Table 5.33. The total number of times the colour degree heuris-

tic (CDH) and the irredundant χ_d^Δ -colouring algorithm (ICA) were called during the execution of the critical χ_d^Δ -colouring algorithm are given in columns 5 and 6 respectively. The execution times of the critical χ_d^Δ -colouring algorithm (for determining the entire Δ -chromatic sequence of the graphs listed in the first column) are listed in column 7. Recall from §5.2.1 that the irredundant χ_d^Δ -colouring algorithm requires that the clique number of the particular graph for which the Δ -chromatic sequence is sought be determined. Since determining the clique number of a graph is a known **NP-complete** problem (see Table 2.2), the execution time for determining the clique number of each subgraph before the call to the irredundant χ_d^Δ -colouring algorithm was captured. The total times dedicated to determining the clique number of a subgraph during the execution of the critical χ_d^Δ -colouring algorithm, are listed in column 8 of Table 5.33. This enables one to see what fraction of the execution time was used to determine clique numbers. For comparison purposes, the execution times of the colour degree heuristic (CDH), the tabu search $\Delta(d, x)$ -colouring heuristic (Tabu) and the irredundant χ_d^Δ -colouring algorithm (ICA) when applied to graphs listed in the first column, are listed in the last three columns of Table 5.33.

G	$p(G)$	u. bounds by CDH	exact seq. (x_i)	Calls to		Execution time (s)				
				CDH	ICA	Algorithm 11		Alg. 8 (CDH)	Alg. 9 (Tabu)	Alg. 10 (ICA)
						(x_i)	clique			
G_{29}	5	2 2 2 2 1	2 2 2 2 1	1	0	–	–	–	0.016	–
G_{30}	5	2 2 2 1	2 2 2 1	1	0	–	–	–	–	–
G_{45}	5	4 2 2 2 1	4 2 2 2 1	7	1	0.031	–	–	0.031	–
G_{46}	5	3 2 2 2 1	3 2 2 2 1	7	1	0.016	–	–	0.016	–
G_{48}	5	3 3 2 1	3 2 2 1	12	2	0.047	–	0.016	0.016	0.016
G_{51}	5	4 3 2 2 1	4 3 2 2 1	13	2	0.078	–	0.016	0.094	0.016
G_{52}	5	5 3 2 2 1	5 3 2 2 1	13	2	0.047	–	0.016	0.109	0.016
C_2	6	3 3 2 1	3 2 2 1	14	2	0.094	0.016	0.016	0.063	0.016
C_4	8	3 3 2 1	3 2 2 1	19	3	0.203	0.047	0.016	0.094	0.031
C_5	8	3 3 2 1	3 2 2 1	24	8	0.374	0.078	0.016	0.094	0.016
Q_3	7	4 3 2 2 1	4 3 2 2 1	20	5	0.234	0.016	0.016	0.188	0.031
Q_6	8	5 3 2 2 1	4 3 2 2 1	25	6	0.359	0.062	0.016	0.234	0.031
Q_7	8	3 3 3 2 1	3 2 2 2 1	24	8	0.390	0.094	0.016	0.313	0.031
Q_8	8	5 2 2 2 1	4 2 2 2 1	10	1	0.109	0.016	0.016	0.203	0.016

TABLE 5.33: The results obtained by the critical χ_d^Δ -colouring algorithm when applied to the small graphs (order at most 10) listed in §F.1. A “–” in a time column indicates that the execution time was too short to be captured by MATHEMATICA.

The software implementation of the critical χ_d^Δ -colouring algorithm was further adapted to terminate whether a value of 2 for the upper bound x on the $\Delta(d)$ -chromatic number of a graph G was obtained via the colour degree heuristic at Step 2 of Algorithm 11, where $\chi_{d-1}^\Delta(G)$ was still greater than 2. See, for example, the graph G_{29} (and G_{30}) in Table 5.33, where the upper bound x on $\chi_0^\Delta(G_{29})$ obtained via the colour degree heuristic at Step 2 of Algorithm 11 was 2, so that Steps 3–42 of the algorithm were not executed for $d = 0$. Since $\chi_0^\Delta(G_{29}) = 2$, $\chi_d^\Delta(G_{29})$ was set equal to 2 for all $1 \leq d \leq \Delta(G_{29}) - 1$ as before, resulting in only one call to the colour degree heuristic and none to the irredundant χ_d^Δ -colouring algorithm during the execution of the critical χ_d^Δ -colouring algorithm.

In some cases where the colour degree heuristic obtained the correct Δ -chromatic sequence for the particular graph G , the colour degree heuristic was called once in Step 2 of Algorithm 11 plus $p(G)$ times in Step 6 for each d such that $\chi_d^\Delta(G) > 2$, since each vertex was tested to determine whether the vertex should be removed in search of a critical $\Delta(d, x)$ -chromatic subgraph H of G . One further call to the colour degree heuristic in Step 2 when an upper bound of 2 for the particular d was obtained, resulted in the total number of calls to the colour degree heuristic. In these cases the only calls to the irredundant χ_d^Δ -colouring algorithm occurred in Step 11 of the critical χ_d^Δ -colouring algorithm and in particular once for each value of d such that $\chi_d^\Delta(G) > 2$. Consider, for example, the graph G_{45} (and G_{46}) in Table 5.33. The colour degree heuristic was called a total of seven times, once in Step 2 and five times in Step 6 ($p(G_{45}) = 5$) for $d = 0$, and only once in Step 2 for $d = 1$, since $\chi_1^\Delta(G_{45}) = 2$ and the algorithm terminated after Step 2 when $d = 1$ as described above. The irredundant χ_d^Δ -colouring algorithm was called once only, namely at Step 11 of Algorithm 11 for $d = 0$. As another example, consider the critical

χ_d^Δ -colouring algorithm applied to the graph $G51$ ($G52$) of order 5. The colour degree heuristic was called a total of thirteen times, once in Step 2 and five times in Step 6 for $d = 0$ first and then for $d = 1$ as well as a final call in Step 2 when $\chi_2^\Delta(G51) = 2$, while the irredundant χ_d^Δ -colouring algorithm was called twice, first in Step 11 for $d = 0$ and then for $d = 1$. In all these cases Steps 15–42 of Algorithm 11 were not executed at all.

The number of times the colour degree heuristic and the irredundant χ_d^Δ -colouring algorithm were called during the execution of the critical χ_d^Δ -colouring algorithm when the colour degree heuristic obtained the correct Δ -chromatic sequence for the particular graph G , do not in general occur as described above. For example, the colour degree heuristic obtained the correct Δ -chromatic sequence for the graph $Q3$, which is, in fact, the circulant $C_7\langle 1, 2 \rangle$. However, during execution of the critical χ_d^Δ -colouring algorithm when applied to $Q3$, five calls to the irredundant χ_d^Δ -colouring algorithm were made although only $\chi_0^\Delta(Q3)$ and $\chi_1^\Delta(Q3)$ are greater than 2. In this case the subgraph H of $Q3$ obtained by removing two adjacent even (or odd) numbered vertices still resulted in an upper bound of 3 on $\chi_1^\Delta(H)$ obtained by the colour degree heuristic, while the irredundant χ_d^Δ -colouring algorithm obtained the exact value of 2. Therefore Steps 15–42 of Algorithm 11 had to be executed, resulting in more than one call to the irredundant χ_d^Δ -colouring algorithm for $d = 1$.

On the other hand, there were also cases where the colour degree heuristic did not obtain the correct Δ -chromatic sequence for a graph, but Steps 15–42 of Algorithm 11 were still not executed. For example, the upper bound on $\chi_1^\Delta(G48)$ obtained via the colour degree heuristic was 3, while the exact value is 2. However, in this case for each vertex v of $G48$ an upper bound of 2 on $\chi_1^\Delta(G48 - v)$ was obtained for each subgraph $G48 - v$, so that the irredundant χ_d^Δ -colouring algorithm in Step 11 of Algorithm 11 was applied to $G48$ itself and the algorithm terminated for $d = 1$ at Step 13. Other than the graphs mentioned above, a few more graphs of order at most 10 are listed in Table 5.33.

It should be clear from columns 7 and 8 of Table 5.33 that the time required to determine the clique numbers of the relevant subgraphs does not significantly increase the overall execution time of the critical χ_d^Δ -colouring algorithm when applied to that particular graph. Furthermore, studying the last five columns of Table 5.33 the irredundant χ_d^Δ -colouring algorithm seems to be the best choice amongst the four algorithms for the graphs of order at most 10 — the execution time is still relatively low in comparison with the fastest algorithm (the colour degree heuristic), whilst the exact values of the entries in the Δ -chromatic sequence are obtained.

Finally, although the original Herrmann–Hertz proper colouring algorithm as described in §2.3, was considered a better algorithm than Brown’s modified proper colouring algorithm regarding execution time, this is not entirely the case for the critical χ_d^Δ -colouring algorithm over the irredundant χ_d^Δ -colouring algorithm. In order to motivate this observation the critical χ_d^Δ -colouring algorithm was applied to the random graphs of orders 20 and 35 and compared to the results obtained by the irredundant χ_d^Δ -colouring algorithm. First, the colour degree heuristic was chosen as HEURISTIC; then the tabu search $\Delta(d, x)$ -colouring heuristic was chosen as HEURISTIC, and the results for both choices are given in Table 5.35. When the colour degree heuristic was implemented as HEURISTIC in Algorithm 11, the upper bound on $\chi_d^\Delta(G)$ was often strictly greater than the exact value, resulting in a larger number of executions of the irredundant χ_d^Δ -colouring algorithm during Steps 15–42 of the critical χ_d^Δ -colouring algorithm. Therefore, although execution of the colour degree heuristic is very fast, the large number of calls to the irredundant χ_d^Δ -colouring algorithm cause the critical χ_d^Δ -colouring algorithm in conjunction with the colour degree heuristic, to reach the time limit of three hours for smaller order graphs or smaller density graphs than in the case of the irredundant χ_d^Δ -colouring algorithm. On the other hand, when the tabu search $\Delta(d, x)$ -colouring heuristic was implemented as HEURISTIC in Algorithm 11, the upper bounds on $\chi_d^\Delta(G)$ was more often equal to the exact value than before, resulting in a smaller number of executions of the irredundant χ_d^Δ -colouring algorithm during Steps 15–42 of the critical χ_d^Δ -colouring algorithm than in the previous case. However, the execution time of the tabu search $\Delta(d, x)$ -colouring heuristic is much larger than that of the colour degree heuristic, so that the large number of calls to HEURISTIC during the execution of the critical χ_d^Δ -colouring algorithm in conjunction with the tabu search $\Delta(d, x)$ -colouring heuristic, again caused the algorithm to reach the time limit of three hours for smaller order graphs or smaller density graphs than in the case of the irredundant χ_d^Δ -colouring algorithm. It is also interesting to note how often the time required to determine $\chi_1^\Delta(G)$ via the critical χ_d^Δ -colouring algorithm with either the colour degree heuristic or the tabu search $\Delta(d, x)$ -colouring

heuristic is larger than the time required to determine $\chi_0^\Delta(G)$. This may be one reason why the critical χ_d^Δ -colouring algorithm was outperformed by the irredundant χ_d^Δ -colouring algorithm when applied to the test graphs considered in this dissertation in contrast to the preference of the Herrmann–Hertz proper colouring algorithm over Brown’s modified proper colouring algorithm.

5.3 Chapter Summary

The aim of this chapter was to develop four $\Delta(d, x)$ -colouring algorithms. The first algorithm, namely the colour degree heuristic presented in §5.1.1, is a greedy heuristic with a very short execution time in comparison with the other three $\Delta(d, x)$ -colouring algorithms — for all 196 test graphs the execution time of the colour degree heuristic was less than 3 minutes. In the case of certain graph classes such as trees, cycles, wheels, complete graphs and complete bipartite graphs, the colour degree heuristic obtained the correct Δ -chromatic sequence for all examples, but performed poorly in the case of complete balanced k -partite graphs, for $k > 2$.

The second algorithm, namely the tabu search $\Delta(d, x)$ -colouring heuristic described in §5.1.2, is a local search technique that was outperformed by the colour degree heuristic when applied to certain graph classes, but when the algorithms were applied to random graphs, the tabu search $\Delta(d, x)$ -colouring heuristic outperformed the colour degree heuristic by far, especially for larger order graphs. However, for small graphs (roughly of order less than 35) the execution time of the tabu search $\Delta(d, x)$ -colouring heuristic was even longer than the two exact algorithms. It is only for larger order graphs that the tabu search $\Delta(d, x)$ -colouring heuristic becomes beneficial, when the algorithm can obtain better upper bounds on the Δ -chromatic sequence of some graph than the colour degree heuristic, while the exact algorithms reach a time limit of three hours in these cases.

In §5.2.1 the first of the two exact algorithms, namely the irredundant χ_d^Δ -colouring algorithm, was presented, while the other one, namely the critical χ_d^Δ -colouring algorithm, was described in §5.2.2. The execution times of the critical χ_d^Δ -colouring algorithm in conjunction with either the colour degree heuristic or the tabu search $\Delta(d, x)$ -colouring heuristic, applied to the test graphs listed in Appendix F, were much longer than those of the irredundant χ_d^Δ -colouring algorithm. For small graphs (roughly of order less than 20) the execution time of the irredundant χ_d^Δ -colouring algorithm was even faster than that of the tabu search $\Delta(d, x)$ -colouring heuristic. Therefore, for small graphs the irredundant χ_d^Δ -colouring algorithm is preferred over the other three algorithms. However, it is uncertain whether there is some point (*i.e.* a specific graph order) when the execution time of the critical χ_d^Δ -colouring algorithm is shorter than that of the irredundant χ_d^Δ -colouring algorithm provided that the time limit of three hours is increased.

For general graphs, *i.e.* excluding classes where the specific structure of the graph may benefit some of the algorithms, the ranking of the four algorithms according to their performance in terms of the correctness of the sequence obtained by the algorithm as well as the execution time of the algorithm, is given in Table 5.34.

In terms of correctness	In terms of execution times	
	For Order < 35	For Order \geq 35
1. Algorithm 10	1. Algorithm 8	1. Algorithm 8
2. Algorithm 11	2. Algorithm 10	2. Adapted Algorithm 9
3. Algorithm 9	3. Algorithm 11 in conjunction with Algorithm 8	3. Algorithm 10
4. Algorithm 8	4. Algorithm 9	4. Algorithm 11

TABLE 5.34: *The ranking of the four $\Delta(d, x)$ -colouring algorithms according to their performances on the test graphs used in this dissertation, where Algorithm 8 is the colour degree heuristic, Algorithm 9 is the tabu search $\Delta(d, x)$ -colouring heuristic, Algorithm 10 is the irredundant χ_d^Δ -colouring algorithm and Algorithm 11 is the critical χ_d^Δ -colouring algorithm.*

G	HEURISTIC = CDH						HEURISTIC = Tabu						Execution time (min)	
	Calls to		Execution time (min)				Calls to		Execution time (min)				CDH	ICA
	CDH	ICA	d=0	d=1	seq.	clique	Tabu	ICA	d=0	d=1	seq.	clique	CDH	ICA
$G_{20,0.2}$ #1	93	51	0.02	0.19	0.21	0.04	53	7	1.05	0.43	1.49	0.00	0.003	0.089
$G_{20,0.2}$ #2	186	105	0.16	0.14	0.50	0.11	111	39	5.75	0.77	6.75	0.04	0.004	0.191
$G_{20,0.2}$ #3	140	85	0.22	0.14	0.37	0.08	98	53	7.94	0.50	8.44	0.05	0.003	0.166
$G_{20,0.2}$ #4	50	15	0.02	0.06	0.09	0.01	37	2	1.12	0.48	1.60	0.00	0.003	0.040
$G_{20,0.2}$ #5	55	14	0.02	0.07	0.09	0.01	55	8	1.98	0.92	2.91	0.01	0.003	0.170
$G_{20,0.5}$ #1	252	113	0.17	0.40	1.14	0.29	165	41	10.44	6.59	20.07	0.12	0.008	0.543
$G_{20,0.5}$ #2	202	82	0.08	0.30	0.82	0.25	138	21	6.30	6.10	13.76	0.06	0.007	0.482
$G_{20,0.5}$ #3	156	44	0.03	0.25	0.79	0.14	121	13	3.41	5.26	12.86	0.05	0.007	0.279
$G_{20,0.5}$ #4	227	96	0.13	0.34	0.94	0.26	154	25	8.63	6.98	17.17	0.07	0.007	0.499
$G_{20,0.5}$ #5	197	68	0.03	0.32	0.77	0.27	130	16	3.14	8.33	13.13	0.07	0.007	0.487
$G_{20,0.8}$ #1	201	34	0.03	14.05	31.51	0.79	195	25	4.94	42.29	73.94	0.60	0.012	1.505
$G_{20,0.8}$ #2	221	54	0.03	2.49	7.39	1.02	195	28	3.89	12.12	29.88	0.56	0.011	0.964
$G_{20,0.8}$ #3	224	58	0.03	2.24	9.10	1.60	193	27	4.38	10.38	29.70	0.78	0.012	1.146
$G_{20,0.8}$ #4	196	33	0.03	3.14	6.80	0.46	186	23	4.75	11.08	31.61	0.33	0.012	1.162
$G_{20,0.8}$ #5	222	55	0.13	2.24	7.32	1.05	192	27	7.62	14.33	35.54	0.67	0.012	1.187
$G_{35,0.2}$ #1	700	445	2.55	7.69	15.26	2.67	307	158	92.11	14.33	35.54	1.05	0.021	1.602
$G_{35,0.2}$ #2	675	441	0.17	18.37	29.44	2.48	387	138	16.94	94.53	124.62	0.85	0.021	1.259
$G_{35,0.2}$ #3	591	354	0.18	4.83	11.46	1.86	337	132	18.52	82.60	109.57	0.73	0.025	1.601
$G_{35,0.2}$ #4	599	306	0.17	2.82	7.96	1.24	368	113	16.67	56.29	83.66	0.45	0.030	1.410
$G_{35,0.2}$ #5	909	642	3.36	9.39	21.05	2.95	472	227	74.28	94.24	179.63	1.22	0.025	1.380

TABLE 5.35: The results obtained by the critical χ_d^A -colouring algorithm when applied to the random graphs of orders 20 and 35 as listed in §F.4.

Chapter 6

Maximum Degree Chromatic Sequences

“No pessimist ever discovered the secret of the stars or sailed an uncharted land, or opened a new doorway for the human spirit.”

Helen Keller (1880–1968)

In this chapter the objective is to investigate whether the necessary conditions in Theorem 3.13 for a sequence of positive integers to be the Δ -chromatic sequence of some graph G , are also sufficient conditions. The chapter opens, in §6.1, with the introduction of the notion of **basic sequences** that satisfy the conditions in Theorem 3.13. In §6.2 basic sequences that can also be classified as Δ -chromatic sequences are identified. The chapter continues in §6.3 with the smallest basic sequence that can as yet not be classified as Δ -chromatic, as well as characteristics of this sequence. Different structure graph classes studied as a possible means to classify the particular basic sequence in §6.3 (as well as some other basic sequences) as Δ -chromatic are investigated in §6.4.

6.1 Basic Sequences

In order to verify whether the conditions in Theorem 3.13 are also sufficient for maximum degree chromatic sequences, graph constructions for each sequence satisfying the conditions in Theorem 3.13 must be found. Let \mathcal{A} be the set of all sequences (x_i) satisfying the conditions in Theorem 3.13. The number of entries in a sequence up to and including the first unit entry, is called the **length** of the sequence. As mentioned in Chapter 3, Table C.1 in Appendix C contains a subset of \mathcal{A} and in particular all the sequences with length at most thirteen and with the minimum number of twos in the sequence. A program was written in MATLAB to determine all sequences in \mathcal{A} up to a length of no more than some specified number, with the minimum number of twos in the sequence, thereby verifying the correctness and completeness of Table C.1. This program was also used to extend Table C.1 by including sequences of length larger than thirteen. The set \mathcal{A} is very large and therefore it is desirable to determine a subset of essential sequences of \mathcal{A} for the purposes of study. Studying the similarities in some sequences in Table C.1, as well as how they may be created from one another, led to the set defined below.

Definition 6.1 Let \mathcal{B} be the set of **basic sequences** (x_i) described by the following rules:

- (a) The sequence (x_i) with $x_i = 1$ for all $i \in \mathbb{N}_0$ is a basic sequence.
- (b) For $n \in \mathbb{N} \setminus \{1\}$ and $r \in \mathbb{N}$, the sequence (x_i) defined by $x_i = \lceil n/k \rceil$ for all $i \in \{r(k-1), \dots, rk-1\}$ and for $k \in \mathbb{N}$ is a basic sequence. ■

An illustration of how to determine a basic sequence from Definition 6.1 is given in the following example.

Example 6.1 Let $n = 4$ and $r = 3$. Then the basic sequence (x_i) in Definition 6.1(b) is defined by $x_i = \lceil 4/k \rceil$ for all $i \in \{3k - 3, 3k - 2, 3k - 1\}$, where $k = 1, 2, 3, \dots$. The different values of k then give the following terms in the sequence:

$$\begin{aligned} k = 1: & \quad x_0 = x_1 = x_2 = \lceil 4/1 \rceil = 4, \\ k = 2: & \quad x_3 = x_4 = x_5 = \lceil 4/2 \rceil = 2, \\ k = 3: & \quad x_6 = x_7 = x_8 = \lceil 4/3 \rceil = 2, \\ k = 4: & \quad x_9 = x_{10} = x_{11} = \lceil 4/4 \rceil = 1, \\ k \geq 5: & \quad x_{3(k-1)} = x_{3k-2} = x_{3k-1} = \lceil 4/k \rceil = 1. \end{aligned}$$

Thus, the basic sequence for $n = 4$ and $r = 3$ is the sequence 4 4 4 2 2 2 2 2 2 1 1 1 ... of length 10. ■

A MATLAB program was also written to generate the set of basic sequences starting with $n \in \{1, 2, \dots, m\}$, where m is some specified positive integer, and up to a specified length. An extract of these sequences, containing all the sequences of length at most 17, starting with $n \in \{1, 2, \dots, 10\}$ is shown in Table 6.1. As in the case of Table C.1, only the first 1 in these sequences is listed in each case.

Other sequences may now be determined from the basic sequences by means of the operation described in the following definition.

Definition 6.2 Let \mathcal{B}^* be the set of all sequences (x_i) obtained by the following rules:

- (a) All basic sequences are in \mathcal{B}^* .
- (b) If (x_i) and (y_i) are in \mathcal{B}^* , then the sequence (z_i) with $z_i = \max\{x_i, y_i\}$ is again in \mathcal{B}^* . ■

In the next proposition it is shown that the sequences in \mathcal{B}^* are precisely all the sequences satisfying the conditions in Theorem 3.13.

Proposition 6.1 $\mathcal{B}^* = \mathcal{A}$.

Proof: To prove that $\mathcal{B}^* \subseteq \mathcal{A}$ it is shown that all basic sequences are in the set \mathcal{A} and that, if the sequences (x_i) and (y_i) are basic sequences, then the sequence (z_i) with $z_i = \max\{x_i, y_i\}$ satisfies the conditions in Theorem 3.13 and is therefore also in \mathcal{A} .

The sequence (x_i) with $x_i = 1$ for all $i \in \mathbb{N}_0$ is clearly in \mathcal{A} , since $1 \leq \lceil (j+1)/(i+1) \rceil$ if $0 \leq i < j$. Let (x_i) be a basic sequence as defined in Definition 6.1(b) with fixed n and r . If $k = n$, then $x_{r(n-1)} = \lceil n/n \rceil = 1$. Therefore the sequence satisfies condition (1) of Theorem 3.13. Now suppose $0 \leq i < j$. Then $i \in \{r(k-1), \dots, rk-1\}$ for some $k \in \mathbb{N}$ and $j \in \{r(k'-1), \dots, rk'-1\}$ for some $k' \in \mathbb{N}$ satisfying $k' \geq k$ and $x_i = \lceil n/k \rceil$ and $x_j = \lceil n/k' \rceil$. Clearly, $x_j \leq x_i$. Since $1 \leq rk - r + 1 \leq i + 1 \leq rk$, it follows that $1/(i+1) \geq 1/rk$. Also, $j+1 \geq rk' - r + 1$. Thus, $(j+1)/(i+1) \geq k'/k - (r-1)/(rk)$. Since $0 \leq (r-1)/(rk) < 1$ and $1/(rk) \leq 1/k$ it follows that $\lceil (j+1)/(i+1) \rceil \geq \lceil k'/k - (r-1)/(rk) \rceil = \lceil k'/k \rceil$. Hence $x_i = \lceil n/k \rceil \leq \lceil n/k' \rceil \lceil k'/k \rceil \leq x_j \lceil (j+1)/(i+1) \rceil$ and therefore the sequence also satisfies condition (2) of Theorem 3.13. Thus, (x_i) is in \mathcal{A} .

Next, suppose the two sequences $(x_i) = x_0 x_1 x_2 \dots x_{n-1} x_n \dots$ with $x_{n-1} > 1$ and $x_n = 1$, and $(y_i) = y_0 y_1 y_2 \dots y_{m-1} y_m \dots$ with $y_{m-1} > 1$ and $y_m = 1$ are basic sequences. It has already been shown that (x_i) and (y_i) are in \mathcal{A} , thus $x_j \leq x_i \leq x_j \lceil (j+1)/(i+1) \rceil$ and $y_j \leq y_i \leq y_j \lceil (j+1)/(i+1) \rceil$ if $0 \leq i < j$. Suppose, without loss of generality, that $m \geq n$ and let $(z_i) = z_0 z_1 z_2 \dots z_{m-1} z_m, \dots$ be the sequence with $z_i = \max\{x_i, y_i\}$ for all $i \in \mathbb{N}_0$. Then $z_m = \max\{x_m, y_m\} = \max\{1, 1\} = 1$. Hence the sequence (z_i) satisfies condition (1) of Theorem 3.13. Furthermore, $z_j = \max\{x_j, y_j\} \leq \max\{x_i, y_i\} = z_i$ if $0 \leq i < j$. Also, $z_i = \max\{x_i, y_i\} \leq \max\{x_j \lceil (j+1)/(i+1) \rceil, y_j \lceil (j+1)/(i+1) \rceil\} = \max\{x_j, y_j\} \times \lceil (j+1)/(i+1) \rceil = z_j \lceil (j+1)/(i+1) \rceil$ if $0 \leq i < j$. The sequence (z_i) therefore also satisfies condition (2) of Theorem 3.13 and is thus in \mathcal{A} . Therefore $\mathcal{B}^* \subseteq \mathcal{A}$.

To show that $\mathcal{A} \subseteq \mathcal{B}^*$, consider a sequence (x_i) in \mathcal{A} . The proof that (x_i) is in \mathcal{B}^* is achieved by means of a construction whereby the sequence (x_i) is decomposed into basic sequences $(x_i^{(1)}), (x_i^{(2)}), (x_i^{(3)}), \dots, (x_i^{(k)})$ in the sense that $x_i = \max\{x_i^{(1)}, x_i^{(2)}, x_i^{(3)}, \dots, x_i^{(k)}\}$.

Let $(z_i^{(0)}) = 1 1 1 \dots$. Let $k = 1$ and do the following. If $(x_i) \neq (z_i^{(k-1)})$, let j_k be the first position for which $z_{j_k}^{(k-1)} \neq x_{j_k}$. Let $n_k = x_{j_k}$ and let r_k be the first position for which $x_{r_k} < x_{j_k}$. Let $(x_i^{(k)})$

n	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
1	1																
2	2	1															
	2	2	1														
	2	2	2	1													
	2	2	2	2	1												
	2	2	2	2	2	1											
	2	2	2	2	2	2	1										
	2	2	2	2	2	2	2	1									
	2	2	2	2	2	2	2	2	1								
	2	2	2	2	2	2	2	2	2	1							
	2	2	2	2	2	2	2	2	2	2	1						
	2	2	2	2	2	2	2	2	2	2	2	1					
	2	2	2	2	2	2	2	2	2	2	2	2	1				
	2	2	2	2	2	2	2	2	2	2	2	2	2	1			
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1		
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
3	3	2	1														
	3	3	2	2	1												
	3	3	3	2	2	2	1										
	3	3	3	3	2	2	2	1									
	3	3	3	3	3	2	2	2	1								
	3	3	3	3	3	3	2	2	2	1							
	3	3	3	3	3	3	3	2	2	2	1						
	3	3	3	3	3	3	3	3	2	2	2	1					
	3	3	3	3	3	3	3	3	2	2	2	2	1				
	3	3	3	3	3	3	3	3	2	2	2	2	2	1			
	3	3	3	3	3	3	3	3	2	2	2	2	2	2	1		
4	4	2	2	1													
	4	4	2	2	2	2	1										
	4	4	4	2	2	2	2	1									
	4	4	4	4	2	2	2	2	1								
	4	4	4	4	4	2	2	2	2	1							
	4	4	4	4	4	2	2	2	2	2	1						
	4	4	4	4	4	2	2	2	2	2	2	1					
5	5	3	2	2	1												
	5	5	3	3	2	2	2	1									
	5	5	5	3	3	3	2	2	2	1							
	5	5	5	5	3	3	3	3	2	2	2	1					
	5	5	5	5	3	3	3	3	2	2	2	2	1				
	5	5	5	5	3	3	3	3	2	2	2	2	2	1			
6	6	3	2	2	2	1											
	6	6	3	3	2	2	2	1									
	6	6	6	3	3	3	2	2	2	2	1						
	6	6	6	3	3	3	2	2	2	2	2	1					
	6	6	6	3	3	3	2	2	2	2	2	2	1				
7	7	4	3	2	2	2	1										
	7	7	4	4	3	3	2	2	2	1							
	7	7	4	4	3	3	2	2	2	2	1						
	7	7	4	4	3	3	2	2	2	2	2	1					
8	8	4	3	2	2	2	2	1									
	8	8	4	4	3	3	2	2	2	2	2	1					
	8	8	4	4	3	3	2	2	2	2	2	2	1				
	8	8	4	4	3	3	2	2	2	2	2	2	2	1			
9	9	5	3	3	2	2	2	2	1								
	9	9	5	5	3	3	3	3	2	2	2	2	1				
	9	9	5	5	3	3	3	3	2	2	2	2	2	1			
	9	9	5	5	3	3	3	3	2	2	2	2	2	2	1		
10	10	5	4	3	2	2	2	2	1								

TABLE 6.1: The basic sequences in \mathcal{B} starting with $n \in \{1, 2, \dots, 10\}$ and with $x_i = 1$ for all $i \geq 16$.

be the basic sequence defined in Definition 6.1(b) with $n = n_k$ and $r = r_k$. Let $(z_i^{(k)})$ be defined by $z_i^{(k)} = \max\{z_i^{(k-1)}, x_i^{(k)}\}$. If $(x_i) = (z_i^{(k)})$, then the decomposition of (x_i) into basic sequences is $(x_i^{(1)}), (x_i^{(2)}), \dots, (x_i^{(k)})$. Otherwise increment the value of k by one and repeat the process.

The number of repetitions in the procedure of the previous paragraph is finite, since every sequence (x_i) in \mathcal{A} is non-increasing and assumes at some $n \in \mathbb{N}$ a value of $x_i = 1$ for all $i \geq n$. Thus, $\mathcal{A} \subseteq \mathcal{B}^*$. ■

The decomposition of a sequence in \mathcal{A} into basic sequences as was done in the proof of Proposition 6.1 is illustrated in the next example.

Example 6.2 Consider the sequence 955443222222221 in \mathcal{A} . The values of the parameters used in the proof of Proposition 6.1 are given below.

Iteration	Parameter Values
Initialization	$(z_i^{(0)}) = 1\ 1\ 1\ \dots$
$k = 1$	$j_1 = 0, n_1 = 9$ and $r_1 = 1$ $(x_i^{(1)}) = 9\ 5\ 3\ 3\ 2\ 2\ 2\ 2\ 1$ $(z_i^{(1)}) = 9\ 5\ 3\ 3\ 2\ 2\ 2\ 2\ 1$
$k = 2$	$j_2 = 2, n_2 = 5$ and $r_2 = 3$ $(x_i^{(2)}) = 5\ 5\ 5\ 3\ 3\ 3\ 2\ 2\ 2\ 2\ 2\ 1$ $(z_i^{(2)}) = 9\ 5\ 5\ 3\ 3\ 3\ 2\ 2\ 2\ 2\ 2\ 1$
$k = 3$	$j_3 = 3, n_3 = 4$ and $r_3 = 5$ $(x_i^{(3)}) = 4\ 4\ 4\ 4\ 4\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1$ $(z_i^{(3)}) = 9\ 5\ 5\ 4\ 4\ 3\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 2\ 1$

Therefore the sequence 955443222222221 may be decomposed into basic sequences 953322221, 555333222221 and 444442222222221. ■

6.2 Δ -Chromatic Basic Sequences

Notice that some of the graphs listed in Table C.1 as achieving the given Δ -chromatic sequences, are the unions of two or more graphs. For any positive integer d and any two graphs G and H suppose a χ_d^Δ -colouring of G in x colours and a χ_d^Δ -colouring of H in y colours exist. Then the union of G and H requires $\max\{x, y\}$ colours for a χ_d^Δ -colouring of $G \cup H$. Therefore, the maximum degree chromatic number, $\chi_d^\Delta(G \cup H)$, of the union of G and H is at most the maximum of the maximum degree chromatic numbers of G and H . Suppose $\max\{x, y\} = y$. Then $\chi_d^\Delta(G \cup H) \geq \chi_d^\Delta(H)$ since $H \subseteq (G \cup H)$. Similar arguments follow if $\max\{x, y\} = x$. Hence, the maximum degree chromatic number, $\chi_d^\Delta(G \cup H)$, of the union of G and H is also at least the maximum of the maximum degree chromatic numbers of G and H . Thus, $\chi_d^\Delta(G \cup H) = \max\{\chi_d^\Delta(G), \chi_d^\Delta(H)\}$. The significance of the notion of basic sequences in terms of the characterization of maximum degree chromatic sequences, may now be stated in the next proposition.

Proposition 6.2 All sequences in \mathcal{A} are Δ -chromatic if and only if all basic sequences in \mathcal{B} are Δ -chromatic.

Proof: Suppose all sequences of \mathcal{A} are Δ -chromatic. Then all basic sequences are Δ -chromatic, since $\mathcal{B} \subseteq \mathcal{B}^* = \mathcal{A}$ by Proposition 6.1.

Now, suppose all basic sequences are Δ -chromatic. From the proof of Proposition 6.1, any sequence (x_i) in \mathcal{A} may be decomposed into basic sequences $(x_i^{(1)}), (x_i^{(2)}), \dots, (x_i^{(k)})$ with $x_i = \max\{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(k)}\}$. Let $G^{(j)}$ be the graph with Δ -chromatic sequence $(x_i^{(j)})$ for all $j = 1, 2, \dots, k$. Then the graph $G = G^{(1)} \cup G^{(2)} \cup \dots \cup G^{(k)}$ has the Δ -chromatic sequence (z_i) with $z_i = \max\{x_i^{(1)}, x_i^{(2)}, \dots, x_i^{(k)}\}$; hence (x_i) is the Δ -chromatic sequence of G . ■

Studying the graph constructions in Table C.1, one notices that all the graphs are complete graphs, complete balanced multipartite graphs or unions of these graph classes. The maximum degree chromatic number of a complete graph was established in §4.3. All basic sequences that may be classified as Δ -chromatic with the help of a complete graph are identified in the following proposition.

Proposition 6.3 *All basic sequences (x_i) with $r = 1$ are Δ -chromatic.*

Proof: From Proposition 4.4 the complete graph K_n , where $n \in \mathbb{N}$, satisfies $\chi_d^\Delta(K_n) = \lceil n/(d+1) \rceil$ for all $d \geq 0$. Therefore, the Δ -chromatic sequence (x_i) of K_n satisfies $x_{k-1} = \lceil n/k \rceil$ for all $k \in \mathbb{N}$, which is exactly the basic sequence defined in Definition 6.1(b) with $r = 1$. ■

Some basic sequences not explicitly listed in Table C.1 and in particular the basic sequences with more than the minimum number of twos in the sequence (as well as the sequence $(x_i) = 2 \ 1 \ 1 \ \dots$) are Δ -chromatic, as proved in the following proposition.

Proposition 6.4 *All basic sequences (x_i) with $n = 2$ are Δ -chromatic.*

Proof: From Proposition 4.1 the complete bipartite graph $K_{m,n}$, where $m, n \in \mathbb{N}$, satisfies $\chi_d^\Delta(K_{m,n}) = 2$ for all $0 \leq d < \max\{m, n\}$ and $\chi_d^\Delta(K_{m,n}) = 1$ for all $d \geq \max\{m, n\}$. Therefore, the Δ -chromatic sequence (x_i) of $K_{m,n}$ satisfies $x_0 = x_1 = \dots = x_{k-1} = 2$ and $x_k = 1$, where $k = \max\{m, n\}$. ■

The maximum degree chromatic number of a complete balanced multipartite graph was studied in §4.6. Special types of complete balanced multipartite graphs may be used to classify certain basic sequences as Δ -chromatic sequences, as outlined in the next propositions.

Proposition 6.5 *All basic sequences (x_i) with $r = 2$ are Δ -chromatic.*

Proof: Let $k \in \mathbb{N} \setminus \{1\}$ and $\ell \in \mathbb{N}$. Consider the basic sequence (x_i) defined by $x_i = \lceil k/\ell \rceil$ for all $i \in \{2\ell - 2, 2\ell - 1\}$. It is shown that (x_i) is the Δ -chromatic sequence of the complete balanced multipartite graph $K_{k \times 2}$.

Let $d \in \{2\ell, 2\ell + 1\}$ for some integer $\ell \geq 0$. Let u_i and v_i be the two vertices in partite set i of the graph $K_{k \times 2}$ for all $i = 1, \dots, k$. Suppose in a colouring of $K_{k \times 2}$ the vertices u_i and v_i are in colour class C_p for any $\ell + 1$ values of $i \in \{1, \dots, k\}$. Then $\Delta(\langle C_p \rangle) = 2\ell \leq d$. Therefore, in a $\Delta(2\ell, x)$ -colouring of $K_{k \times 2}$ where both vertices in a partite set are monocoloured and each colour class contains $2\ell + 2$ vertices, except perhaps one colour class that will contain fewer than $2\ell + 2$ vertices, $x = \lceil k/(\ell + 1) \rceil$ colours are used. Now suppose in another $\Delta(2\ell, x)$ -colouring of $K_{k \times 2}$ for some $i \in \{1, \dots, k\}$ vertex u_i is coloured with colour p and vertex v_i is coloured with colour q where $q \neq p$. Then, only the two vertices in ℓ other partite sets may be coloured with colour p as well. Similarly for colour q . Therefore, both colour classes C_p and C_q contain fewer than $2\ell + 2$ vertices. Therefore, in a $\Delta(2\ell, x)$ -colouring of $K_{k \times 2}$ where in some partite sets the two vertices are not monocoloured, at least two colour classes have fewer than $2\ell + 2$ vertices and thus, such a $\Delta(2\ell, x)$ -colouring will require at least as many colours as one where both vertices in a partite set are monocoloured for all values of $i \in \{1, \dots, k\}$. Thus, $\chi_d^\Delta(K_{k \times 2}) = \lceil k/(\ell + 1) \rceil$ for $d \in \{2\ell, 2\ell + 1\}$ and $\ell = 0, 1, 2, \dots$, i.e. $\chi_d^\Delta(K_{k \times 2}) = \lceil k/\ell \rceil$ for $d \in \{2\ell, 2\ell + 1\}$ and $\ell = 1, 2, 3, \dots$. Hence, (x_i) is the Δ -chromatic sequence of $K_{k \times 2}$. ■

Proposition 6.6 *All basic sequences (x_i) with $n = 3$ are Δ -chromatic.*

Proof: Two colours are insufficient for a χ_d^Δ -colouring of the complete balanced 3-partite graph $K_{3 \times r}$ if $0 \leq d < r$, since at least one vertex in a colour class of each 2-colouring will be adjacent to r vertices in the same colour class. However, if $0 \leq d < r$, then a χ_d^Δ -colouring of $K_{3 \times r}$ in three colours may be obtained by colouring all the vertices in the same partite set with the same colour.

If $r \leq d < 2r$, then one colour is insufficient for a χ_d^Δ -colouring of $K_{3 \times r}$, since $d < \Delta(K_{3 \times r}) = 2r$. However, if $r \leq d < 2r$, then a χ_d^Δ -colouring of $K_{3 \times r}$ in two colours may be obtained by colouring all the vertices in two of the partite sets with one colour and all the vertices in the third partite set with a second colour. Finally, if $d \geq 2r$ one colour is sufficient for a χ_d^Δ -colouring of $K_{3 \times r}$, since $d \geq \Delta(K_{3 \times r}) = 2r$.

Therefore the Δ -chromatic sequence (x_i) of $K_{3 \times r}$ satisfies $x_i = 3$ for all $i \in \{0, 1, \dots, r-1\}$ and $x_i = 2$ for all $i \in \{r, r+1, \dots, 2r-1\}$, while ending with a tail of ones from $i = 2r$ onwards. These are exactly the basic sequences defined in Definition 6.1(b) with $n = 3$. ■

Two more basic sequences may be classified as Δ -chromatic sequences with the help of complete balanced multipartite graphs.

Proposition 6.7 *The basic sequence $(x_i) = 4442222221 \dots$ is Δ -chromatic.*

Proof: Three colours are insufficient for a χ_d^Δ -colouring of the complete balanced 4-partite graph $K_{4 \times 3}$ if $0 \leq d < 3$, since at least one vertex in a colour class of a 3-colouring will be adjacent to three vertices in the same colour class. However, if $0 \leq d < 3$, then a χ_d^Δ -colouring of $K_{4 \times 3}$ in four colours may be obtained by colouring all the vertices in the same partite set with the same colour.

If $3 \leq d < 9$, then one colour is insufficient for a χ_d^Δ -colouring of $K_{4 \times 3}$, since $d < \Delta(K_{4 \times 3}) = 9$. However, if $3 \leq d < 9$, then a χ_d^Δ -colouring of $K_{4 \times 3}$ in two colours may be obtained by colouring all the vertices in two of the partite sets with the same colour and all the vertices in the remaining two partite sets with a second colour. Finally, if $d \geq 9$ one colour is sufficient for a χ_d^Δ -colouring of $K_{4 \times 3}$, since $d \geq \Delta(K_{4 \times 3}) = 9$. Therefore the Δ -chromatic sequence of $K_{4 \times 3}$ is given by $(x_i) = 4442222221 \dots$ ■

Proposition 6.8 *The basic sequence $(x_i) = 5553332222221 \dots$ is Δ -chromatic.*

Proof: Four colours are insufficient for a χ_d^Δ -colouring of the complete balanced 5-partite graph $K_{5 \times 3}$ if $0 \leq d < 3$, since at least one vertex in a colour class of any 4-colouring will be adjacent to three vertices in the same colour class. However, if $0 \leq d < 3$, then a χ_d^Δ -colouring of $K_{5 \times 3}$ in five colours may be obtained by colouring all the vertices in the same partite set with the same colour.

If $3 \leq d < 6$, then two colours are insufficient for a χ_d^Δ -colouring of $K_{5 \times 3}$, since at least one vertex in a colour class of any 2-colouring will be adjacent to 6 vertices in the same colour class. However, if $3 \leq d < 6$, then a χ_d^Δ -colouring of $K_{5 \times 3}$ in three colours may be obtained by colouring all the vertices in two of the partite sets with one colour, all the vertices in two further partite sets with a second colour and all the vertices in the remaining partite set with a third colour.

If $6 \leq d < 12$, then one colour is insufficient for a χ_d^Δ -colouring of $K_{5 \times 3}$, since $d < \Delta(K_{5 \times 3}) = 12$. However, if $6 \leq d < 12$, then a χ_d^Δ -colouring of $K_{5 \times 3}$ in two colours may be obtained by colouring all the vertices in three of the partite sets with one colour and all the vertices in the other two partite set with a second colour.

Finally, if $d \geq 12$, one colour is sufficient for a χ_d^Δ -colouring of $K_{5 \times 3}$, since $d \geq \Delta(K_{5 \times 3}) = 12$. Therefore the Δ -chromatic sequence of $K_{5 \times 3}$ is given by $(x_i) = 5553332222221 \dots$ ■

Basic sequences that have thus far been classified as Δ -chromatic (as well as graphs achieving these Δ -chromatic sequences) are shown in Table 6.2.

In §4.6 an algorithmic upper bound on the maximum degree chromatic number of a complete balanced multipartite graph $K_{k \times n}$ was determined. This algorithm was implemented in MATLAB to determine upper bounds on the values in the maximum degree sequence for each complete balanced multipartite graph $K_{k \times n}$, given ranges of values for k and n . An extract of the output of this program is given in Table 6.3. It may be seen from the table that, except for the complete balanced multipartite graphs in Propositions 6.5–6.8, the Δ -chromatic sequences of most complete balanced multipartite graphs are not basic. One may notice why this is the case by studying the basic sequences in Table 6.2 that have been classified as Δ -chromatic with the help of complete balanced multipartite graphs. In all cases where the Δ -chromatic sequence of the complete balanced multipartite graph $K_{k \times n}$ is basic, the Δ -chromatic sequence has $x_i = k$ for all $i \in \{0, 1, \dots, n-1\}$ and for $0 \leq d < n$ a χ_d^Δ -colouring of $K_{k \times n}$ in k colours may be obtained by colouring all the vertices in the same partite set with the same colour. However, if

$$\left\lfloor \frac{k}{2} \right\rfloor + \left\lceil \frac{2(n-d) \lfloor \frac{k}{2} \rfloor}{d+1} \right\rceil + (k \pmod{2}) < k \quad (6.2.1)$$

n	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	Graph
1	1																	K_1^\dagger
2	2	1																$K_2^{\dagger,*}$
	2	2	1															$K_{m,2}^{*,\ddagger}$
	2	2	2	1														$K_{m,3}^*$
	2	2	2	2	1													$K_{m,4}^*$
	\vdots																	\vdots
	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	$K_{m,16}^*$
3	3	2	1															$K_3^{\dagger,\circ}$
	3	3	2	2	1													$K_{3 \times 2}^{\ddagger,\circ}$
	3	3	3	2	2	2	1											$K_{3 \times 3}^\circ$
	\vdots																	\vdots
	3	3	3	3	3	3	3	3	2	2	2	2	2	2	2	2	1	$K_{3 \times 8}^\circ$
4	4	2	2	1														K_4^\dagger
	4	4	2	2	2	2	1											$K_{4 \times 2}^{\ddagger}$
	4	4	4	2	2	2	2	2	2	1								$K_{4 \times 3}^\bullet$
	4	4	4	4	2	2	2	2	2	2	2	2	1					?
	4	4	4	4	4	2	2	2	2	2	2	2	2	2	2	1		?
5	5	3	2	2	1													K_5^\dagger
	5	5	3	3	2	2	2	2	1									$K_{5 \times 2}^{\ddagger}$
	5	5	5	3	3	3	2	2	2	2	2	1						$K_{5 \times 3}^\bullet$
	5	5	5	5	3	3	3	3	3	2	2	2	2	2	2	2	1	?
6	6	3	2	2	2	1												K_6^\dagger
	6	6	3	3	2	2	2	2	2	2	1							$K_{6 \times 2}^{\ddagger}$
	6	6	6	3	3	3	2	2	2	2	2	2	2	2	2	1		?
7	7	4	3	2	2	2	1											K_7^\dagger
	7	7	4	4	3	3	2	2	2	2	2	2	1					$K_{7 \times 2}^{\ddagger}$
8	8	4	3	2	2	2	2	1										K_8^\dagger
	8	8	4	4	3	3	2	2	2	2	2	2	2	2	1			$K_{8 \times 2}^{\ddagger}$
9	9	5	3	3	2	2	2	2	1									K_9^\dagger
	9	9	5	5	3	3	3	3	2	2	2	2	2	2	2	2	1	$K_{9 \times 2}^{\ddagger}$
10	10	5	4	3	2	2	2	2	2	1								K_{10}^\dagger

TABLE 6.2: The basic sequences starting with $n \in \{1, 2, \dots, 10\}$ and with length at most 17, and the corresponding graphs on the right hand side for those basic sequences that are Δ -chromatic sequences. Graphs obtained from \dagger Proposition 6.3. $*$ Proposition 6.4, where $m \in \mathbb{N}$ such that for $K_{m,n}$, $m \leq n$. \ddagger Proposition 6.5. \circ Proposition 6.6. \bullet Propositions 6.7 and 6.8, respectively. Basic sequences in bold face are not as yet classified as Δ -chromatic sequences.

for some $0 \leq d < n$, then the above k -colouring strategy is no longer optimal. In this case, for x equal to the lefthand side of (6.2.1) and $0 \leq d < n$, a χ_d^Δ -colouring of $K_{k \times n}$ in $x (< k)$ colours may be found by means of the algorithmic strategy in §4.6.

For the complete balanced 3-partite graph $K_{3 \times n}$ and (6.2.1), it follows that $2 + \lceil 2(n-d)/(d+1) \rceil \geq 3$ for all $n \in \mathbb{N}$ and $0 \leq d < n$, confirming the fact that the colouring strategy in Proposition 6.6) is optimal for the complete balanced 3-partite graph $K_{3 \times n}$. However, for $k \geq 4$ the complete balanced k -partite graph $K_{k \times n}$ often cannot be used to classify any of the basic sequences as Δ -chromatic, as indicated in Table 6.4 where the smallest values of n for which (6.2.1) holds true for some $0 \leq d < n$, are summarised for $4 \leq k \leq 10$.

Graph	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}
$K_{3 \times 2}$	≤ 3	≤ 3	2	2	1												
$K_{3 \times 3}$	≤ 3	≤ 3	≤ 3	2	2	2	1										
$K_{3 \times 4}$	≤ 3	≤ 3	≤ 3	≤ 3	2	2	2	2	1								
$K_{3 \times 5}$	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	2	2	2	2	2	1						
$K_{3 \times 6}$	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	2	2	2	2	2	2	1				
$K_{3 \times 7}$	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	2	2	2	2	2	2	2	1		
$K_{3 \times 8}$	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	≤ 3	2	2	2	2	2	2	2	2	1
$K_{4 \times 2}$	≤ 4	≤ 4	2	2	2	2	1										
$K_{4 \times 3}$	≤ 4	≤ 4	≤ 4	2	2	2	2	2	2	1							
$K_{4 \times 4}$	≤ 4	≤ 4	≤ 4	≤ 3	2	2	2	2	2	2	2	2	1				
$K_{4 \times 5}$	≤ 4	≤ 4	≤ 4	≤ 4	≤ 3	2	2	2	2	2	2	2	2	2	2	1	
$K_{5 \times 2}$	≤ 5	≤ 5	≤ 3	≤ 3	2	2	2	2	1								
$K_{5 \times 3}$	≤ 5	≤ 5	≤ 5	≤ 3	≤ 3	≤ 3	2	2	2	2	2	2	1				
$K_{5 \times 4}$	≤ 5	≤ 5	≤ 5	≤ 4	≤ 3	≤ 3	≤ 3	≤ 3	2	2	2	2	2	2	2	2	1
$K_{6 \times 2}$	≤ 6	≤ 6	≤ 3	≤ 3	2	2	2	2	2	2	1						
$K_{6 \times 3}$	≤ 6	≤ 6	≤ 5	≤ 3	≤ 3	≤ 3	2	2	2	2	2	2	2	2	2	1	

TABLE 6.3: Upper bounds on the values in the maximum degree sequence for some complete balanced multipartite graphs as determined by the algorithmic strategy in §4.6.

k	Inequality to be satisfied by n and d	n	range for d	d
4	$d \geq (4n - 1)/5$	4	$0 \leq d < 4$	3
5	$d \geq (4n - 1)/5$	4	$0 \leq d < 4$	3
6	$d \geq (3n - 1)/4$	3	$0 \leq d < 3$	2
7	$d \geq (3n - 1)/4$	3	$0 \leq d < 3$	2
8	$d \geq (8n - 3)/11$	3	$0 \leq d < 3$	2
9	$d \geq (8n - 3)/11$	3	$0 \leq d < 3$	2
10	$d \geq (5n - 2)/7$	3	$0 \leq d < 3$	2

TABLE 6.4: The smallest values of n and d for which (6.2.1) holds true, for $4 \leq k \leq 10$. In the second column the inequality to be satisfied by d and n in order for (6.2.1) to hold true, is listed.

6.3 The Basic Sequence 444422222221

In §6.2 graphs were presented by which basic sequences may be classified as Δ -chromatic sequences. The first basic sequence in Table 6.2 for which no such graph could be found, is the sequence 44442222222221. In this section some characteristics of a graph achieving this sequence are given (if such a graph exists).

First of all, from (3.2.1) it follows that a graph G with Δ -chromatic sequence 4444222222221, contains a clique of order at most 4, since $\chi_0^\Delta(G) = 4$. Also, since $\chi_0^\Delta(G) = \chi(G) = 4$, the graph G must be 4-partite. Furthermore, since $\chi_{12}^\Delta(G) = 1$ and $\chi_{11}^\Delta(G) > 1$ it follows that $\Delta(G) = 12$.

6.3.1 On the Order of a Graph with Δ -chromatic Sequence 444422222221

From (3.2.2) it follows that a graph G with Δ -chromatic sequence 4444222222221, must have an order of at least 13. However, a better lower bound on the order of a graph G with Δ -chromatic sequence 4444222222221 may be obtained, as outlined by the following proposition.

Proposition 6.9 *Any graph with Δ -chromatic sequence 4444222222221 has order at least 17.*

Proof: Let G be a graph with Δ -chromatic sequence 4444222222221. Then G is 4-partite and $\Delta(G) = 12$. Assume, to the contrary, that G has at most 16 vertices and consider a 4-partition V_1, V_2, V_3, V_4 of the vertices of G . Assume, without loss of generality, that the partite sets V_1, V_2, V_3 and V_4 are ordered in non-decreasing order of cardinality. If $|V_i| \leq 4$ for all $i = 1, 2, 3, 4$, choose $v_i \in V_i$ for all $i = 1, 2, 3, 4$. Then the colouring with colour classes $\mathcal{C}_1 = \{v_1, v_2, v_3, v_4\}$, $\mathcal{C}_2 = (V_1 - \{v_1\}) \cup (V_2 - \{v_2\})$ and $\mathcal{C}_3 = (V_3 - \{v_3\}) \cup (V_4 - \{v_4\})$ shows that $\chi_3^\Delta(G) \leq 3$, contradicting the supposition that $x_3 = 4$. If $|V_1| \leq 3$ and $|V_2| \leq 3$ then the colouring with colour classes $\mathcal{C}_1 = V_1 \cup V_2$, $\mathcal{C}_2 = V_3$ and $\mathcal{C}_3 = V_4$ shows that $\chi_3^\Delta(G) \leq 3$, again contradicting the supposition that $x_3 = 4$.

Therefore, assume that $|V_1| \leq 3$, $|V_2| \geq 4$, $|V_3| \geq 4$ and $|V_4| \geq 5$. Choose $v_1, v_2, v_3 \in V_2$ and choose $v_4, v_5, v_6 \in V_3$, and consider the colour classes $\mathcal{C}_1 = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $\mathcal{C}_2 = V_4$ and $\mathcal{C}_3 = V(G) - (\mathcal{C}_1 \cup \mathcal{C}_2)$. If $|\mathcal{C}_3| = 3$ or $|\mathcal{C}_3| = 4$, then the above colouring shows that $\chi_3^\Delta(G) \leq 3$, again contradicting the supposition that $x_3 = 4$. Therefore, suppose that $|\mathcal{C}_3| = 5$. There are three cases to be considered.

Case (a): $|V_1| = 1$, $|V_2| = 5$, $|V_3| = 5$ and $|V_4| = 5$. Let v_7 be the vertex in V_1 . Since $\Delta(G) = 12$, v_7 cannot be adjacent to all vertices in V_2, V_3 and V_4 . Assume, without loss of generality that v_7 is non-adjacent to $v_8 \in (V_2 - \{v_1, v_2, v_3\})$. Then the above colouring shows that $\chi_3^\Delta(G) \leq 3$, again contradicting the supposition that $x_3 = 4$.

Case (b): $|V_1| = 2$, $|V_2| = 4$, $|V_3| = 5$ and $|V_4| = 5$. Since $\Delta(G) = 12$, $v_7, v_8 \in V_1$ cannot be adjacent to all vertices in V_2, V_3 and V_4 . In particular, $v_7, v_8 \in V_1$ must be non-adjacent to at least two vertices in V_2, V_3 and V_4 . There are three possibilities of non-adjacency to consider in this case.

Case (b)(1): Assume, without loss of generality, that $v_7 \in V_1$ is non-adjacent to $v_9 \in (V_2 - \{v_1, v_2, v_3\})$. Then the above colouring shows that $\chi_3^\Delta(G) \leq 3$, contradicting the supposition that $x_3 = 4$.

Case (b)(2): Assume, without loss of generality, that $v_7 \in V_1$ is non-adjacent to any two vertices in V_3 . Choose $v_9, v_{10} \in V_4$. Then the colouring with new colour classes $\mathcal{C}_1 = \{v_7\} \cup V_3$, $\mathcal{C}_2 = \{v_1, v_8, v_9, v_{10}\}$ and $\mathcal{C}_3 = (V_2 - \{v_1\}) \cup (V_4 - \{v_9, v_{10}\})$ shows that $\chi_3^\Delta(G) \leq 3$, contradicting the supposition that $x_3 = 4$.

Case (b)(3): Assume, without loss of generality, that $v_7 \in V_1$ is non-adjacent to $v_4 \in V_3$ and $v_9 \in V_4$, and that $v_8 \in V_1$ is non-adjacent to $v_5 \in V_3$ and $v_{10} \in V_4$. If $v_4 = v_5$ [$v_9 = v_{10}$, respectively], then v_4 [v_9 , respectively] may also be included in V_1 during the original 4-partition of G . Thus, the colouring problem is the same as the colouring problem with $|V_1| = 3$, $|V_2| = 4$, $|V_3| = 4$ and $|V_4| = 5$. (case (c) below). Hence, assume $v_4 \neq v_5$ and $v_9 \neq v_{10}$. Then the colouring with new colour classes $\mathcal{C}_1 = V_1 \cup \{v_4, v_5, v_9, v_{10}\}$, $\mathcal{C}_2 = V_2$ and $\mathcal{C}_3 = (V_3 - \{v_4, v_5\}) \cup (V_4 - \{v_9, v_{10}\})$ shows that $\chi_3^\Delta(G) \leq 3$, contradicting the supposition that $x_3 = 4$.

Case (c): $|V_1| = 3$, $|V_2| = 4$, $|V_3| = 4$ and $|V_4| = 5$. Choose $v_7, v_8 \in V_4$. Then the colouring with colour classes $\mathcal{C}_1 = \{v_1, v_4, v_7, v_8\}$, $\mathcal{C}_2 = V_1 \cup (V_2 - \{v_1\})$ and $\mathcal{C}_3 = (V_3 - \{v_4\}) \cup (V_4 - \{v_7, v_8\})$ shows that $\chi_3^\Delta(G) \leq 3$, yet again contradicting the supposition that $x_3 = 4$.

In all the above cases $x_3 \leq 3$, which contradicts the given basic sequence. Thus, a graph with Δ -chromatic sequence 4444222222221 has order at least 17. ■

6.3.2 On the Size of a Graph with Δ -chromatic Sequence 4444222222221

In order to determine upper bounds on the size of a graph of given order with Δ -chromatic numbers in the sequence 4444222222221, the following lemma is necessary.

Lemma 6.1 *The maximum number of edges in a complete k -partite graph of order n , occurs when the cardinality of all the partite sets are the same or differ by at most 1 if $k \nmid n$.*

Proof: The maximum number of edges in a complete k -partite graph of order n may be achieved by removing the minimum number of edges from K_n to form the complete k -partite graph. Suppose first that $k \mid n$ and suppose all partite sets are of cardinality n/k . Then $2\binom{n/k}{2} = n/k(n/k - 1)$ edges have to

be removed from K_n to form two of the partite sets, say partite set i and partite set j . Now suppose partite set i has cardinality $n/k + \ell$ and partite set j has cardinality $n/k - \ell$, while the other $k - 2$ partite sets still have cardinality n/k . In this case the number of edges removed from K_n to form partite set i and partite set j , are $\binom{n/k + \ell}{2} + \binom{n/k - \ell}{2} = n/k(n/k - 1) + \ell^2$. Therefore in this case more edges were removed from K_n to form partite sets i and j than before.

Next, suppose $k \nmid n$ and let $\alpha = n - (n \pmod{k})$. Suppose also that $n \pmod{k}$ partite sets are of cardinality $\alpha/k + 1$ and the remaining partite sets are of cardinality α/k . Consider three partite sets, where one is of cardinality $\alpha/k + 1$ (partite set i) and the other two partite sets are of cardinality α/k (partite sets j and m). Then $\binom{\alpha/k + 1}{2} + 2\binom{\alpha/k}{2} = 3/2(\alpha/k)^2 - 1/2(\alpha/k)$ edges have to be removed from K_n to form these partite sets. Now suppose the cardinality of one of these partite sets is more than one less than the other two, *i.e.* partite set j is now also of cardinality $\alpha/k + 1$, while the cardinality of partite set m becomes $\alpha/k - 1$. (The cardinality of partite set j is still $\alpha/k + 1$.) In this case the number of edges removed from K_n to form partite sets i , j and m are $2\binom{\alpha/k + 1}{2} + \binom{\alpha/k - 1}{2} = 3/2(\alpha/k)^2 - 1/2(\alpha/k) + 1$. In this case, therefore, more edges have to be removed from K_n to form partite sets i and j than before. ■

An upper bound on the size of a graph of order n for which $x_0 = 4$, is given below.

Proposition 6.10 *For any graph G of order n with $\chi_0^\Delta(G) = 4$,*

$$q(G) \leq \binom{n}{2} - \left[\binom{\lfloor n/4 \rfloor + 1}{2} \times (n \pmod{4}) + \binom{\lfloor n/4 \rfloor}{2} \times (4 - n \pmod{4}) \right].$$

Proof: Since $\chi_0^\Delta(G) = 4$, the graph G is 4-partite. Suppose the number of vertices in each partite set is n_1, n_2, n_3 and n_4 , respectively. Then, one may view G as the subgraph of K_n from which the edges of cliques of orders n_1, n_2, n_3 and n_4 , respectively, have been removed. In order to determine the maximum size of G , one wishes to remove the smallest number of edges from K_n in the manner described above. It follows by Lemma 6.1 that the smallest total number of edges in the cliques occurs when the orders of the four cliques differ by at most 1. Hence, $4 - n \pmod{4}$ partite sets are of order $\lfloor n/4 \rfloor$, and $n \pmod{4}$ partite sets are of order $\lfloor n/4 \rfloor + 1$. Therefore, the maximum size of G is obtained by subtracting $\binom{\lfloor n/4 \rfloor + 1}{2} \times (n \pmod{4}) + \binom{\lfloor n/4 \rfloor}{2} \times (4 - n \pmod{4})$ from a possible $\binom{n}{2}$ edges for a graph of order n . ■

If $\chi_d^\Delta(G) = m$ for some graph G , it is clear that a graph G' with $\chi_{d+1}^\Delta(G') = m$ may contain more edges than G . Hence, only the positions in the sequence 444422222221 where a specific number appears for the first time are considered. Therefore, an upper bound on the size of a graph of order n for which $x_4 = 2$, is considered next.

Proposition 6.11 *For any graph G of order n with $\chi_4^\Delta(G) = 2$,*

$$q(G) \leq \left(2 + \left\lfloor \frac{n}{2} \right\rfloor \right) \left\lceil \frac{n}{2} \right\rceil + 2 \left\lfloor \frac{n}{2} \right\rfloor.$$

Proof: Since $\chi_4^\Delta(G) = 2$, there exists a 2-partition of $V(G)$ such that every partition-induced maximum degree is at most 4. From Lemma 6.1 the maximum number of edges in G occurs when the cardinalities of the two partition sets differ by at most 1. Thus, the partition sets of the partition comprise $\lfloor n/2 \rfloor$ and $\lceil n/2 \rceil$ vertices, respectively. The maximum number of edges wholly within each partition set occurs when both partition-induced subgraphs of G are 4-regular, *i.e.* when $2\lfloor n/2 \rfloor$ edges join vertices in one partition set and when $2\lceil n/2 \rceil$ edges join vertices in the other partition set. Finally, the maximum number of edges between vertices in one partition set and vertices in the other partition set is $\lfloor n/2 \rfloor \times \lceil n/2 \rceil$. ■

Finally, an upper bound on the size of a graph of order n for which $x_{12} = 1$, is given below.

Proposition 6.12 *For any graph G of order n with $\chi_{12}^\Delta(G) = 1$, $q(G) \leq 6n$.*

Proof: Since $\chi_{12}^\Delta(G) = 1$ it follows that $\Delta(G) = 12$. Thus, the maximum number of edges in G occurs when G is 12-regular. From Theorem 2.1 it follows that $q = 6n$ for a 12-regular graph. ■

An upper bound on the size of a graph G of order n with Δ -chromatic sequence 4 4 4 4 2 2 2 2 2 2 2 1 may now be given as

$$q(G) \leq \min \left\{ \binom{n}{2} - \left[\binom{\lfloor n/4 \rfloor + 1}{2} \times n \pmod{4} + \binom{\lfloor n/4 \rfloor}{2} \times (4 - n \pmod{4}) \right], \right. \\ \left. \left(2 + \lfloor \frac{n}{2} \rfloor \right) \left\lceil \frac{n}{2} \right\rceil + 2 \lfloor \frac{n}{2} \rfloor, 6n \right\}. \quad (6.3.1)$$

Values for the different upper bounds on $q(G)$ in (6.3.1) for $17 \leq n \leq 30$ are given in Table 6.5, where in all cases Proposition 6.12 provides the best upper bound.

n	Bound on $q(G)$ in Proposition 6.10	Bound on $q(G)$ in Proposition 6.11	Bound on $q(G)$ in Proposition 6.12	Minimum in (6.3.1)
17	108	106	102	102
18	121	117	108	108
19	135	128	114	114
20	150	140	120	120
21	165	152	126	126
22	181	165	132	132
23	198	178	138	138
24	216	192	144	144
25	234	206	150	150
26	253	221	156	156
27	273	236	162	162
28	294	252	168	168
29	315	268	174	174
30	337	285	180	180

TABLE 6.5: Values of the upper bounds on $q(G)$ in (6.3.1) for $17 \leq n \leq 30$.

Investigating the phenomenon reflected in Table 6.5, *i.e.* that Proposition 6.12 appears to provide the best upper bound, it may be shown that this is indeed the case. In particular, it may be shown that $(2 + \lfloor n/2 \rfloor) \lceil n/2 \rceil + 2 \lfloor n/2 \rfloor \geq 6n$ for all $n \geq 19$ and $\binom{n}{2} - \left[\binom{\lfloor n/4 \rfloor + 1}{2} \times n \pmod{4} + \binom{\lfloor n/4 \rfloor}{2} \times (4 - n \pmod{4}) \right] \geq 6n$ for all $n \geq 23$. Therefore, in the case of the basic sequence 4 4 4 4 2 2 2 2 2 2 2 1 the value $x_{12} = 1$ has a greater restriction on the size of a graph realizing this sequence (if such a graph exists) than the values $x_0 = 4$ and $x_4 = 2$.

6.4 In Search of Graphs realizing Δ -Chromatic Basic Sequences

In this section, all attempts towards finding a graph realizing a specific basic sequence, are presented. Although a general graph construction realizing a general Δ -chromatic basic sequence was sought, the emphasis in these attempts was on the basic sequence 4 4 4 4 2 2 2 2 2 2 2 1, since this is the first basic sequence that is as yet not classified as Δ -chromatic.

6.4.1 The Product K_n^r

After complete balanced multipartite graphs failed as a means to classify a general basic sequence (x_i) defined in Definition 6.1(b) as Δ -chromatic, the second part of the proof of Lovász's Theorem (Theorem 3.8) provided the idea for the next attempt. The repeated partitioning of a partition of $V(G)$ of a graph G into partite sets such that each partition induced maximum degree may be at most some

specified value, led to studying the cartesian product, K_n^r , between r copies of the complete graph, K_n , as a way to classify a basic sequence as Δ -chromatic.

It was suspected that the product K_n^r may realize the basic sequence in Definition 6.1 for a given n and r . The following corollary follows directly from the fact that $\chi(G_1 \times G_2) = \max\{\chi(G_1), \chi(G_2)\}$ for any two graphs G_1 and G_2 .

Corollary 6.1 $\chi_0^\Delta(K_n^r) = n$. ■

Therefore, given a general basic sequence (x_i) , the product K_n^r indeed gives the desired value for x_0 in the basic sequence in Definition 6.1. For a general basic sequence (x_i) , the product K_n^r also provides the desired value of d for which one colour is sufficient for a χ_d^Δ -colouring of K_n^r , in order for the product K_n^r to possibly realize the basic sequence in Definition 6.1, as stated in the following proposition.

Proposition 6.13 $\chi_{(n-1)r}^\Delta(K_n^r) = 1$.

Proof: Since $\Delta(K_n) = n - 1$, it follows that $\Delta(K_n^r) = (n - 1)r$. Hence, one colour is sufficient for a $\chi_{(n-1)r}^\Delta$ -colouring of K_n^r . ■

It should be clear that the Δ -chromatic sequence of K_2^r is the basic sequence in Definition 6.1(b) with $n = 2$. The product K_3^r also has Δ -chromatic sequence equal to the basic sequence in Definition 6.1(b) with $n = 3$. These basic sequences, however, have already been classified as Δ -chromatic in §6.2. The emphasis here is on unclassified basic sequences.

Basic Sequences with $x_0 = 4$

For the Δ -chromatic sequence (x_i) of K_4^r , it follows by Corollary 6.1 that $x_0 = 4$, giving the desired value in the basic sequence in Definition 6.1(b) with $n = 4$. Furthermore, for the Δ -chromatic sequence (x_i) of K_4^r , it follows by Proposition 6.13 that $x_{3r} = 1$ and $x_{3r-1} > 1$. But, it follows by Theorem 3.13 that $x_{3r-1} \leq x_{3r} \lceil (3r+1)/(3r) \rceil = 2$. Hence, $x_{3r-1} = 2$, again giving the desired values for x_{3r} and x_{3r-1} of the basic sequences in Definition 6.1(b) with $n = 4$. The next proposition indicates the first position in the Δ -chromatic sequence (x_i) of K_4^r , for which $x_i = 2$.

Proposition 6.14 $\chi_r^\Delta(K_4^r) = 2$.

Proof: By induction over r . The theorem is certainly true for $r = 1$, since by Propostition 4.4, $\chi_1^\Delta(K_4) = \lceil 4/(1+1) \rceil = 2$. Suppose now that $\chi_r^\Delta(K_4^r) = 2$. Let $\{u_1, \dots, u_m\}$ denote the vertices of K_4^r , where $m = 4^r$, and let $\{v_1, \dots, v_4\}$ denote the vertices of K_4 . Find a χ_r^Δ -colouring of K_4^r in two colours, and colour the vertices (u_i, v_1) of K_4^{r+1} for all $i \in \{1, \dots, m\}$ in this way, *i.e.* colour (u_i, v_1) in K_4^{r+1} with the same colour as u_i in K_4^r . Then a χ_{r+1}^Δ -colouring of K_4^{r+1} in two colours may be obtained as follows. If the vertex (u_i, v_1) of K_n^{r+1} is coloured with colour k , then colour the vertex (u_i, v_3) of K_n^{r+1} with colour k , and colour the vertices (u_i, v_2) and (u_i, v_4) of K_n^{r+1} with colour $\ell \neq k$, where $k, \ell \in \{1, 2\}$, for all $i \in \{1, \dots, m\}$. ■

These values are consistent with the desired values for x_0 as well as x_r, x_{r+1}, \dots in order to classify the basic sequence (x_i) in Definition 6.1(b) with $n = 4$, as Δ -chromatic. However, the basic sequence in Definition 6.1(b) with $n = 4$ and $r = 3$ has $x_2 = 4$ (see Example 6.1), but the Δ -chromatic sequence of K_4^3 satisfies $x_2 \leq 3$, as indicated by the $\Delta(2, 3)$ -colouring in Figure 6.1. Thus, the product K_4^r cannot be used to classify the basic sequence in Definition 6.1 for $n = 4$ and a given r as Δ -chromatic.

Other Basic Sequences

For other basic sequences the product K_n^r failed as a means to classify a general basic sequence (x_i) defined in Definition 6.1(b) as Δ -chromatic at an even earlier value for r than in the case $n = 4$. For example, the basic sequence defined in Definition 6.1(b) with $n = 6$ and $r = 2$ is 6 6 3 3 2 2 2 2 2 1. Hence,

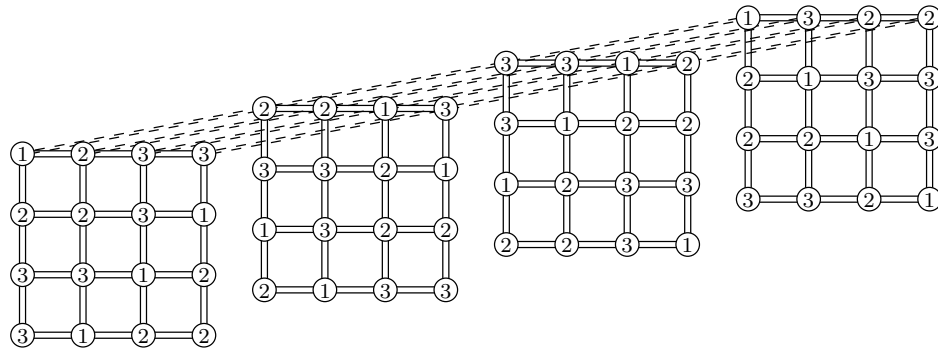


FIGURE 6.1: A graphical representation of a $\Delta(2, 3)$ -colouring of K_4^3 . The double lines between vertices in each of the four grids indicate that all the vertices in the same row or in the same column are adjacent, since each row and each column represents a copy of K_4 . As illustrated for the top row of vertices only, the dashed double lines between those vertices at the same position in each grid indicate that all the vertices along these lines are adjacent, since each dashed double line also represents a copy of K_4 .

in this sequence $x_1 = 6$. However, the Δ -chromatic sequence of K_6^2 satisfies $x_1 \leq 5$, as indicated by the $\Delta(1, 5)$ -colouring in Figure 6.2. Note that this colouring obtains the lower bound in Proposition 4.9 and is therefore also a χ_1^Δ -colouring of $K_6 \times K_6$.

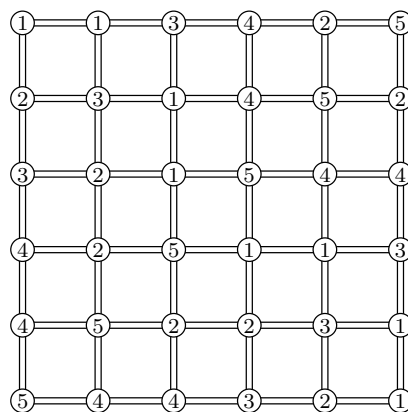


FIGURE 6.2: A graphical representation of a $\Delta(1, 5)$ -colouring of K_6^2 . The double lines between vertices indicate that all the vertices in the same row or in the same column are adjacent, since each row and each column represents a copy of K_6 .

6.4.2 The Product $K_{k \times n} \times K_4$

When the product K_n^r failed as a means to classify the basic sequence 444422222221 as Δ -chromatic, the cartesian product of K_4 and a 4-partite graph was considered, because such a cartesian product may still be a 4-partite graph, which is necessary for x_0 in the Δ -chromatic sequence of this product to be four. First the cartesian product of K_4 and the complete balanced multipartite graph $K_{4 \times 3}$ was considered. $\chi_0^\Delta(K_{4 \times 3} \times K_4) = \max\{\chi(K_{4 \times 3}), \chi(K_4)\} = 4$ and a χ_0^Δ -colouring strategy of $K_{4 \times 3} \times K_4$ may be obtained as follows. Let $\{u_1, \dots, u_{12}\}$ denote the vertices of $K_{4 \times 3}$ where $\{u_{3(k-1)+1}, u_{3(k-1)+2}, u_{3(k-1)+3}\}$ are the vertices in partite set k for all $k = 1, 2, 3, 4$ and let $\{v_1, v_2, v_3, v_4\}$ denote the vertices of K_4 . Then a χ_0^Δ -colouring of $K_{4 \times 3} \times K_4$ in 4 colours may be obtained by colouring the vertices $(u_{3(k-1)+i}, v_j)$ for all $i = 1, 2, 3$ (i.e. all vertices in the same partite set in $K_{4 \times 3}$) with colour $\ell \equiv k + j - 1 \pmod{4}$, for all $k = 1, 2, 3, 4$ and all $j = 1, 2, 3, 4$. This χ_0^Δ -colouring strategy of $K_{4 \times 3} \times K_4$ is illustrated in Figure 6.3.

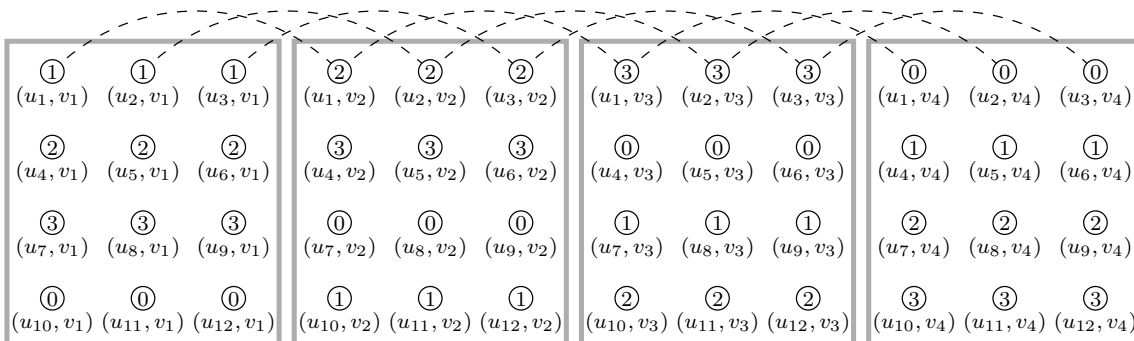


FIGURE 6.3: A graphical representation of a χ_0^Δ -colouring of $K_{4 \times 3} \times K_4$, where each block represents a copy of $K_{4 \times 3}$ and each row in a block represents a partite set. All the vertices at the same position in each block, as indicated by the dashed curves for the vertices in the top row of each block, are pairwise adjacent, since all the vertices at the same position in each block represents a copy of K_4 .

Furthermore, $\Delta(K_{4 \times 3} \times K_4) = 12$ so that $\chi_{12}^\Delta(K_{4 \times 3} \times K_4) = 1$ and $\chi_{11}^\Delta(K_{4 \times 3} \times K_4) > 1$. These values are still consistent with the desired values for x_0 , x_{11} and x_{12} in order to classify the basic sequence 4444222222221 as Δ -chromatic. However, the basic sequence 4444222222221 has $x_3 = 4$, but the Δ -chromatic sequence of $K_{4 \times 3} \times K_4$ satisfies $x_3 \leq 3$, as indicated by the $\Delta(3,3)$ -colouring in Figure 6.4. Therefore, similarly to the complete balanced multipartite graph $K_{4 \times 4}$, the graph structure $K_{4 \times 3} \times K_4$ again fails as a means to classify the basic sequence 4444222222221 as Δ -chromatic at the fourth value in the sequence.

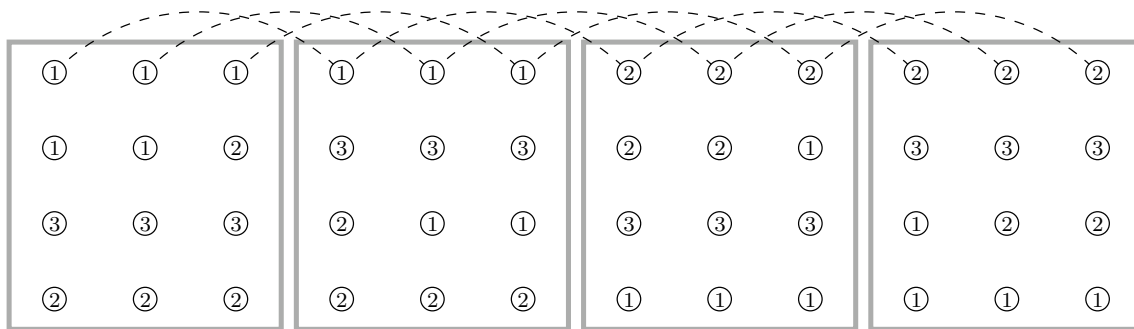


FIGURE 6.4: A graphical representation of a $\Delta(3,3)$ -colouring of $K_{4 \times 3} \times K_4$, where each block represents a copy of $K_{4 \times 3}$ and each row in a block represents a partite set. All the vertices at the same position in each block, as indicated by the dashed curves for the vertices in the top row of each block, are pairwise adjacent, since all the vertices at the same position in each block represents a copy of K_4 .

The cartesian product $G \times K_4$, where $G = K_{4 \times 4} - J$ with J an edge set such that $\Delta(G) = 9$, was also considered. Since $\deg_{K_{4 \times 4}}(v) = 12$ for each vertex $v \in V(K_{4 \times 4})$, the edge set J needs to be compiled in such a way that each vertex $v \in V(K_{4 \times 4})$ is nonadjacent to at least three other vertices in the resulting graph G . The edge set J was compiled by hand in such a way that each vertex v in G was nonadjacent to exactly one vertex in each of the other three partite sets of G , since that seemed to be the best way to delete the minimum number of edges from $K_{4 \times 4}$ such that $\deg_G(v) = 9$ for each vertex v in the resulting graph G . (During the course of this study, the author came to believe that a graph G with a higher density may force $\chi_d^\Delta(G)$ for values of d in the middle part of the lower range of $0, \dots, \Delta(G)$ to be higher than a graph with the same maximum degree and partiteness, but with a lower density.) Let $\{u_{1,1}, \dots, u_{4,4}\}$ denote the vertices of $K_{4 \times 4}$ where vertex $u_{i,j}$ is the j -th vertex in partite set i for all $i, j = 1, 2, 3, 4$. Then one example of an edge set J contains the edges $u_{i,j}u_{i+1 \pmod{4}, j+3 \pmod{4}}$, $u_{i,j}u_{i+2 \pmod{4}, j+2 \pmod{4}}$ and $u_{i,j}u_{i+3 \pmod{4}, j+1 \pmod{4}}$ for all $i, j = 1, 2, 3, 4$. Other examples of the edge set J that were considered are rather technical to describe and are not included here. Again, all these

attempts at deleting an edge set J , even with minimum cardinality, from $K_{4 \times 4}$ led to a $\Delta(3, 3)$ -colouring of $G \times K_4$. These attempts also failed as a means to classify the particular sequence as Δ -chromatic at the fourth entry in the basic sequence 444422222221.

6.4.3 Circulants

Besides the complete graphs and complete balanced multipartite graphs used to classify basic sequences as Δ -chromatic as indicated in Table 6.2, certain circulants also achieve some of these Δ -chromatic sequences. For example, the Δ -chromatic sequence of the circulant $C_8(2, 4)$ is 4221, while the circulant $C_8(1, 2, 3)$ has Δ -chromatic sequence 4422221. Circulants were therefore considered next in an attempt at classifying the basic sequence 444422222221 as Δ -chromatic. Since (from §4.5) no analytic method is known to determine the maximum degree chromatic number of a circulant, a brute force search method was developed to search for a circulant realizing the Δ -chromatic sequence 444422222221. This method was based upon a characteristic of a general Δ -chromatic sequence that is discussed next.

Let G be a graph with $\chi_d^\Delta(G) = x$. Then there exists an x -partition of $V(G)$ such that every partition induced maximum degree is at most d . Furthermore, there does *not* exist an $(x - 1)$ -partition of $V(G)$ such that every partition induced maximum degree is at most d . Suppose also that $\chi_{d-1}^\Delta(G) = x + j$. Then there exists an $(x + j)$ -partition of $V(G)$ such that every partition induced maximum degree is at most $d - 1$ and there does *not* exist an x -partition of $V(G)$ such that every partition induced maximum degree is at most $d - 1$.

For a graph G with Δ -chromatic sequence 444422222221, so that $\chi_3^\Delta(G) = 4$, it means that there exists a 4-partition of $V(G)$ such that every partition induced maximum degree is at most 3, but there does *not* exist a 3-partition of $V(G)$ such that every partition induced maximum degree is at most 3. Furthermore, since $\chi_4^\Delta(G) = 2$, there exists a 2-partition of $V(G)$ such that every partition induced maximum degree is at most 4. The brute force method mentioned above in the search for a circulant graph realizing the Δ -chromatic sequence 444422222221, used these partition principles to *eliminate* circulants from further consideration in the search for a graph with Δ -chromatic sequence 444422222221.

Since, from §2.1, a non-singular composite circulant G with connection set $\{i_1, \dots, i_z\}$ is $2z$ -regular, only non-singular composite circulants with connection sets of cardinality 6 were considered. Thus, $\chi_d^\Delta(G) \geq 2$ for all $d \in \{0, \dots, 11\}$ and $\chi_d^\Delta(G) = 1$ for all $d \geq 12$. First, a MATHEMATICA program was written to determine the chromatic numbers of a selection of 12-regular composite circulants. Any composite circulant of order n where the maximum value of any of the elements in the connection set is $\lceil n/2 \rceil - 1$ is called a **basic circulant**. For a given order n the MATHEMATICA program commences by determining 12-regular *basic* circulants only, since it follows from §2.1 that any circulant with an element in the connection set that is greater than $\lceil n/2 \rceil - 1$, is isomorphic to some basic circulant. The program, however, does not test for isomorphisms amongst the basic circulants. All basic circulants with chromatic number equal to 4 were placed in a list, while all basic circulants with chromatic number *not* equal to 4 were disregarded from further consideration since $\chi_0^\Delta(G)$ must be 4. When a certain time-out condition was satisfied before MATHEMATICA was able to determine the chromatic number of a specific circulant, this circulant was placed in a separate list. The list of basic circulants with chromatic number equal to 4, as well as the list of basic circulants for which the time-out condition was satisfied, was then given as input to a second program, which further processed these circulants. This program will be described next.

The second program was written to search among the circulants not disregarded by the first program, for a graph realizing the Δ -chromatic sequence 444422222221 by eliminating circulants not conforming to the partition characteristics. This second program, written in JAVA, searches through all 3-partitions of a given circulant G to determine whether there exists a 3-partition such that each colour class has maximum degree at most 3. As soon as such a 3-partition is found, the circulant is disregarded from any further consideration. The program also searches through all 2-partitions of circulants not disregarded so far on the 3-partition characteristic, to determine whether a 2-partition *does* exist such that each colour class has maximum degree at most 4. Five filters were used in the search through the 3- and 2-partitions of a circulant. The execution of the first three filters is much faster than the remaining

two filters. They, therefore, were placed at the beginning of the program in order to eliminate as many circulants as quickly as possible before the slower Filters 4 and 5 were executed on those circulants that had not been eliminated by Filters 1 or 2. These five filters are discussed next.

Filter 1 (Symmetric 3-partitions via binary numbers) The execution of this filter is the fastest and it eliminates many circulants. The program, therefore, commences with this filter. However, this filter is only executed if the order, n , of the circulant G is divisible by three, because this filter is based on binary numbers of length $n/3$ that are used to determine 3-partitions of the given circulant under consideration. In this regard, all binary numbers of length $n/3$ are generated. Let v_1, \dots, v_n be the vertices of a circulant of order n . Then, for each binary number, the variables $v_1, \dots, v_{n/3}$ are labelled in sequence with the corresponding digit in the binary number. Next, the variables $v_{n/3+1}, \dots, v_{2n/3}$ are also labelled in sequence with the corresponding digit in the binary number and finally the variables $v_{2n/3+1}, \dots, v_n$ are labelled in sequence with the corresponding digit in the binary number. A 3-colouring of G is formed by colouring all vertices in $\{v_1, \dots, v_{n/3}\}$ labelled with 0 and all vertices in $\{v_{n/3+1}, \dots, v_{2n/3}\}$ labelled with 1 with colour 1. All vertices in $\{v_1, \dots, v_{n/3}\}$ labelled with 1 and all vertices in $\{v_{2n/3+1}, \dots, v_n\}$ labelled with 0 are coloured with colour 2. The colouring of all vertices in $\{v_{n/3+1}, \dots, v_{2n/3}\}$ labelled with 0 and all vertices in $\{v_{2n/3+1}, \dots, v_n\}$ labelled with 1 with colour 3 completes the 3-colouring of G . The colour class induced maximum degree of only one of the colour classes is determined, since the colour class induced maximum degree of the other two colour classes will be the same by the symmetry of the colour classes. If the colour class induced maximum degree is 3, the circulant may be disregarded as a candidate graph realizing the Δ -chromatic sequence 4444222222221 ; otherwise the next binary number is generated until all binary numbers of length $n/3$ have been generated.

The strategy employed in Filter 1 is illustrated in Figure 6.5(a) for any circulant of order 27 and for the binary number 001010001 (of length 9).

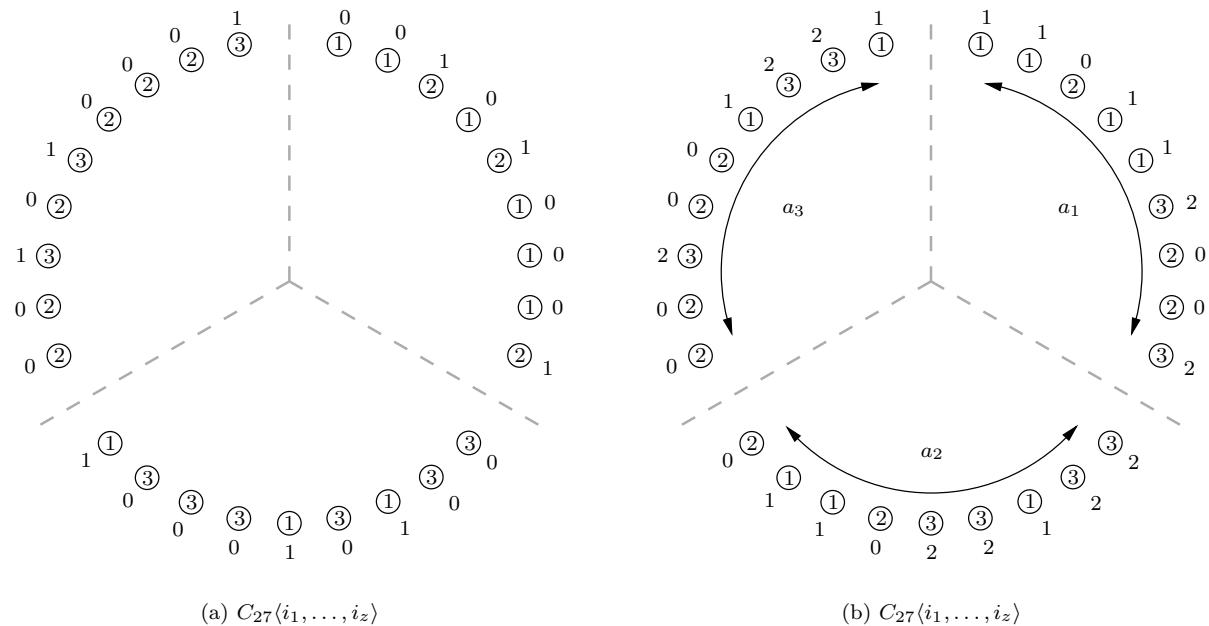


FIGURE 6.5: A graphical representation of (a) the Filter 1 strategy and (b) the Filter 2 strategy in search of a 3-partition of a circulant for which each partition induced maximum degree is at most 3.

Filter 2 (Symmetric 3-partitions via ternary numbers) Similarly to Filter 1, this filter is also executed rapidly, but the number of 3-partitions formed is larger than those of Filter 1. The principle behind Filter 2 is basically the same as that behind Filter 1, but here all ternary numbers of length $n/3$ are generated instead. Hence, Filter 2 is also only executed if the order, n , of the

circulant G is divisible by three. The ternary number of length $n/3$ is extended to a ternary number of length n as follows. Suppose the original ternary number of length $n/3$ is denoted by a_1 and let the ternary number of length n formed from a_1 be $a = a_1 a_2 a_3$ where a_2 and a_3 are also ternary numbers of length $n/3$. The numbers a_2 and a_3 are formed from a_1 by replacing each 0 in a_1 with a 1 in a_2 and with a 2 in a_3 , each 1 in a_1 with a 2 in a_2 and with a 0 in a_3 , and each 2 in a_1 with a 0 in a_2 and with a 1 in a_3 . A 3-colouring of G is formed by colouring all the vertices according to the ternary number a . Again, only the colour class induced maximum degree of one colour class needs to be determined. If the colour class induced maximum degree is 3, the circulant may be disregarded as a candidate graph realizing the Δ -chromatic sequence 4444222222221; otherwise the next ternary number is generated until all ternary numbers of length $n/3$ have been generated.

The strategy employed in Filter 2 is illustrated in Figure 6.5(b) for any circulant of order 27 and for the ternary number $a_1 = 110112002$ (of length 9). The ternary number a formed from a_1 is thus 110112002|221220110|002001221.

Filter 3 (Symmetric 2-partitions via binary numbers) Filter 3 is similar to Filter 2, but in this case 2-partitions instead of 3-partitions are considered and therefore a binary number, a_1 , of length $n/2$ is generated to form a binary number, $a = a_1 a_2$, of length n . Filter 3 is therefore only executed if the order, n , of the circulant G is even. Furthermore, where the circulant G was disregarded when a 3-colouring with a colour class induced maximum degree of 3 was found in Filter 2, Filter 3 terminates and the program moves on to Filter 5 once a 2-colouring with a colour class induced maximum degree of 4 has been found, since $\chi_4^\Delta(G) = 2$ requires that G must have a 2-partitioning with partition induced maximum degree of 4. If no 2-colouring with a colour class induced maximum degree of 4 has been found during the execution of Filter 3, the program moves on to Filter 4 to search through more 2-partitionings for one with a colour class induced maximum degree of 4.

Figure 6.6 illustrates the strategy in Filter 3 for any circulant of order 14 and for the binary number $a_1 = 1101001$ (of length 7). The binary number a formed from a_1 is 1101001|0010110.

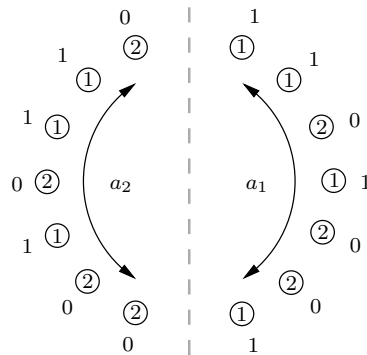


FIGURE 6.6: A graphical representation of the Filter 3 strategy in search of a 2-partition of a circulant for which at least one of the partition induced maximum degrees is 4.

As already mentioned, although Filters 1 to 3 may be executed quickly because of their symmetries, these filters may only be executed if $2|n$ in Filter 3 and $3|n$ in Filters 1 and 2, respectively. If either a circulant has not yet been disregarded by Filters 1 or 2, or if $3 \nmid n$ or $2 \nmid n$ (in which case Filters 1 and 2 or Filter 3, respectively, have not been executed) asymmetric partitions need to be considered as well. This is done in Filters 4 and 5. Since Filter 5 is the most time expensive part of the program, this filter is considered last, so that Filter 5 is executed on as few graphs as possible. However, Filter 5 is discussed before Filter 4, since Filter 4 is a simplified version of Filter 5.

Filter 5 (Asymmetric 3-partitions) The main idea in Filter 5 is to determine a partition with a large cardinality for which the partition induced maximum degree is at most 3, and then to determine whether the remaining vertices may be partitioned into two partitions such that both partition

induced maximum degrees are at most 3. The strategy in Filter 5 is given in pseudo-code as Algorithm 12, followed by a detailed description of the working of the filter.

Algorithm 12 Filter 5 algorithm

Input: A 12-regular basic circulant G .

Output: A 3-partition of a circulant for which each partition induced maximum degree is at most 3, if such a partition exists.

```

1: for all  $p_1 = \lceil n/2 \rceil, \dots, \lceil n/3 \rceil$  do
2:   if FindFirst( $G, p_1, \text{part1}$ ) then
3:     repeat
4:       for all  $p_2 = p_1, \dots, \lceil p_1/2 \rceil$  do
5:         if FindFirst( $G - \text{part1}, p_2, \text{part2}$ ) and  $p_1 + p_2 \leq n$  then
6:           repeat
7:             if  $\Delta(G - \text{part1} - \text{part2}) \leq 3$  then
8:               Stop, Return  $\text{part1}, \text{part2}, G - \text{part1} - \text{part2}$ 
9:             end if
10:          until NOT NextPart( $G, \text{part2}$ )
11:        end if
12:      end for
13:    until NOT NextPart( $G, \text{part1}$ )
14:  end if
15: end for
16: Print No such partition

```

The parameter p_1 in Step 1 of Algorithm 12 is the cardinality of the first partition, denoted by **part1** in the algorithm. The interval for p_1 is chosen such that partition **part1** has the largest cardinality of the three partitions. Hence, the smallest possible value that p_1 may have is $\lceil n/3 \rceil$. The largest cardinality of partition **part1** was chosen to be $\lceil n/2 \rceil$ in order to limit the number of iterations in Algorithm 12. If, however, any candidate circulants still remain at the end of the program, Filter 5 may be repeated on these circulants only, for values of p_1 larger than $\lceil n/2 \rceil$. The function FindFirst is a boolean function that returns the value **true** if a partition of cardinality p_1 for which the partition induced maximum degree is at most 3 has been found. As a consequence the lexicographic minimum partition, if any partition exists, is determined by the function FindFirst and is saved as the variable **part1**.

After one partition (**part1**) with partition induced maximum degree at most 3 has been found, a second partition for which the partition induced maximum degree is at most 3, is sought. This second partition, if it exists, is denoted by **part2** in the algorithm and has cardinality p_2 . Again the interval for the cardinality of the second partition is chosen such that its cardinality is larger than the third and final partition. Therefore, the largest possible cardinality of partition **part2** is p_1 , in which case the first and second partitions have the same cardinality, while the smallest possible cardinality of partition **part2** is $p_1/2$, in which case the second and third partitions have the same cardinality. For the specific first partition, **part1**, determined in Step 2 of Algorithm 12, the loop spanning Steps 4 to 10 considers all possible second and third partitions of $V(G)$. This is done by first determining in Step 5 (by means of the boolean function FindFirst) whether a partition of cardinality p_2 for which the partition induced maximum degree is at most 3, exists among the remaining vertices. If such a partition exists, this partition is stored as the variable **part2**. Next, the partition induced maximum degree of the third and final partition formed by the remaining vertices is determined. If the partition induced maximum degree of the third partition is at most 3, then Algorithm 12 terminates at Step 8, since a 3-partition for which each partition induced maximum degree is at most 3, has been found. On the other hand, if the partition induced maximum degree of the third partition is greater than 3, the next lexicographic minimum partition of cardinality p_2 for which the partition induced maximum degree is at most 3, is determined (by means of the boolean function NextPart) in Step 10 of Algorithm 12, if such a partition exists. As before, if such a partition exists, the partition induced maximum degree of the new third partition formed by the remaining vertices is determined and according to the value of this partition induced

maximum degree, the algorithm terminates or searches for a further second partition of cardinality p_2 for which the partition induced maximum degree is at most 3. This process continues until no more second partitions of cardinality p_2 for which the partition induced maximum degree is at most 3, can be found, in which case the cardinality of the second partition is incremented by one and the whole process is repeated. If still no 3-partition for which each partition induced maximum degree is at most 3, has been found, another first partition of cardinality p_1 is sought (by means of the boolean function `NextPart`) in Step 13 and the above process is repeated. Finally, the cardinality of the first partition is increased and the search for a 3-partition for which each partition induced maximum degree is at most 3, continues.

Filter 4 (Asymmetric 2-partitions) The working of Filter 4 is similar to that of Filter 5, where a 2-partition for which each partition induced maximum degree is at most 4 is sought, instead of a 3-partition for which each partition induced maximum degree is at most 3, as in Filter 5. The strategy in Filter 4 is given in pseudo-code as Algorithm 13, followed by a description of the working of the filter in terms of descriptions given in the discussion of Filter 5. In this case all 2-partitions

Algorithm 13 Filter 4 algorithm

Input: A 12-regular basic circulant G .

Output: A 2-partition of a circulant such that each partition induced maximum degree is at most 4, if such a partition exists.

```

1: for all  $p_1 = n, \dots, \lceil n/2 \rceil$  do
2:   if FindFirst( $G, p_1, \text{part1}$ ) then
3:     repeat
4:       if  $\Delta(G\text{-part1}) \leq 4$  then
5:         Stop, Return  $\text{part1}, G\text{-part1}$ 
6:       end if
7:     until NOT NextPart( $G, \text{part1}$ )
8:   end if
9: end for
10: Print No such partition

```

are considered so that the cardinality, p_1 , of the larger first partition ranges from n down to $\lceil n/2 \rceil$. Similarly to Filter 5, if a partition of cardinality p_1 for which the partition induced maximum degree is at most 4 was found by means of the boolean function `FirstPart`, this partition is stored as the variable `part1`. For the specific first partition, `part1`, determined in Step 2 of Algorithm 13, the second partition of $V(G)$ formed by the remaining vertices is determined. As before, if the partition induced maximum degree of the second partition is at most 4, then Algorithm 13 terminates at Step 5, since a 2-partition for which each partition induced maximum degree is at most 4 has been found. On the other hand, if the partition induced maximum degree of the second partition is greater than 4, then it needs to be determined whether another first partition of cardinality p_1 for which the partition induced maximum degree is at most 4, exists. This is done by means of the boolean function `NextPart` in Step 7 of Algorithm 13. The process of determining and evaluating the partition induced maximum degree of the second partition corresponding to each first partition continues until either a 2-partition for which each partition induced maximum degree is at most 4 has been found in which case the algorithm terminates, or all 2-partitions have been determined, in which case such a 2-partition does not exist.

During the course of this study the author attempted to construct a symmetric 12-regular graph structure realizing the Δ -chromatic sequence 4444222222221 (in order to easily adapt this structure to graphs realizing other Δ -chromatic basic sequences than 4444222222221). Therefore, circulants of order $12n$ were considered first, and since 24 is the smallest order of $12n$ larger than 17, these circulants were considered first. All 462 12-regular basic circulants of order 24 were generated and processed by the first program in MATHEMATICA. The output file of this process is given on the CD accompanying this dissertation (see Appendix G). Of the 462 12-regular basic circulants, a total of 118 circulants have a chromatic number of 4 and for 91 circulants the time-out condition was satisfied. The list of 118 circulants of order 24 with a chromatic number of 4 was the first list processed by the second (JAVA)

program. A 3-partition such that each partition set induced maximum degree is at most 3 was found for 53 of these 118 circulants by means of Filter 1, while a 3-partition for a further 63 circulants was found by means of Filter 2. The remaining 2 circulants, $C_{24}\langle 1, 2, 6, 7, 9, 10 \rangle$ and $C_{24}\langle 2, 3, 5, 6, 10, 11 \rangle$,

No	Circulant	Filter	No	Circulant	Filter	No	Circulant	Filter
1	$C_{24}\langle 1, 2, 3, 5, 6, 7 \rangle$	1	41	$C_{24}\langle 1, 3, 4, 5, 7, 11 \rangle$	2	80	$C_{24}\langle 2, 3, 5, 6, 9, 10 \rangle$	2
2	$C_{24}\langle 1, 2, 3, 5, 6, 9 \rangle$	2	42	$C_{24}\langle 1, 3, 4, 5, 9, 11 \rangle$	2	81	$C_{24}\langle 2, 3, 5, 6, 9, 11 \rangle$	2
3	$C_{24}\langle 1, 2, 3, 5, 6, 10 \rangle$	2	43	$C_{24}\langle 1, 3, 4, 6, 8, 9 \rangle$	1	82	$C_{24}\langle 2, 3, 5, 6, 10, 11 \rangle$	5
4	$C_{24}\langle 1, 2, 3, 5, 6, 11 \rangle$	2	44	$C_{24}\langle 1, 3, 4, 6, 8, 11 \rangle$	2	83	$C_{24}\langle 2, 3, 5, 7, 9, 10 \rangle$	2
5	$C_{24}\langle 1, 2, 3, 5, 7, 9 \rangle$	2	45	$C_{24}\langle 1, 3, 4, 6, 9, 11 \rangle$	2	84	$C_{24}\langle 2, 3, 5, 7, 9, 11 \rangle$	1
6	$C_{24}\langle 1, 2, 3, 5, 7, 10 \rangle$	2	46	$C_{24}\langle 1, 3, 4, 7, 9, 11 \rangle$	2	85	$C_{24}\langle 2, 3, 5, 7, 10, 11 \rangle$	2
7	$C_{24}\langle 1, 2, 3, 5, 7, 11 \rangle$	2	47	$C_{24}\langle 1, 3, 4, 8, 9, 11 \rangle$	2	86	$C_{24}\langle 2, 3, 5, 9, 10, 11 \rangle$	2
8	$C_{24}\langle 1, 2, 3, 5, 9, 10 \rangle$	2	48	$C_{24}\langle 1, 3, 5, 6, 7, 9 \rangle$	1	87	$C_{24}\langle 2, 3, 6, 7, 8, 11 \rangle$	1
9	$C_{24}\langle 1, 2, 3, 5, 9, 11 \rangle$	2	49	$C_{24}\langle 1, 3, 5, 6, 7, 10 \rangle$	1	88	$C_{24}\langle 2, 3, 6, 7, 9, 10 \rangle$	1
10	$C_{24}\langle 1, 2, 3, 5, 10, 11 \rangle$	2	50	$C_{24}\langle 1, 3, 5, 6, 7, 11 \rangle$	1	89	$C_{24}\langle 2, 3, 6, 7, 9, 11 \rangle$	2
11	$C_{24}\langle 1, 2, 3, 6, 7, 9 \rangle$	2	51	$C_{24}\langle 1, 3, 5, 6, 8, 10 \rangle$	1	90	$C_{24}\langle 2, 3, 6, 7, 10, 11 \rangle$	2
12	$C_{24}\langle 1, 2, 3, 6, 7, 10 \rangle$	2	52	$C_{24}\langle 1, 3, 5, 6, 9, 10 \rangle$	2	91	$C_{24}\langle 2, 3, 6, 9, 10, 11 \rangle$	2
13	$C_{24}\langle 1, 2, 3, 6, 7, 11 \rangle$	2	53	$C_{24}\langle 1, 3, 5, 6, 9, 11 \rangle$	2	92	$C_{24}\langle 2, 3, 7, 9, 10, 11 \rangle$	2
14	$C_{24}\langle 1, 2, 3, 6, 9, 10 \rangle$	2	54	$C_{24}\langle 1, 3, 5, 6, 10, 11 \rangle$	1	93	$C_{24}\langle 2, 4, 5, 6, 10, 11 \rangle$	2
15	$C_{24}\langle 1, 2, 3, 6, 9, 11 \rangle$	2	55	$C_{24}\langle 1, 3, 5, 7, 9, 10 \rangle$	2	94	$C_{24}\langle 2, 4, 6, 7, 9, 10 \rangle$	1
16	$C_{24}\langle 1, 2, 3, 6, 10, 11 \rangle$	2	56	$C_{24}\langle 1, 3, 5, 7, 10, 11 \rangle$	1	95	$C_{24}\langle 2, 5, 6, 7, 9, 10 \rangle$	1
17	$C_{24}\langle 1, 2, 3, 7, 9, 10 \rangle$	2	57	$C_{24}\langle 1, 3, 5, 9, 10, 11 \rangle$	1	96	$C_{24}\langle 2, 5, 6, 7, 9, 11 \rangle$	1
18	$C_{24}\langle 1, 2, 3, 7, 9, 11 \rangle$	2	58	$C_{24}\langle 1, 3, 6, 7, 9, 10 \rangle$	2	97	$C_{24}\langle 2, 5, 6, 7, 10, 11 \rangle$	2
19	$C_{24}\langle 1, 2, 3, 7, 10, 11 \rangle$	2	59	$C_{24}\langle 1, 3, 6, 7, 9, 11 \rangle$	2	98	$C_{24}\langle 2, 5, 6, 9, 10, 11 \rangle$	1
20	$C_{24}\langle 1, 2, 3, 9, 10, 11 \rangle$	2	60	$C_{24}\langle 1, 3, 6, 7, 10, 11 \rangle$	1	99	$C_{24}\langle 2, 5, 7, 9, 10, 11 \rangle$	1
21	$C_{24}\langle 1, 2, 4, 6, 7, 9 \rangle$	1	61	$C_{24}\langle 1, 3, 6, 8, 9, 11 \rangle$	2	100	$C_{24}\langle 2, 6, 7, 9, 10, 11 \rangle$	1
22	$C_{24}\langle 1, 2, 4, 6, 7, 10 \rangle$	2	62	$C_{24}\langle 1, 3, 6, 9, 10, 11 \rangle$	2	101	$C_{24}\langle 3, 4, 5, 6, 7, 8 \rangle$	1
23	$C_{24}\langle 1, 2, 4, 6, 9, 10 \rangle$	2	63	$C_{24}\langle 1, 3, 7, 9, 10, 11 \rangle$	1	102	$C_{24}\langle 3, 4, 5, 6, 7, 9 \rangle$	1
24	$C_{24}\langle 1, 2, 4, 7, 9, 10 \rangle$	2	64	$C_{24}\langle 1, 4, 5, 7, 9, 11 \rangle$	2	103	$C_{24}\langle 3, 4, 5, 6, 8, 9 \rangle$	1
25	$C_{24}\langle 1, 2, 5, 6, 7, 9 \rangle$	1	65	$C_{24}\langle 1, 4, 6, 7, 9, 10 \rangle$	1	104	$C_{24}\langle 3, 4, 5, 6, 10, 11 \rangle$	1
26	$C_{24}\langle 1, 2, 5, 6, 7, 10 \rangle$	2	66	$C_{24}\langle 1, 4, 6, 8, 9, 11 \rangle$	2	105	$C_{24}\langle 3, 4, 5, 7, 8, 9 \rangle$	1
27	$C_{24}\langle 1, 2, 5, 6, 7, 11 \rangle$	2	67	$C_{24}\langle 1, 5, 6, 7, 9, 10 \rangle$	1	106	$C_{24}\langle 3, 4, 5, 7, 9, 11 \rangle$	1
28	$C_{24}\langle 1, 2, 5, 6, 8, 9 \rangle$	1	68	$C_{24}\langle 1, 5, 6, 7, 9, 11 \rangle$	2	107	$C_{24}\langle 3, 4, 6, 7, 8, 9 \rangle$	1
29	$C_{24}\langle 1, 2, 5, 6, 9, 10 \rangle$	2	69	$C_{24}\langle 1, 5, 6, 7, 10, 11 \rangle$	1	108	$C_{24}\langle 3, 4, 6, 8, 9, 11 \rangle$	1
30	$C_{24}\langle 1, 2, 5, 6, 9, 11 \rangle$	2	70	$C_{24}\langle 1, 5, 6, 9, 10, 11 \rangle$	1	109	$C_{24}\langle 3, 5, 6, 7, 8, 9 \rangle$	1
31	$C_{24}\langle 1, 2, 5, 6, 10, 11 \rangle$	2	71	$C_{24}\langle 1, 5, 7, 9, 10, 11 \rangle$	1	110	$C_{24}\langle 3, 5, 6, 7, 9, 10 \rangle$	1
32	$C_{24}\langle 1, 2, 5, 7, 9, 10 \rangle$	2	72	$C_{24}\langle 1, 6, 7, 9, 10, 11 \rangle$	1	111	$C_{24}\langle 3, 5, 6, 7, 9, 11 \rangle$	1
33	$C_{24}\langle 1, 2, 5, 7, 9, 11 \rangle$	2	73	$C_{24}\langle 2, 3, 4, 5, 6, 10 \rangle$	1	112	$C_{24}\langle 3, 5, 6, 7, 10, 11 \rangle$	1
34	$C_{24}\langle 1, 2, 5, 9, 10, 11 \rangle$	2	74	$C_{24}\langle 2, 3, 4, 5, 6, 11 \rangle$	1	113	$C_{24}\langle 3, 5, 6, 9, 10, 11 \rangle$	1
35	$C_{24}\langle 1, 2, 6, 7, 9, 10 \rangle$	5	75	$C_{24}\langle 2, 3, 4, 5, 10, 11 \rangle$	2	114	$C_{24}\langle 3, 5, 7, 9, 10, 11 \rangle$	1
36	$C_{24}\langle 1, 2, 6, 7, 9, 11 \rangle$	2	76	$C_{24}\langle 2, 3, 4, 6, 10, 11 \rangle$	2	115	$C_{24}\langle 3, 6, 7, 9, 10, 11 \rangle$	1
37	$C_{24}\langle 1, 2, 6, 7, 10, 11 \rangle$	2	77	$C_{24}\langle 2, 3, 5, 6, 7, 9 \rangle$	1	116	$C_{24}\langle 4, 5, 6, 7, 8, 9 \rangle$	1
38	$C_{24}\langle 1, 2, 6, 9, 10, 11 \rangle$	2	78	$C_{24}\langle 2, 3, 5, 6, 7, 10 \rangle$	1	117	$C_{24}\langle 5, 6, 7, 9, 10, 11 \rangle$	1
39	$C_{24}\langle 1, 2, 7, 9, 10, 11 \rangle$	2	79	$C_{24}\langle 2, 3, 5, 6, 7, 11 \rangle$	1	118	$C_{24}\langle 6, 7, 8, 9, 10, 11 \rangle$	1
40	$C_{24}\langle 1, 3, 4, 5, 7, 9 \rangle$	1						

TABLE 6.6: Circulants of order 24 with a chromatic number of 4 disregarded as candidate graphs realizing the Δ -chromatic sequence 4444222222221. The Filter column indicates by means of which filter the specific circulant was disregarded.

were disregarded by means of Filter 5. The list of 118 circulants of order 24 with a chromatic number of 4 is given in Table 6.6 as well as during which filter each circulant was disregarded as a candidate graph realizing the Δ -chromatic sequence 444422222221. The list of 91 circulants of order 24 that satisfied the time-out condition was processed by the JAVA program next. For 84 of these 91 circulants a 3-partition such that each partition set induced maximum degree is at most 3 was found by means of Filter 1. The remaining 7 circulants were disregarded by means of Filter 2, so that none of the circulants of order 24 remained as a candidate graph realizing the Δ -chromatic basic sequence 444422222221. The output files of the JAVA program for the circulants of order 24 with a chromatic number of 4 and the ones which satisfied the time-out condition, are also given on the CD accompanying this dissertation (see Appendix G).

Circulants of other orders than 24 were considered in the same fasion. In some cases all 12-regular basic circulants of that order were generated and processed by the MATHEMATICA program, but in other cases (for relatively large orders) only a selection of the 12-regular basic circulants were used, since the total number of these circulants were very large. The results for these circulants are summarized in Table 6.7 and the particular output files of both programs are included on the CD accompanying this dissertation (see Appendix G). In the cases of circulants of orders 17–23 all 12-regular basic circulants of that order were generated and processed by the MATHEMATICA program, while in the other cases only a selection of circulants of that order was generated. As evident from columns three, four and nine in Table 6.7, all circulants of orders 17–23, 27, 30, 33, 36 and 48 considered were also disregarded as candidate graphs realizing the Δ -chromatic basic sequence 444422222221.

Order	Number of circulants							
	Generated	with $\chi = 4$	Time-out condition	Disregarded by means of				Total disregarded
				Filter 1	Filter 2	Filter 4	Filter 5	
17	28	0	20	–	–	20	0	20
18	28	0	12	12	0	0	0	12
19	84	0	31	–	–	31	0	31
20	84	7	14	–	–	7	14	21
21	210	3	96	73	13	13	0	99
22	210	5	44	–	–	39	10	49
23	462	11	170	–	–	181	0	181
27	792	30	494	352	163	9	0	524
30	3 003	148	1 158	1 103	200	3	0	1 306
33	3 003	241	1 529	1 078	674	18	0	1 770
36	3 862	459	1 465	1 083	841	0	0	1 924
48	7 851	2 156	3 786	1 849	4 093	0	0	5 942

TABLE 6.7: Circulants disregarded as candidate graphs realizing the Δ -chromatic sequence 444422222221. The first column lists the order of the specific group of circulants, while the second column lists the number of circulants of a particular order that were generated by the MATHEMATICA program. The number of circulants among those generated by the MATHEMATICA program with a chromatic number of four or that satisfy the time-out condition are listed in columns three and four respectively. Columns 5–8 list the number of circulants disregarded by means of Filters 1, 2, 4 or 5, while the final column lists the total number of circulants processed and disregarded by the JAVA program.

6.4.4 Other constructions

Other attempts towards finding a graph realizing the basic sequence 444422222221 were also made. For example, one may start with a number of copies of K_4 , therefore forcing χ_0^Δ to be 4. Then edges may be added such that the resulting graph is r -regular until $r = 12$, in which case the value χ_{12}^Δ is still equal to 1. These edges should of course be added in such a way as to keep χ_d^Δ equal to 2 for all $4 \leq d \leq 11$. As soon as a candidate graph was obtained, Filters 4 and 5 from the circulant program described in §6.4.3 were used to determine whether this graph indeed has basic sequence 444422222221. Detailed descriptions of these graphs are rather technical and are therefore not included here. Unfortunately, all these attempts also failed as a means to classify basic sequence 444422222221 as Δ -chromatic.

6.5 Chapter Summary

The aim of this chapter was to investigate whether the necessary conditions in Theorem 3.13 for a sequence of positive integers to be the Δ -chromatic sequence of some graph G , are also sufficient conditions. In order to reduce the number of different sequences to be considered, the notion of basic sequences that satisfy the conditions in Theorem 3.13 was introduced in §6.1. Basic sequences that may be classified as Δ -chromatic sequences were discussed in §6.2. Unfortunately, not all basic sequences could be classified as Δ -chromatic sequences and §6.3 was devoted to the characteristics of the smallest basic sequence that could not be classified as Δ -chromatic, namely 444422222221. Various graph structure classes, studied mainly in an attempt to classify the basic sequence 444422222221 as Δ -chromatic, as well as why these graph structure classes failed, were reported in §6.4. Structure classes included in this section were the product K_n^r (§6.4.1), the product $K_{4 \times 3} \times K_4$ (§6.4.2), circulants (§6.4.3) and a few *ad hoc* graphs (§6.4.4).

Chapter 7

Conclusion

“Now this is not the end. It is not even the beginning of the end. But perhaps, the end of the beginning.”

Sir Winston Churchill (1874–1965)
Speech in November 1942

“As long as a branch of science offers an abundance of problems, so long is it alive; a lack of problems foreshadows extinction or the cessation of independent development. Just as every human undertaking pursues certain objects, so also mathematical research requires its problems. It is by the solution of problems that the investigator tests the temper of his steel; he finds new methods and new outlooks, and gains a wider and freer horizon.”

David Hilbert (1862–1943)
Mathematical problems, International Congress of Mathematicians, Paris, 1900

The purpose of this chapter is twofold. In the first place, the work presented in this dissertation is summarised briefly in §7.1. Secondly, possible improvements to the work presented in this dissertation, as well as some possible future avenues of investigation, are outlined in §7.2.

7.1 Dissertation Summary

This dissertation opened in Chapter 1 with a brief summary of the strange history of the four-colour problem filled with attempted proofs, the publication of Kempe’s incorrect proof and the first ever computer aided mathematical proof, which resolved the four-colour conjecture at last. In the next section of Chapter 1 it was demonstrated how graph colouring is related to the four-colour problem, which in turn may be viewed as the origin of graph colouring problems. Different types of graph colouring, as well as graph colouring application possibilities, were also presented.

The necessary basic terminology from graph theory and complexity theory were introduced in Chapter 2. This was followed by the formal graph theoretic definition of the notion of graph colouring, as well as some well-known graph colouring algorithms found in the literature. The remainder of Chapter 2 was divided into two key topics essential to the work contained in this dissertation. First, a review was given of the notion of, and the requirements for, generalizations of the classical process of graph colouring, thus achieving objective I(a) of §1.5. Finally, the notion of a chromatic sequence was introduced.

The notion of maximum degree colourings was defined formally in Chapter 3, followed by some basic results in the literature on the maximum degree chromatic number (thereby achieving objective I(b) in §1.5). The known necessary conditions (Theorem 3.13) from the literature for a sequence of positive integers to be the Δ -chromatic sequence of some graph G were also reviewed, as well as a few other results on Δ -chromatic sequences from the literature. Chapter 3 closed with the introduction of the

new inverted strategy towards determining maximum degree colourings where the maximum degree of the colour induced subgraphs for a given number of colours are minimized. A relationship between the parameters $\chi_d^\Delta(G)$ and $D_x^\Delta(G)$ was also established, as well as the significance of the new parameter $D_x^\Delta(G)$ when documenting the Δ -chromatic sequence of a graph G . Objective II of §1.5 was therefore addressed in certain parts of Chapter 3.

Chapter 4 addressed objectives III and IV of §1.5. First, the exact values of the maximum degree chromatic number and of the inverted parameter D_x^Δ were determined for graphs from certain simple structure classes, and in particular bipartite graphs, cycles, wheels, complete graphs and products of paths and cycles. In the case of products of complete graphs, the maximum degree chromatic number or the inverted parameter D_x^Δ was obtained for certain values only of the parameters d and x respectively. For other values of the parameters d and x , it was found that it was still exceedingly technical even to determine reasonably good bounds on the value of the maximum degree chromatic number or the inverted parameter D_x^Δ , and bounds were determined for certain cases only. Analogous to the classical chromatic number, it was found that determination of the maximum degree chromatic number of a circulant is also very technical in nature. The maximum degree chromatic number and the inverted parameter D_x^Δ of an elementary circulant were determined entirely. In the case of a circulant with connection set of cardinality two, the maximum degree chromatic number was determined only partially. Only a few results were established for other circulants with a higher maximum degree than four. Finally, in the case of complete balanced multipartite graphs the maximum degree chromatic number could not be obtained exactly, but the specific structure of complete balanced multipartite graphs was exploited in order to determine a good algorithmic upper bound on the maximum degree chromatic number of a complete balanced multipartite graph.

Four new $\Delta(d, x)$ -colouring algorithms, derived from well-known classical graph colouring algorithms found in the literature, were presented in Chapter 5. The first two algorithms, namely the colour degree heuristic and the tabu search $\Delta(d, x)$ -colouring heuristic, are (meta)heuristics that only produce an upper bound on the maximum degree chromatic number of a graph (thereby achieving objective V(b) in §1.5), where the tabu search $\Delta(d, x)$ -colouring heuristic in general achieves better upper bounds than the colour degree heuristic. The colour degree heuristic, which is a greedy algorithm, is highly time efficient, while the tabu search $\Delta(d, x)$ -colouring heuristic, which is a local search technique, also becomes time efficient for graphs of order roughly greater than 35. The subsequent two exact algorithms, namely the irredundant χ_d^Δ -colouring algorithm and the critical χ_d^Δ -colouring algorithm, give the exact value of the maximum degree chromatic number, but utilise more basic operations to arrive at these values (thereby achieving objective V(a) of §1.5). It was found that the irredundant χ_d^Δ -colouring algorithm is far more time efficient than the critical χ_d^Δ -colouring algorithm. Finally, it was found that for small graphs (of order at most 35) the irredundant χ_d^Δ -colouring algorithm is preferred to the other three algorithms, while for larger order graphs, the tabu search $\Delta(d, x)$ -colouring heuristic is the preferred choice.

In Chapter 6 the necessary conditions in Theorem 3.13 for a sequence of positive integers to be the Δ -chromatic sequence of some graph G were investigated in order to determine whether these conditions are also sufficient, thereby achieving objective VI of §1.5. First, all integral sequences up to a certain length satisfying the aforementioned necessary conditions were generated, after which a representative subset of these integral sequences, called basic sequences, which suffice in the study of maximum degree chromatic sequences, was defined. Basic sequences that could be classified as Δ -chromatic sequences were also discussed. Unfortunately, not all basic sequences could be classified as Δ -chromatic and the remainder of Chapter 6 was devoted to attempts to classify other basic sequences as Δ -chromatic with the emphasis on the basic sequence 4444222222221. This sequence is the first (smallest) basic sequence that can as yet not be classified as Δ -chromatic and some characteristics of this sequence were studied as well. In this regard, it was shown that a graph G with Δ -chromatic sequence 4444222222221 must be 4-partite, has a maximum degree of 12, must be of order at least 17, and may contain a clique of order at most 4. Certain restrictions on the number of edges were also given, and there must exist a 4-partition of $V(G)$ such that every partition induced maximum degree is at most 3, but there does not exist a 3-partition of $V(G)$ such that every partition induced maximum degree is at most 3.

Various graph structure classes were studied in an attempt to classify basic sequences as Δ -chromatic. It was first shown that the product K_n^r failed as a general graph structure realizing a Δ -chromatic basic sequence. In the case of complete balanced multipartite graphs, the graph $K_{4 \times 4}$ achieves the value 3 as

an upper bound on x_3 in its Δ -chromatic sequence, while the basic sequence 4 4 4 4 2 2 2 2 2 2 2 1 exhibits the value $x_3 = 4$. Although the product $K_{4 \times 3} \times K_4$ has $x_0 = 4$ and $x_{12} = 1$ in its Δ -chromatic sequence, as is the case with the basic sequence in question, it was shown that $K_{4 \times 3} \times K_4$ again has the value 3 as an upper bound on x_3 in its Δ -chromatic sequence, while the basic sequence 4 4 4 4 2 2 2 2 2 2 2 1 exhibits the value $x_3 = 4$. In the case of circulants, a brute force method was developed to search through all 3-partitions of a given circulant with a maximum degree of 12, in order to determine whether a 3-partition of $V(G)$ exists such that every partition induced maximum degree is at most 3. In all circulants with classical chromatic number 4 considered, such a 3-partition could be found.

7.2 Further Work

During the course of research for this dissertation, the author came across a number of open problems. The first open problem concerns proper colourings.

Question 7.1 *Is it possible to determine analytically a better upper bound on the complexity of Brown's modified colouring algorithm (Algorithm 4) than the factorial upper bound in §2.3?* ■

In Theorem 3.6 (and included in the bounds summarised in (3.2.1)) the lower bound $p(G)/\beta_d(G)$ on $\chi_d^\Delta(G)$ for a graph G , was given. In terms of the related inverted parameter, $D_x^\Delta(G)$, one may ask the following question.

Question 7.2 *For the lower bound $p(G)/\beta_d(G)$ on $\chi_d^\Delta(G)$ for a graph G , does there exist a related lower bound on $D_x^\Delta(G)$?* ■

Similarly to the notion of a critical $\Delta(d, x)$ -chromatic graph, a D_x^Δ -**critical graph** may be defined as a graph G for which $D_x^\Delta(H) < D_x^\Delta(G)$ for some fixed value of x and every proper subgraph H of G . The characterization of critical $\Delta(d, x)$ -chromatic graphs of smallest order for all $x \in \mathbb{N} \setminus \{1, 2\}$ and $d \in \mathbb{N}_0$ given in Theorem 3.10, leads to the question below.

Question 7.3 *Is it possible to characterise D_x^Δ -critical graphs of smallest order?* ■

The maximum degree chromatic number for some structure classes was investigated in Chapter 4 and for some of these classes possibilities for future work have already opened up. For example, in §4.4 $\chi_1^\Delta(K_m \times K_n)$ was determined for $n \geq 2m - 1$ only, due to the technical nature of determining $\chi_d^\Delta(K_m \times K_n)$ for $d \in \mathbb{N}_0$. The case $n \leq 2m$ thus remains open. Another possibility for future work is to determine $\chi_d^\Delta(K_m \times K_n)$ for other values of $d \in \{2, \dots, m+n-3\}$. The following question summarises the two open problems regarding the maximum degree chromatic number for the cartesian product of two complete graphs described above.

Question 7.4 *Is it possible to determine*

- (a) $\chi_1^\Delta(K_m \times K_n)$ for $n \leq 2m$, and
- (b) $\chi_d^\Delta(K_m \times K_n)$ for some values of $2 \leq d \leq m+n-3$?

Instead of attempting to answer Question 7.4 due to the technical detail with which $\chi_1^\Delta(K_m \times K_n)$ was determined for $n \geq 2m - 1$, the following question may be explored.

Question 7.5 *Is it possible to determine $D_x^\Delta(K_m \times K_n)$ for certain or all values of $x \in \mathbb{N}$?* ■

If either Question 7.4 or Question 7.5 may be answered in the affirmative, the remaining question may easily be answered in the affirmative using Proposition 3.3. The cartesian products studied in §4.4 included $P_m \times P_n$, $P_m \times C_n$, $C_m \times C_n$ and $K_m \times K_n$. Two obvious structure classes to be considered next are $P_m \times K_n$ and $C_m \times K_n$ as stated below.

Question 7.6

- (a) Is $\chi_d^\Delta(P_m \times K_n) = \chi_d^\Delta(K_n)$ for all values of $d \in \mathbb{N}_0$?
- (b) Is $\chi_d^\Delta(C_m \times K_n) = \chi_d^\Delta(K_n)$ for all values of $d \in \mathbb{N}_0$? ■

In the case of circulants, a possibility for future work is to complete the characterization of the maximum degree chromatic number of a connected composite circulant, $C_n\langle i_1, i_2 \rangle$, of order n with connection set $\{i_1, i_2\}$. Another avenue of investigation may be to consider the maximum degree chromatic number of a connected singular composite circulant with a connection set of cardinality three.

As stated in §4.6.1 it is believed that Algorithm $\bar{\alpha}(x, k)$ produces good upper bounds, but which are not optimal in general as illustrated in Example 4.3. The following question is therefore posed.

Question 7.7 *Is it possible to determine exactly when the rare cases, as illustrated in Example 4.3, will occur during the course of execution of Algorithm $\bar{\alpha}(x, k)$?* ■

Finally, in terms of the maximum degree chromatic number of a graph from various structure classes, the following two questions regarding multipartite graphs arise.

Question 7.8 *May the strategy used in §4.6 to determine an upper bound on the maximum degree chromatic number of a complete balanced multipartite graph, be adopted to obtain an upper bound on the maximum degree chromatic number of a complete multipartite graph with partite sets of unequal cardinalities?* ■

Question 7.9 *Is it possible to exploit the partiteness of a multipartite graph to determine either the exact value of, or an upper bound on the maximum degree chromatic number of a not necessarily complete multipartite graph?* ■

The $\Delta(d, x)$ -colouring algorithms in Chapter 5 are basic algorithms for determining either an upper bound on or the exact value of the maximum degree chromatic number of a general graph in the sense that all four algorithms are merely adaptations from known proper graph colouring algorithms. More research may be done on improvements to these algorithms as stated in the following questions.

Question 7.10 *May the colour degree heuristic discussed in §5.1.1, be improved by not only considering the colour degree of each uncoloured vertex when the next vertex, v , to be coloured is selected, but by considering the colour class induced degrees of the coloured vertices in $N(v)$ as well?* ■

Question 7.11 *Is it possible to improve the working of the colour degree heuristic discussed in §5.1.1 by incorporating a colour interchange approach into the algorithm?* ■

It was found in §5.1.3 that for a specific set of tabu parameter values and for some graphs of order close to the lower bound for which another set of tabu parameter values was chosen, the tabu search $\Delta(d, x)$ -colouring heuristic obtained higher upper bounds on the $\Delta(d)$ -chromatic number for small values of d than did the colour degree heuristic. Also, for some of these graphs, the tabu search $\Delta(d, x)$ -colouring heuristic found a valid $\Delta(d, 2)$ -colouring for larger values of d than did the colour degree heuristic. In this regard, the following question is posed.

Question 7.12 *Is it possible to obtain better upper bounds if certain values for the tabu parameters are selected for the largest part of the algorithm, but if the tabu search $\Delta(d, x)$ -colouring heuristic is adapted to use a different set of tabu parameters (with possibly a larger value for \maxit) for small values of d , as well as when the number of colours become small?* ■

For some graphs, the irredundant χ_d^Δ -colouring algorithm (Algorithm 10) discussed in §5.2.1 has to backtrack all the way to one of the first clique vertices coloured. This leads to the following question.

Question 7.13 *If $d \geq 1$, is it possible to devise a method by which the initial clique in Steps 9 and 10 of the irredundant χ_d^Δ -colouring algorithm may be coloured in order to reduce backtracking as far back as one of the first clique vertices coloured?* ■

In Table 5.34 the ranking of the four $\Delta(d, x)$ -colouring algorithms according to their performance on the test graphs listed in Appendix F was given. These rankings may need to be re-evaluated if the following question may be answered in the affirmative.

Question 7.14 *Is there a graph order for which the critical χ_d^Δ -colouring algorithm becomes more time efficient than the irredundant χ_d^Δ -colouring algorithm?* ■

In §1.4 a scheduling problem where some threshold of conflict may be tolerated was introduced as an application to maximum degree colourings. Clearly, in such an application a balanced $\Delta(d, x)$ -colouring, where the colour classes have more or less the same cardinality, will be preferable to a $\Delta(d, x)$ -colouring that only attempts to find a colouring in the minimum number of colours. Such a requirement may easily be introduced in the colour degree heuristic by selecting at Step 13 of Algorithm 8 the colour that has been used the smallest number of times thus far, rather than the smallest possible colour. This change leads to the following two questions.

Question 7.15 *Will the performance in terms of the quality of the upper bound on the maximum degree chromatic number of a general graph obtained by the colour degree heuristic with a balanced $\Delta(d, x)$ -colouring be much worse than the performance of the original colour degree heuristic?* ■

Question 7.16 *How may the tabu search $\Delta(d, x)$ -colouring heuristic, given in Algorithm 9, also be adapted to obtain a balanced $\Delta(d, x)$ -colouring?* ■

During the search for a graph with Δ -chromatic sequence 4 4 4 4 2 2 2 2 2 2 2 1, all promising graphs failed at the term where the subsequence of 4's are followed by the value 2. This led the author to believe that the repetition of four 4's in a Δ -chromatic sequence followed directly by the value 2, without having the value 3 in the sequence first, is not possible. If the following question may be answered in the negative, it will provide valuable information in terms of the sufficiency of the necessary conditions in Theorem 3.13.

Question 7.17 *In the Δ -chromatic sequence (x_i) of a graph, can a relatively large number of repetitions of the same value, i.e. $x_i = x_{i+1} = \dots = x_{i+m} = k$ for some $m \in \mathbb{N}$) be followed by a jump in the integer value to the next position in the sequence (i.e. $x_{i+m+1} = k - \ell$ with ℓ relatively large and related to m)?* ■

When circulants were investigated as a possible means of classifying the basic sequence 4 4 4 4 2 2 2 2 2 2 2 1 as Δ -chromatic, the ease with which a 3-partition of the vertex set of larger order circulants (around 100) such that every partition induced maximum degree is at most 3 could be found by hand, led to the following question.

Question 7.18 *Is there a lower bound (as well as an upper bound) on the density of a graph with Δ -chromatic sequence 4 4 4 4 2 2 2 2 2 2 2 1?* ■

If Question 7.18 may be answered in the affirmative, it will limit the number of graph structures to be considered in the search for a graph with Δ -chromatic sequence 4 4 4 4 2 2 2 2 2 2 2 1.

Yet another future approach towards determining whether the necessary conditions in Theorem 3.13 are also sufficient is to consider another unclassified basic sequence instead of concentrating on the basic sequence 4 4 4 4 2 2 2 2 2 2 2 1, and attempt to classify this sequence as Δ -chromatic. More specifically, the following question is posed.

Question 7.19 *Is it possible to classify some other as yet unclassified basic sequence than 4 4 4 4 2 2 2 2 2 2 2 1 as Δ -chromatic?* ■

Filter 5 in the strategy used in §6.4.3 to disregard circulants possessing a 3-partition of the vertex set such that every partition induced maximum degree is at most 3, as a possible graph with Δ -chromatic sequence 444422222221, may be adapted to search for other sized partitions. This adapted filter may then be used to assist in the process of answering Question 7.19. If Question 7.19 may be answered in the affirmative, the corresponding graph structure may give some ideas in terms of a general graph structure realizing Δ -chromatic basic sequences. If the sequence in Question 7.19 may be classified as *not* Δ -chromatic, then the following question may be answered indirectly in the affirmative.

Question 7.20 *Is there another condition/other conditions which, together with the necessary conditions in Theorem 3.13, may form sufficient conditions for an integral sequence to be the Δ -chromatic sequence of some graph?* ■

The sequence of the related inversion numbers $\{D_x^\Delta(\bullet)\}_{x=1,2,3,\dots}$ of a graph G was briefly mentioned in §4.6.1. One avenue of possible future work on this sequence is stated below.

Question 7.21 *Are there necessary conditions for a sequence of positive integers to be the sequence of the related inversion numbers $\{D_x^\Delta(\bullet)\}_{x=1,2,3,\dots}$ of some graph G ?* ■

More research on the topic stated in Question 7.21 may provide some insight into the problem of determining sufficient conditions for a sequence of positive integers to be the Δ -chromatic sequence of some graph.

Although the focus in this dissertation on maximum degree colourings was from a purely mathematical point of view, an obvious final question, especially in the context of South Africa as a developing country, is stated below.

Question 7.22 *Besides the application to resource allocation where some threshold of conflict may be tolerated, is it possible to apply maximum degree colourings to any unsolved application problems?* ■

References

- [1] ALBERTSON, MO, JAMISON, RE, HEDETNIEMI, ST & LOCKE, SC, *The subchromatic number of a graph*, Discrete Mathematics, **74**, pp. 33–49, 1989.
- [2] AMERICAN MATHEMATICAL SOCIETY, *Fermat's last theorem*, [Online], [cited 2003, May 14], Available from: <http://www.ams.org/new-in-math/fermat.html>
- [3] ANDREWS, JA & JACOBSON, MS, *On a generalization of chromatic number*, Congressus Numerantium, **47**, pp. 33–48, 1985.
- [4] ANDREWS, JA & JACOBSON, MS, *On a generalization of chromatic number and two kinds of Ramsey numbers*, Ars Combinatoria, **23**, pp. 97–102, 1987.
- [5] APPEL, K, HAKEN, W & KOCH, J, *Every planar map is four-colourable*, Illinois Journal of Mathematics, **21**, pp. 429–567, 1977.
- [6] ARCHDEACON, D, *A note on defective colourings of graphs in surfaces*, Journal of Graph Theory, **11**, No 4, pp. 517–519, 1987.
- [7] BENADÉ, G, BROERE, I, JONCK, B & FRICK, M, *Uniquely $(m, k)^\tau$ -colourable graphs and k - τ -saturated graphs*, Discrete Mathematics, **162**, pp. 13–22, 1996.
- [8] BERGE, C, *Graphs and hypergraphs*, North-Holland Publishers, Amsterdam, 1973.
- [9] BOLLOBÁS, B, *Modern graph theory*, Graduate Texts in Mathematics Series, Springer-Verlag, New York, 1998.
- [10] BOLLOBÁS, B & WEST, DB, *A note on generalized chromatic number and generalized girth*, Discrete Mathematics, **213**, pp. 29–34, 2000.
- [11] BLACK, PE, *Dictionary of algorithms and data structures*, U.S. National Institute of Standards and Technology [Online], [cited 2007, January 23], Available from: <http://www.nist.gov/dads/>, last updated 2006, December 5.
- [12] BRADBURY, J, *Four-colour problem*, [Online], [cited 2002, October 15], Available from: <http://www.cs.queensu.ca/~bradbury>
- [13] BRELAZ, D, *New methods to colour the vertices of a graph*, Communications of the Association for Computing Machinery, **22**, pp. 251–256, 1979.
- [14] BROERE, I & FRICK, M, *A characterization of the sequence of generalized chromatic numbers of a graph*, Graph Theory, Combinatorics and Applications: Proceedings of the Sixth Quadrennial International Conference on the Theory and Applications of Graphs, edited by Y Alavi *et al.*, John Wiley and Sons, New York, pp. 179–186, 1991.
- [15] BROERE, I & FRICK, M, *On the order of colour critical graphs*, Proceedings of the Sixteenth Southeastern International Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium, **47**, pp. 125–130, 1985.
- [16] BROERE, I & FRICK, M, *On the order of uniquely (k, m) -colourable graphs*, Discrete Mathematics, **82**, pp. 225–232, 1990.

- [17] BROERE, I & FRICK, M, *Two results on generalized chromatic numbers*, Quaestiones Mathematicae, **13**, pp. 183–190, 1990.
- [18] BROOKS, RL, *On colouring the nodes of a network*, Proceedings of the Cambridge Philosophical Society, **37**, pp. 194–197, 1941.
- [19] BROWN, JI, *The complexity of generalized graph colourings*, Discrete Applied Mathematics, **69**, pp. 257–270, 1996.
- [20] BROWN, JI & CORNEIL, DG, *On generalized colourings*, Journal of Graph Theory, **11**, No 1, pp. 87–99, 1987.
- [21] BROWN, JR, *Chromatic scheduling and the chromatic number problem*, Management Science, **19**, No 4, pp. 456–463, 1972.
- [22] BURGER, AP & GROBLER, PJP, *Aspects of (m,k) -colourings*, Research Report **146/93(7)**, University of South Africa, Pretoria, 1993.
- [23] BURGER, AP & VAN VUUREN, JH, *On the placement of a number of strings in a collection of hats*, submitted.
- [24] BUTENKO, S, FESTA, P & PARDALOS, PM, *On the chromatic number of graphs*, Journal of Optimization Theory and Applications, **109**, No 1, pp. 51–67, 2001.
- [25] CAYLEY, A, *On the colouring of maps*, Proceedings of the Royal Geographical Society (New Series), **1**, pp. 259–261, 1879.
- [26] CAYLEY, A, *Open problem*, Proceedings of the London Mathematical Society, **9**, p. 148, 1878.
- [27] CHARTRAND, G, *Introductory graph theory*, Dover Publications, New York, 1977.
- [28] CHARTRAND, G, GELLER, DP & HEDETNIEMI, ST, *A generalization of the chromatic number*, Proceedings of the Cambridge Philosophical Society, **64**, 265–271, 1968.
- [29] CHARTRAND, G & OELLERMANN, O, *Applied and algorithmic graph theory*, McGraw-Hill, New York, 1993.
- [30] CHIARANDINI, M, *Bibliography on graph colouring*, [Online], [cited 2007, January 26], Available from: <http://www.imada.sdu.dk/~marco/gcp/>, last updated 2007, January 25.
- [31] CLARK, J & HOLTON, DA, *A first look at graph theory*, World Scientific Publishing Company, Singapore, 1991.
- [32] COOK, SA, *The complexity of theorem proving procedures*, Third Annual ACM Symposium on Theory of Computing, New York, pp. 151–158, 1971.
- [33] COWEN, L, COWEN, R & WOODALL, D, *Defective colourings of graphs in surfaces: Partitions into subgraphs of bounded valence*, Journal of Graph Theory, **10**, pp. 187–195, 1986.
- [34] COWEN, L, GODDARD, W & JESURUM, CE, *Defective colouring revisited*, Journal of Graph Theory, **24**, pp. 205–219, 1997.
- [35] CRANDALL, R & POMERANCE, C, *Prime numbers: A computational perspective*, Springer-Verlag, New York, 2001.
- [36] DHARWADKER, A, *A new proof of the four-colour theorem*, [Online], [cited 2005, July 22], Available from: <http://www.geocities.com/dharwadker/main.html>
- [37] DIRAC, GA, *A property of 4-chromatic graphs and some remarks on critical graphs*, Journal of the London Mathematical Society, **27**, pp. 85–92, 1952.
- [38] EDWARDS, K & FARR, G, *On monochromatic component size for improper colourings*, Discrete Applied Mathematics, **148**, pp. 89–105, 2005.

- [39] ERDÖS, P & SIMONOVITS, M, *A limit theorem in graph theory*, *Studia Scientiarum Mathematicarum Hungarica*, **1**, pp. 51–57, 1966.
- [40] EVEN, S, *Graph algorithms*, Computer Science Press, Maryland, 1979.
- [41] FOLKMAN, JH, *Graphs with monochromatic complete subgraphs in every edge colouring*, *SIAM Journal of Applied Mathematics*, **18**, pp. 19–24, 1970.
- [42] FOULDS, LR, *Graph theory applications*, Springer–Verlag, New York, 1992.
- [43] FRICK, M, *A survey of (m,k) -colourings, quo vadis, graph theory?*, *Annals of Discrete Mathematics*, **55**, pp. 45–58, 1993.
- [44] FRICK, M, *Generalised colourings of graphs*, PhD–dissertation, Rand Afrikaans University, Johannesburg, South Africa, 1986.
- [45] FRICK, M & BULLOCK, F, *Detour chromatic numbers*, *Discussiones Mathematicae Graph Theory*, **21**, pp. 283–291, 2001.
- [46] FRICK, M & HENNING, MA, *Extremal results on defective colourings of graphs*, *Discrete Mathematics*, **126**, pp. 151–158, 1994.
- [47] GAREY, MR & JOHNSON, DS, *Computers and intractability: A guide to the theory of NP-completeness*, W.H. Freeman, New York, 1979.
- [48] GIBBONS, A, *Algorithmic graph theory*, Cambridge University Press, Cambridge, 1985.
- [49] GLOVER, F, *Future paths for integer programming and links to artificial intelligence*, *Computers and Operations Research*, **13**, No 5, pp. 533–549, 1986.
- [50] GLOVER, F & LAGUNA, M, *Tabu search*, *Modern heuristic techniques for combinatorial problems*, edited by CR Reeves, Blackwell Scientific Publications, Oxford, 1993.
- [51] GODDARD, W, *Acyclic colourings of planar graphs*, *Discrete Mathematics*, **91**, pp. 91–94, 1991.
- [52] GOULD, R, *Graph theory*, The Benjamin/Cummings Publishing Company, Menlo Park, 1988.
- [53] GROSS, JL & YELLEN, J, *Graph theory and its applications*, CRC Press, London, 1999.
- [54] HARARY, F, *Conditional colourability in graphs*, *Graphs and applications*, *Proceedings of the 1st Colorado Symposium on Graph Theory*, John Wiley and Sons, New York, pp. 127–136, 1985.
- [55] HARARY, F & JONES, K, *Conditional colourability II: Bipartite variations*, *Congressus Numerantium*, **50**, pp. 205–218, 1985.
- [56] HARTMAN, CM, *Extremal problems in graph theory*, PhD–dissertation, University of Illinois, Chicago, Illinois, 1997.
- [57] HARTSFIELD, N & RINGEL, G, *Pearls in graph theory — A comprehensive introduction*, Academic Press, Boston, 1990.
- [58] HATTINGH, JH, *Some aspects of the theory of circulant graphs*, PhD–dissertation, Rand Afrikaans University, Johannesburg, South Africa, 1989.
- [59] HATTINGH, JH, Professor and Chair, Department of Mathematics and Statistics, Georgia State University, Atlanta, USA, Personal communications, 2007.
- [60] HEAWOOD, PJ, *Map-colour theorem*, *Quarterly Journal of Pure Applied Mathematics*, **24**, pp. 332–338, 1890.
- [61] HEESCH, H, *Untersuchungen zum Vierfarben-problem*, B-I-Hochschulschriften **810/ 810a/810b**, Bibliographisches Institut, Mannheim, Vienna, Zürich, 1969.
- [62] HENNING, MA & VAN VUUREN, JH, *A first course in graph theory*, in press.

- [63] HERRMANN, F & HERTZ, A, *Finding the chromatic number by means of critical graphs*, The ACM Journal of Experimental Algorithmics, **7**, pp. 1–12, 2002.
- [64] HERTZ, A & DE WERRA, D, *Using tabu search techniques for graph colouring*, Computing, **39**, pp. 345–351, 1987.
- [65] HEUBERGER, C, *On planarity and colourability of circulant graphs*, Discrete Mathematics, **68**, pp. 153–169, 2003.
- [66] HILBERT, D, *Mathematical problems*, Lecture delivered before the International Congress of Mathematicians at Paris in 1900, Bulletin of the American Mathematical Society, **8**, pp. 437–479, 1902.
- [67] HOWE, D, *The free on-line dictionary of computing*, [Online], [cited 2003, May 28], Available from: <http://foldoc.doc.ic.ac.uk/foldoc/contents.html>
- [68] JANSSEN, J, *An algorithmic approach to multicolouring*, [Online], [cited 2003, May 28], Available from: <http://www.math.umt.edu/bigsky/colloq99.html>
- [69] JENSEN, TR & TOFT, B, *Graph colouring problems*, John Wiley & Sons, New York, 1995.
- [70] JOHNS, G & SABA, F, *On the path chromatic number of a graph*, Annals of the New York Academy of Sciences, **576**, pp. 275–280, 1989.
- [71] JOHNSON, DS, ARAGON, CR, MCGEOCH, LA & SCHEVON, C, *Optimization by simulated annealing: An experimental evaluation — Part II: Graph colouring and number partitioning*, Operations Research, **39**, No 3, pp. 378–406, 1991.
- [72] JOHNSON, DS & TRICK, MA, *DIMACS series in discrete mathematics and theoretical computer science*, **26** [Online], [cited 2006, November 29], Available from: <http://dimacs.rutgers.edu/Volumes/Vol26.html>
- [73] JOHRI, A & MATULA, DW *Probabilistic bounds and heuristic algorithms for colouring large random graphs*, Technical Report **82-CSE-6**, Southern Methodist University, Dallas, 1982.
- [74] KARP, RM, *Reducibility among combinatorial problems*, Complexity of computer computations, Plenum Press, New York, pp. 85–103, 1972.
- [75] KEMPE, AB, *On the geographical problems of four colours*, American Journal of Mathematics, **2**, pp. 193–200, 1879.
- [76] KUMANDURI, R & ROMERO, C, *Number theory with computer applications*, Prentice Hall, Upper Saddle River, 1998.
- [77] LEIGHTON, FT, *A graph colouring algorithm for large scheduling problems*, Journal of Research of the National Bureau of Standards, **84**, pp. 489–506, 1979.
- [78] LICK, DR & WHITE, AT, *k-Degenerate graphs*, Canadian Journal of Mathematics, **22**, No 5, pp. 1082–1096, 1970.
- [79] LOVÁSZ, L, *On decompositions of graphs*, Studia Scientiarum Mathematicarum Hungarica, **1**, pp. 237–238, 1966.
- [80] MATULA, DW, MARBLE, G & ISAACSON, JD, *Graph colouring algorithms*, Graph Theory and Computing, edited by RC Read, Academic press, New York, pp. 109–122, 1972.
- [81] MESZKA, M, Assistant professor, Faculty of Applied Mathematics, AGH University of Science and Technology, Kraków, Poland, Personal communications, 2006 & 2007.
- [82] MESZKA, M, NEDELA, R & ROSA, A, *Circulants and the chromatic index of Steiner triple systems*, Mathematica Slovaca, **56**, No 4, pp. 371–378, 2006.
- [83] MEHORTA, A, & TRICK, M, *A column generation approach for graph coloring*, INFORMS Journal on Computing, **8**, No 4, pp. 344–354, 1996.

- [84] MICHALEWICZ, Z & FOGEL, DB, *How to solve it: Modern heuristics*, Springer-Verlag, Berlin, 2000.
- [85] MITCHEM, J, *An extension of Brooks' theorem to n -degenerate graphs*, *Discrete Mathematics*, **17**, pp. 291–298, 1977.
- [86] MITCHEM, J, *On the genus of graphs with Lick-White number k* , *Proceedings of the American Mathematical Society*, **69**, pp. 349–354, 1978.
- [87] MYCIELSKI, J, *Sur le coloriage des graphes*, *Colloquium Mathematicum*, **3**, pp. 161–162, 1955.
- [88] MYNHARDT, CM & BROERE, I, *Generalized colourings of graphs*, *Graph Theory and its Applications to Algorithms and Computer Science*, edited by Y Alavi *et al.*, John Wiley and Sons, New York, pp. 583–594, 1985.
- [89] NARAYANAN, L, *Channel assignment and graph multicolouring*, [Online], [cited 2003, June 12], Available from: <http://www.cs.concordia.ca/~teaching/comp691k/>
- [90] O'CONNOR, JJ & ROBERTSON, EF, *The four-colour theorem*, [Online], [cited 2002, April 11], Available from: http://www-groups.dcs.st-and.ac.uk/~history/HistTopics/The_four_colour_theorem.html
- [91] OELLERMANN, O, *The four-colour problem and its connection to South African flora*, [Online], [cited 2002, October 15], Available from: <http://www.uwinnipeg.ca/~oellerm/guthrie/FourColor.html>
- [92] PEEMÖLLER, J, *A correction to Brelaz's modification of Brown's colouring algorithm*, *Communications of the Association for Computing Machinery*, **26**, No 8, pp. 595–597, 1983.
- [93] RAYWARD-SMITH, VJ, OSMAN, IH, REEVES, CR & SMITH, GD, *Modern heuristic search methods*, John Wiley & Sons, Chichester, 1996.
- [94] READ, RC & WILSON, RJ, *An atlas of graphs*, Oxford University Press, New York, 1998.
- [95] REGO, C & ALIDAEI, B, *Metaheuristic optimization via memory and evolution: Tabu search and scatter search*, Kluwer Academic Publishers, Boston, 2005.
- [96] ROBERTSON, N, SANDERS, DP, SEYMOUR, PD & THOMAS, R, *The four-colour theorem*, *Journal of Combinatorial Theory, Series B*, **70**, pp. 2–44, 1997.
- [97] ROBERTSON, N, SANDERS, DP, SEYMOUR, PD & THOMAS, R, *A new proof of the four-colour theorem*, *Electronic Research Announcement of the American Mathematical Society*, **2**, pp. 17–25, 1996 [Online], [cited 2003, March 14], Available from: <http://www.ams.org/journals/era/1996-02-01/>
- [98] ROSS, SM, *A first course in probability — 6th Edition*, Prentice Hall, Upper Saddle River, 2002.
- [99] ROTMAN, JJ, *An introduction to the theory of groups*, *Graduate Texts in Mathematics Series*, Springer-Verlag, New York, 1995.
- [100] SACHS, H, *Finite graphs*, *Recent Progress in Combinatorics*, *Proceedings of the Third Waterloo Conference on Combinatorics*, Academic Press, New York, pp. 175–184, 1969.
- [101] SCHAERF, A, *Local search techniques for large high school timetabling problems*, *IEEE Transactions on systems, man, and cybernetics — Part A: Systems and humans*, **29**, No 4, pp. 368–377, 1999.
- [102] SZEKERES, G & WILF, HS, *An inequality for the chromatic number of a graph*, *Journal of Combinatorial Theory*, **4**, pp. 1–3, 1968.
- [103] SUDERMAN, M, *Skeletons in the closet*, *Complexity and computability lecture notes* [Online], [cited 2007, January 24], Available from: <http://cgm.cs.mcgill.ca/~msuder/courses/250/lectures/complexity/>, last updated 2002, April 16.

- [104] THOMAS, R, *The four-colour theorem*, [Online], [cited 2002, October 15], Available from: <http://www.math.gatech.edu/~thomas/FC/fourcolor.html>
- [105] THOMASSEN, C, *Every planar graph is 5-choosable*, Journal of Combinatorial Theory, Series B, **62**, pp. 180–181, 1994.
- [106] TRICK, MA, *Network resources for colouring a graph*, [Online], [cited 2003, May 28], Available from: <http://mat.gsia.cmu.edu/COLOR/color.html>
- [107] TUTTE, WT, *Colouring problems*, The Mathematical Intelligencer, **1**, pp. 72–75, 1978.
- [108] TUZA, Z, *Graph colourings with local constraints — A survey*, Discussiones Mathematicae Graph Theory, **17**, No 2, pp. 161–228, 1997.
- [109] WALLIS, WD, *A beginner's guide to graph theory*, Birkhäuser Boston, New York, 2000.
- [110] WEISSTEIN, EW, *Circulant graph*, [Online], [cited 2006, October 5], Available from: <http://mathworld.wolfram.com/CirculantGraph.html>
- [111] WELSH, DJA & POWELL, MB, *An upper bound for the chromatic number of a graph and its application to timetabling problems*, The Computer Journal, **10**, pp. 85–87, 1967.
- [112] WEST, DB, *Introduction to graph theory — 2nd Edition*, Prentice Hall, Upper Saddle River, 2001.
- [113] WILSON, R, *Four colours suffice*, Penguin Books, London, 2002.
- [114] WOODALL, D, *Improper colourings of graphs*, Graph colourings, Pitman research notes in mathematics series, edited by R Nelson & RJ Wilson, Longman Scientific and Technical, Harlow, 1990.

Appendix A

The Four-colour Theorem for Map Colouring

A.1 Kempe's Attempted Proof of the Four-colour Theorem

Alfred Bray Kempe used an inductive argument known as the method of Kempe chains in his 1879 attempted proof of the four-colour theorem [75]. Consider a map in which every region is coloured, using the colours 1, 2, 3 or 4 except for one region, say X. If this final region X is not surrounded by regions of all four colours, then a colour remains to be assigned to X. Hence suppose that regions of all four colours surround X. If X is surrounded by regions A, B, C, D in order, coloured 1, 2, 3 and 4 respectively then there are two cases to consider, as illustrated in Figure A.1: (1) there is no chain of adjacent regions from A to C alternately coloured 1 and 3, or (2) there is a chain of adjacent regions from A to C alternately coloured 1 and 3.

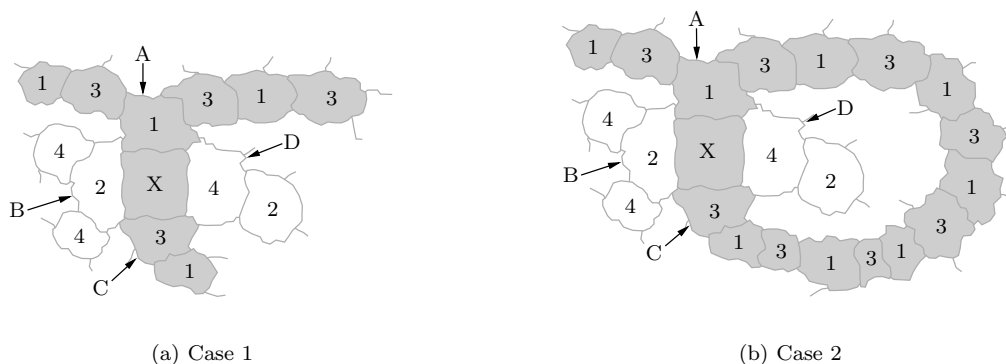
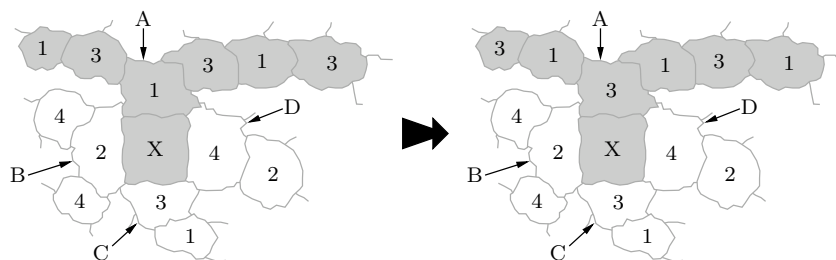


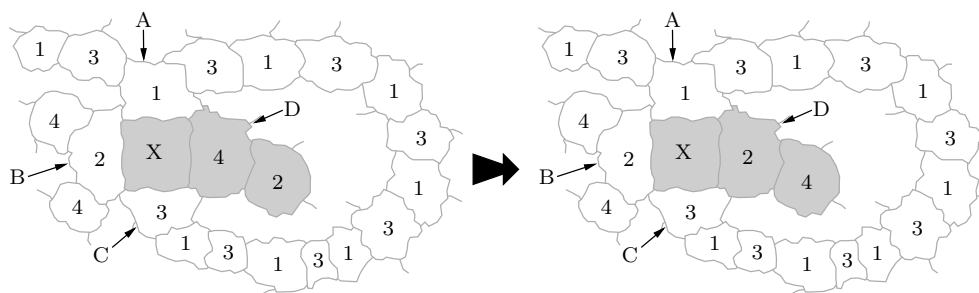
FIGURE A.1: An illustration of the two cases that may develop when all four surrounding regions of an uncoloured region X, are already coloured. Case 1, where there is no chain of adjacent regions from A to C alternately coloured 1 and 3, is shown in (a), and case 2, where there is a chain of adjacent regions from A to C alternately coloured 1 and 3, is shown in (b).

Suppose case (1) holds. Change the colour of region A to colour 3, and then interchange the colours in all chains of successive regions coloured alternately with colours 1 and 3, starting from region A, as illustrated in Figure A.2(a). Since region C is not in any such chain, it remains coloured 3 and there is now no region adjacent to region X coloured 1. Colour region X, using colour 1. If case (2) holds then there can be no chain of successive regions coloured alternately with colours 2 and 4, starting from region D and ending in region B, as illustrated in Figure A.2(b). (Such a chain cannot cross the chain of successive regions coloured alternately with colours 1 and 3, starting from region A.) Hence case (1) holds for D and one may interchange colours 2 and 4 as above, and finally colour region X, using colour 4

[90]. Kempe finally generalized the above procedure to be able to colour an uncoloured country X which is surrounded by five regions coloured in all four colours [113].



(a) Recolouring of Case 1



(b) Recolouring of Case 2

FIGURE A.2: *Kempe chains* may be used to recolour parts of a map.

A.2 The Error in Kempe's Proof of the Four-colour Theorem

As mentioned in §1.1, Kempe's proof was refuted by Percy John Heawood in his first paper. In this paper, Heawood gave an example of a map, shown in Figure A.3 which, although it could easily be four-coloured, showed that Kempe's proof technique did not work in general [29, 113].

Each of the twenty-five countries of the map in Figure A.3 has been coloured, using colours 1, 2, 3 and 4, except for the country P in the middle. Attempting to recolour two of P's neighbours in such a way that there is a colour available for P, one first notices that the neighbours of P coloured with colours 2 or 3 are connected by a chain of successive countries coloured alternatively with colours 2 and 3. This chain separates the part of a chain of successive countries coloured alternatively with colours 1 and 4 above P, from the part of a chain of successive countries coloured alternatively with colours 1 and 4 below P, as shown in Figure A.3(a). Thus, the colours of the part of a chain of successive countries coloured alternatively with colours 1 and 4 above P may be interchanged without affecting the part of a chain of successive countries coloured alternatively with colours 1 and 4 below P, as shown in Figure A.3(b). Alternatively, one could have carried out a different interchange of colours. The neighbours of P coloured with colours 2 and 4, are connected by a chain of successive countries coloured alternatively with colours 2 and 4. This chain of countries separates the part of a chain of successive countries coloured alternatively with colours 1 and 3 above P from the part a chain of successive countries coloured alternatively with colours 1 and 3 below P, as shown in Figure A.3(c). The colours of the part of a chain of successive countries coloured alternatively with colours 1 and 3 below P may thus be interchanged without affecting the part of a chain of successive countries coloured alternatively with colours 1 and 3 above P, as in Figure A.3(d). Either of these colour interchanges is permissible on its own, but not simultaneously. If the colours of the part of a chain of successive countries coloured alternatively with colours 1 and 4 above

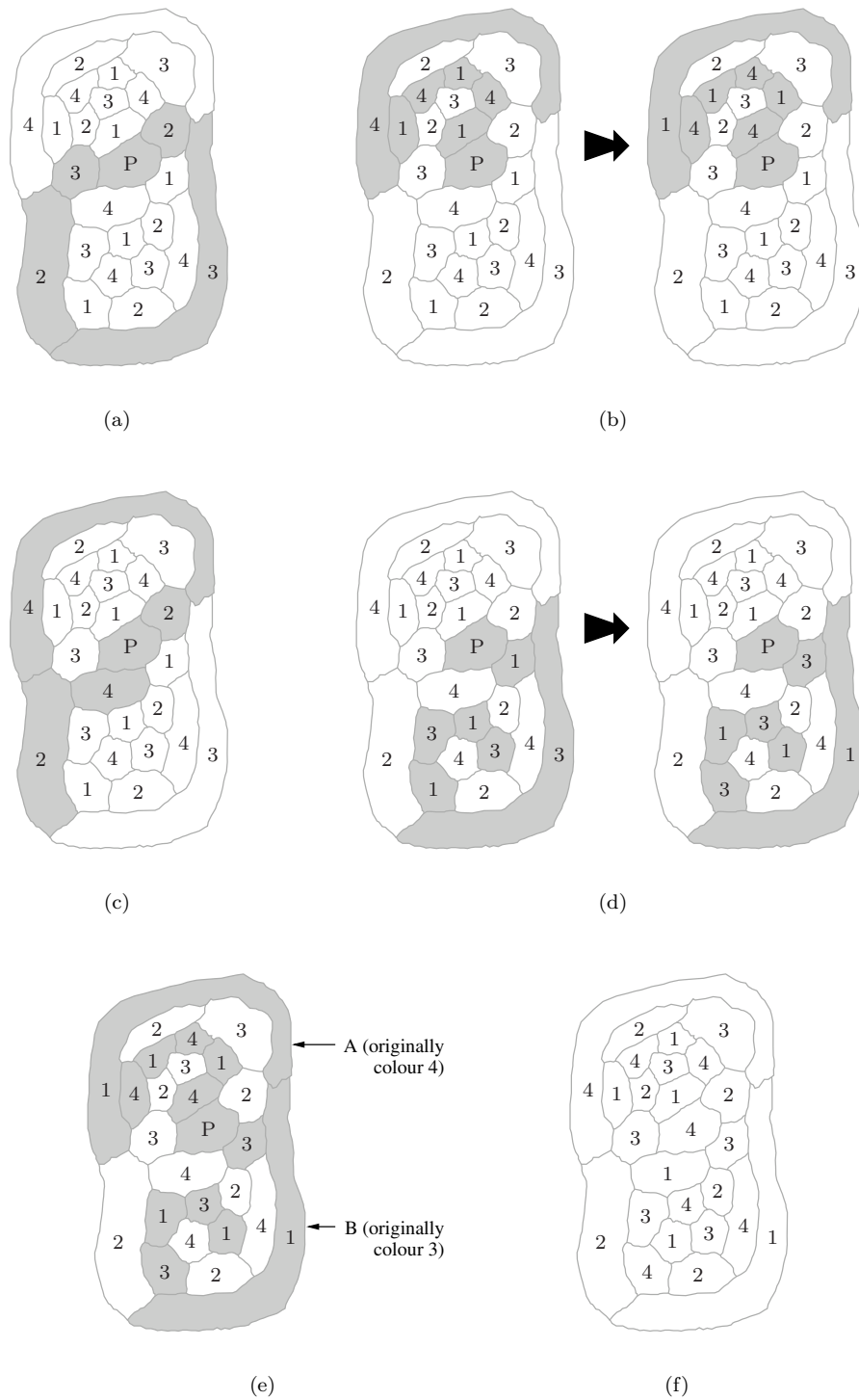


FIGURE A.3: Heawood's example of a map which, although easily four-colourable, showed that Kempe's proof technique did not work in general. In (a) and (b) a Kempe chain (case 2) is used to recolour one of the regions surrounding P , and in (c) and (d) another Kempe chain (also case 2) is used to recolour another one of the regions surrounding P . The result when both colour interchanges in (a) and (b), and (c) and (d) are performed simultaneously, is illustrated in (e). A four-colouring of the map is given in (f).

P and the part of a chain of successive countries coloured alternatively with colours 1 and 3 below P, are both interchanged, then country A which was originally coloured with colour 4, and country B which was originally coloured with colour 3, are both coloured with colour 1 as indicated in Figure A.3(e), which is not permissible. Although Kempe's proof did not work for Heawood's example, this example could easily be coloured with four colours as indicated in Figure A.3(f) [113].

Appendix B

The Five-colour Theorem

Although Kempe's proof of the four-colour theorem contained a flaw as pointed out by Heawood (described in Appendix A), Heawood was able to use Kempe's technique to prove the five-colour theorem, *i.e.* that every graph may be five-coloured [29, 91].

B.1 Heawood's Cartographic Proof of the 5-colour Theorem

To prove the five-colour theorem Heawood utilised the notion of Kempe chains as well as an idea of Cayley, called minimal non-five-colourable maps, *i.e.* maps that cannot be coloured with five colours and they have as few countries as possible (hence any map with fewer countries can be coloured with five colours). Assume that the five-colour theorem is false, so that there are some maps that *cannot* be coloured with five colours. Among these special maps that require more than five colours, consider a minimal non-five-colourable map. Thus, this map *cannot* be coloured with five colours, but any map with fewer countries *can* be coloured with five colours. It can further be shown that every map has at least one country with five or fewer neighbours. This implies that the chosen minimal non-five-colourable map must contain a country C with at most five neighbours as indicated in Figure B.1(a). Remove a boundary line of country C as indicated in Figure B.1(b) and merge C with its former neighbour. This yields a map with fewer countries which, by our assumption of a minimal non-five-colourable map, can be coloured with five colours, as indicated in Figure B.1(c). Reinststate the boundary line of country C as indicated in Figure B.1(d). If country C has two, three or four neighbours, a colour remains to colour this country. Hence a minimal non-five-colourable map cannot contain a country with two, three or four neighbours. If country C has five neighbours, it is possible that these neighbours are coloured

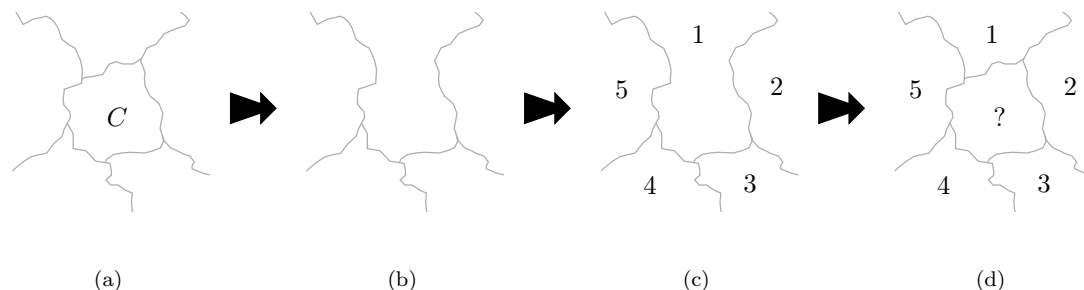


FIGURE B.1: Obtain a new map (c) from the minimal non-five-colourable map in (a), by removing a boundary line of country C as indicated in (b) and merge C with its former neighbour. Colour this map as indicated in (c) and try to colour the original map after the boundary line of country C has been reinstated as indicated in (d).

with all five available colours, as is the case in Figure B.1(d). Kempe chains may then be used to perform one colour interchange so that the neighbours of C are now coloured with only four colours, and country C may be coloured with the fifth colour. Thus, a minimal non-five-colourable map can also not contain a country with five neighbours, which contradicts our assumption, and completes the proof of the five-colour theorem [52, 57, 113].

B.2 Heawood's Graph Theoretic Proof of the 5-colour Theorem

Heawood's proof of the five-colour theorem also embodies a polynomial time algorithm to five-colour the vertices of a planar graph [48]. The following well known graph theorem is used in the proof of the five-colour theorem, and a proof of this theorem may be found in [29].

Theorem B.1 *Every planar graph contains a vertex of degree 5 or less.* ■

In order to proof the five-colour theorem one also needs to define the subgraph $\langle i, j \rangle_G$ of the graph G as the subgraph induced by the vertices of G that are coloured either with colour i or with colour j ¹ [9].

Theorem B.2 (The Five-Colour Theorem) *Every planar graph is 5-colourable.*

Proof: The proof is by induction on the order p of a planar graph. Since all graphs of order 1, 2, 3, 4 or 5 may be coloured using five or fewer colours, the result is true if $1 \leq p \leq 5$.

Assume that all planar graphs of order $k \geq 5$ are 5-colourable, and let G be a planar graph of order $k + 1$. By Theorem B.1, G contains a vertex v of degree 5 or less. Embed G in the plane, so that G is a plane graph. Let $H = G - v$, then H is a plane graph of order k . By the inductive hypothesis, H is 5-colourable. Consider a 5-colouring of H , using the colours 1, 2, 3, 4 and 5. If all five colours were not assigned to the vertices adjacent to v , then the 5-colouring of H may be extended to the graph G by assigning to v a colour not assigned to the adjacent vertices to v . Finally, assume that $\deg_G(v) = 5$, and that all five colours are assigned to the vertices adjacent to v in a 5-colouring of H . Suppose that v_1, v_2, \dots, v_5 are the five vertices adjacent with v , arranged cyclically about v . Suppose, also, that v_1 has been coloured with colour 1, v_2 has been coloured with colour 2, and so on.

The subgraph $\langle 2, 4 \rangle_H$ includes v_2 and v_4 . The vertices v_2 and v_4 may or may not belong to distinct components of $\langle 2, 4 \rangle_H$. First, suppose that v_2 and v_4 belong to distinct components of $\langle 2, 4 \rangle_H$ as indicated in Figure B.2. Interchange colours 2 and 4 of the vertices of the component of $\langle 2, 4 \rangle_H$ containing v_2 , to produce a different 5-colouring of H . This 5-colouring assigns the colour 4 to both v_2 and v_4 . Colour 2 may then be assigned to v , which produces a 5-colouring of G , as shown in Figure B.2.

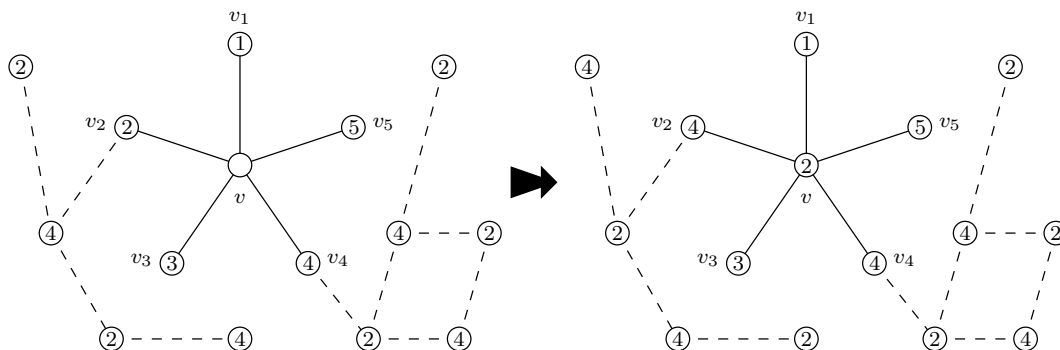


FIGURE B.2: A Kempe chain consisting of vertices coloured alternatively with colours 2 and 4, may be used to recolour a part of the graph such that a colour becomes available to colour vertex v .

¹For definitions of other graph theoretic terminology, see §2.1.

Suppose now that v_2 and v_4 belong to the same component of $\langle 2, 4 \rangle_H$ as indicated in Figure B.3. Then there exists a $v_2 - v_4$ path P in H , all of whose vertices are coloured alternatively with colour 2 and colour 4. The path P together with the path v_2, v, v_4 produces a cycle in G which either encloses v_1 and v_5 , or encloses v_3 as shown in Figure B.3. Hence, there does not exist a $v_3 - v_5$ path in H all of whose vertices are coloured alternatively with colour 3 and colour 5. Interchange colours 3 and 5 of the vertices of the component of $\langle 3, 5 \rangle_H$ containing v_3 , to produce a new 5-colouring of H in which both v_3 and v_5 are coloured with colour 5. If colour 3 is then assigned to v , a 5-colouring of G is obtained [9, 29, 53]. ■

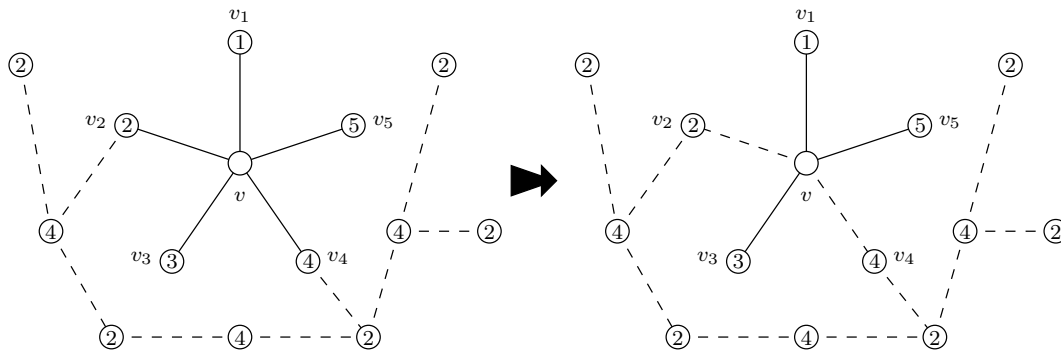


FIGURE B.3: A Kempe chain consisting of vertices coloured alternatively with colours 2 and 4, may be extended to a cycle separating v_3 and v_5 .

Appendix C

Sequences Satisfying the Conditions of Theorem 3.13

This appendix contains the table constructed by Burger and Grobler [22] which lists all sequences (x_i) up to and including x_{12} satisfying the conditions of Theorem 3.13 where the minimum number of twos in a sequence is given. Also, only the first one of the infinite tail of ones is listed.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	Graph
1													K_1
2	1												K_2
3	2	1											K_3
4	2	2	1										K_4
3	3	2	2	1									K_{222}
4	3	2	2	1									$K_{222} \cup K_4$
5	3	2	2	1									K_5
6	3	2	2	2	1								K_6
4	4	2	2	2	2	1							K_{2222}
5	4	2	2	2	2	1							$K_{2222} \cup K_5$
6	4	2	2	2	2	1							$K_{2222} \cup K_6$
3	3	3	2	2	2	1							K_{333}
4	3	3	2	2	2	1							$K_{333} \cup K_4$
5	3	3	2	2	2	1							$K_{333} \cup K_5$
6	3	3	2	2	2	1							$K_{333} \cup K_6$
4	4	3	2	2	2	1							$K_{2222} \cup K_{333}$
5	4	3	2	2	2	1							$K_{2222} \cup K_{333} \cup K_5$
6	4	3	2	2	2	1							$K_{2222} \cup K_{333} \cup K_6$
7	4	3	2	2	2	1							K_7
8	4	3	2	2	2	2	1						K_8

TABLE C.1: Sequences up to and including x_{12} satisfying the conditions of Theorem 3.13. Note that only the first one of the infinite tail of ones is listed and in all cases only the sequence with the minimum number of twos is listed.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	Graph
4	4	4	2	2	2	2	2	2	1				K_{3333}
5	4	4	2	2	2	2	2	2	1				$K_{3333} \cup K_5$
6	4	4	2	2	2	2	2	2	1				$K_{3333} \cup K_6$
7	4	4	2	2	2	2	2	2	1				$K_{3333} \cup K_7$
8	4	4	2	2	2	2	2	2	1				$K_{3333} \cup K_8$
3	3	3	3	2	2	2	2	1					K_{444}
4	3	3	3	2	2	2	2	1					$K_{444} \cup K_4$
5	3	3	3	2	2	2	2	1					$K_{444} \cup K_5$
6	3	3	3	2	2	2	2	1					$K_{444} \cup K_6$
4	4	3	3	2	2	2	2	1					$K_{2222} \cup K_{444}$
5	4	3	3	2	2	2	2	1					$K_{2222} \cup K_{444} \cup K_5$
6	4	3	3	2	2	2	2	1					$K_{2222} \cup K_{444} \cup K_6$
7	4	3	3	2	2	2	2	1					$K_{444} \cup K_7$
8	4	3	3	2	2	2	2	1					$K_{444} \cup K_8$
5	5	3	3	2	2	2	2	1					K_{22222}
6	5	3	3	2	2	2	2	1					$K_{22222} \cup K_6$
7	5	3	3	2	2	2	2	1					$K_{22222} \cup K_7$
8	5	3	3	2	2	2	2	1					$K_{22222} \cup K_8$
9	5	3	3	2	2	2	2	1					K_9
6	6	3	3	2	2	2	2	2	1				K_{222222}
7	6	3	3	2	2	2	2	2	1				$K_{222222} \cup K_7$
8	6	3	3	2	2	2	2	2	1				$K_{222222} \cup K_8$
9	6	3	3	2	2	2	2	2	1				$K_{222222} \cup K_9$
4	4	4	3	2	2	2	2	2	1				$K_{3333} \cup K_{444}$
5	4	4	3	2	2	2	2	2	1				$K_{3333} \cup K_{444} \cup K_5$
6	4	4	3	2	2	2	2	2	1				$K_{3333} \cup K_{444} \cup K_6$
7	4	4	3	2	2	2	2	2	1				$K_{3333} \cup K_{444} \cup K_7$
8	4	4	3	2	2	2	2	2	1				$K_{3333} \cup K_{444} \cup K_8$
5	5	4	3	2	2	2	2	2	1				$K_{22222} \cup K_{3333}$
6	5	4	3	2	2	2	2	2	1				$K_{22222} \cup K_{3333} \cup K_6$
7	5	4	3	2	2	2	2	2	1				$K_{22222} \cup K_{3333} \cup K_7$
8	5	4	3	2	2	2	2	2	1				$K_{22222} \cup K_{3333} \cup K_8$
9	5	4	3	2	2	2	2	2	1				$K_{3333} \cup K_9$
10	5	4	3	2	2	2	2	2	1				K_{10}
6	6	4	3	2	2	2	2	2	2	1			$K_{222222} \cup K_{3333}$
7	6	4	3	2	2	2	2	2	2	1			$K_{222222} \cup K_{3333} \cup K_7$
8	6	4	3	2	2	2	2	2	2	1			$K_{222222} \cup K_{3333} \cup K_8$
9	6	4	3	2	2	2	2	2	2	1			$K_{222222} \cup K_{3333} \cup K_9$
10	6	4	3	2	2	2	2	2	2	1			$K_{222222} \cup K_{10}$
4	4	4	4	2	2	2	2	2	2	2	2	1	
5	4	4	4	2	2	2	2	2	2	2	2	1	
6	4	4	4	2	2	2	2	2	2	2	2	1	
7	4	4	4	2	2	2	2	2	2	2	2	1	
8	4	4	4	2	2	2	2	2	2	2	2	1	

Table C.1 (continued): Sequences satisfying the conditions of Theorem 3.13.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	Graph
5	5	4	4	2	2	2	2	2	2	2	2	1	
6	5	4	4	2	2	2	2	2	2	2	2	1	
7	5	4	4	2	2	2	2	2	2	2	2	1	
8	5	4	4	2	2	2	2	2	2	2	2	1	
9	5	4	4	2	2	2	2	2	2	2	2	1	
10	5	4	4	2	2	2	2	2	2	2	2	1	
6	6	4	4	2	2	2	2	2	2	2	2	1	
7	6	4	4	2	2	2	2	2	2	2	2	1	
8	6	4	4	2	2	2	2	2	2	2	2	1	
9	6	4	4	2	2	2	2	2	2	2	2	1	
10	6	4	4	2	2	2	2	2	2	2	2	1	
3	3	3	3	3	2	2	2	2	2	1			K_{555}
4	3	3	3	3	2	2	2	2	2	1			$K_{555} \cup K_4$
5	3	3	3	3	2	2	2	2	2	1			$K_{555} \cup K_5$
6	3	3	3	3	2	2	2	2	2	1			$K_{555} \cup K_6$
4	4	3	3	3	2	2	2	2	2	1			$K_{2222} \cup K_{555}$
5	4	3	3	3	2	2	2	2	2	1			$K_{2222} \cup K_{555} \cup K_5$
6	4	3	3	3	2	2	2	2	2	1			$K_{2222} \cup K_{555} \cup K_6$
7	4	3	3	3	2	2	2	2	2	1			$K_{555} \cup K_7$
8	4	3	3	3	2	2	2	2	2	1			$K_{555} \cup K_8$
5	5	3	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{555}$
6	5	3	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{555} \cup K_6$
7	5	3	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{555} \cup K_7$
8	5	3	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{555} \cup K_8$
9	5	3	3	3	2	2	2	2	2	1			$K_{555} \cup K_9$
6	6	3	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{555}$
7	6	3	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{555} \cup K_7$
8	6	3	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{555} \cup K_8$
9	6	3	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{555} \cup K_9$
4	4	4	3	3	2	2	2	2	2	1			$K_{3333} \cup K_{555}$
5	4	4	3	3	2	2	2	2	2	1			$K_{3333} \cup K_{555} \cup K_5$
6	4	4	3	3	2	2	2	2	2	1			$K_{3333} \cup K_{555} \cup K_6$
7	4	4	3	3	2	2	2	2	2	1			$K_{3333} \cup K_{555} \cup K_7$
8	4	4	3	3	2	2	2	2	2	1			$K_{3333} \cup K_{555} \cup K_8$
5	5	4	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{3333} \cup K_{555}$
6	5	4	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{3333} \cup K_{555} \cup K_6$
7	5	4	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{3333} \cup K_{555} \cup K_7$
8	5	4	3	3	2	2	2	2	2	1			$K_{22222} \cup K_{3333} \cup K_{555} \cup K_8$
9	5	4	3	3	2	2	2	2	2	1			$K_{3333} \cup K_{555} \cup K_9$
10	5	4	3	3	2	2	2	2	2	1			$K_{555} \cup K_{10}$
6	6	4	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{3333} \cup K_{555}$
7	6	4	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{3333} \cup K_{555} \cup K_7$
8	6	4	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{3333} \cup K_{555} \cup K_8$
9	6	4	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{3333} \cup K_{555} \cup K_9$
10	6	4	3	3	2	2	2	2	2	1			$K_{222222} \cup K_{555} \cup K_{10}$
11	6	4	3	3	2	2	2	2	2	1			K_{11}
12	6	4	3	3	2	2	2	2	2	2	1		K_{12}

Table C.1 (continued): Sequences satisfying the conditions of Theorem 3.13.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	Graph
4	4	4	4	3	2	2	2	2	2	2	2	1	
5	4	4	4	3	2	2	2	2	2	2	2	1	
6	4	4	4	3	2	2	2	2	2	2	2	1	
7	4	4	4	3	2	2	2	2	2	2	2	1	
8	4	4	4	3	2	2	2	2	2	2	2	1	
5	5	4	4	3	2	2	2	2	2	2	2	1	
6	5	4	4	3	2	2	2	2	2	2	2	1	
7	5	4	4	3	2	2	2	2	2	2	2	1	
8	5	4	4	3	2	2	2	2	2	2	2	1	
9	5	4	4	3	2	2	2	2	2	2	2	1	
10	5	4	4	3	2	2	2	2	2	2	2	1	
6	6	4	4	3	2	2	2	2	2	2	2	1	
7	6	4	4	3	2	2	2	2	2	2	2	1	
8	6	4	4	3	2	2	2	2	2	2	2	1	
9	6	4	4	3	2	2	2	2	2	2	2	1	
10	6	4	4	3	2	2	2	2	2	2	2	1	
11	6	4	4	3	2	2	2	2	2	2	2	1	
12	6	4	4	3	2	2	2	2	2	2	2	1	
3	3	3	3	3	3	2	2	2	2	2	2	1	K_{666}
4	3	3	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_4$
5	3	3	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_5$
6	3	3	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_6$
4	4	3	3	3	3	2	2	2	2	2	2	1	$K_{2222} \cup K_{666}$
5	4	3	3	3	3	2	2	2	2	2	2	1	$K_{2222} \cup K_{666} \cup K_5$
6	4	3	3	3	3	2	2	2	2	2	2	1	$K_{2222} \cup K_{666} \cup K_6$
7	4	3	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_7$
8	4	3	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_8$
5	5	3	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{666}$
6	5	3	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{666} \cup K_6$
7	5	3	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{666} \cup K_7$
8	5	3	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{666} \cup K_8$
9	5	3	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_9$
6	6	3	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{666}$
7	6	3	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{666} \cup K_7$
8	6	3	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{666} \cup K_8$
9	6	3	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{666} \cup K_9$
4	4	4	3	3	3	2	2	2	2	2	2	1	$K_{3333} \cup K_{666}$
5	4	4	3	3	3	2	2	2	2	2	2	1	$K_{3333} \cup K_{666} \cup K_5$
6	4	4	3	3	3	2	2	2	2	2	2	1	$K_{3333} \cup K_{666} \cup K_6$
7	4	4	3	3	3	2	2	2	2	2	2	1	$K_{3333} \cup K_{666} \cup K_7$
8	4	4	3	3	3	2	2	2	2	2	2	1	$K_{3333} \cup K_{666} \cup K_8$
5	5	4	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{3333} \cup K_{666}$
6	5	4	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{3333} \cup K_{666} \cup K_6$
7	5	4	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{3333} \cup K_{666} \cup K_7$
8	5	4	3	3	3	2	2	2	2	2	2	1	$K_{22222} \cup K_{3333} \cup K_{666} \cup K_8$
9	5	4	3	3	3	2	2	2	2	2	2	1	$K_{3333} \cup K_{666} \cup K_9$
10	5	4	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_{10}$

Table C.1 (continued): Sequences satisfying the conditions of Theorem 3.13.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	Graph
6	6	4	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{3333} \cup K_{666}$
7	6	4	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{3333} \cup K_{666} \cup K_7$
8	6	4	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{3333} \cup K_{666} \cup K_8$
9	6	4	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{3333} \cup K_{666} \cup K_9$
10	6	4	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{3333} \cup K_{666} \cup K_{10}$
11	6	4	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_{11}$
12	6	4	3	3	3	2	2	2	2	2	2	1	$K_{666} \cup K_{12}$
5	5	5	3	3	3	2	2	2	2	2	2	1	K_{33333}
6	5	5	3	3	3	2	2	2	2	2	2	1	$K_{33333} \cup K_6$
7	5	5	3	3	3	2	2	2	2	2	2	1	$K_{33333} \cup K_7$
8	5	5	3	3	3	2	2	2	2	2	2	1	$K_{33333} \cup K_8$
9	5	5	3	3	3	2	2	2	2	2	2	1	$K_{33333} \cup K_9$
10	5	5	3	3	3	2	2	2	2	2	2	1	$K_{33333} \cup K_{10}$
6	6	5	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{33333}$
7	6	5	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{33333} \cup K_7$
8	6	5	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{33333} \cup K_8$
9	6	5	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{33333} \cup K_9$
10	6	5	3	3	3	2	2	2	2	2	2	1	$K_{222222} \cup K_{33333} \cup K_{10}$
11	6	5	3	3	3	2	2	2	2	2	2	1	$K_{33333} \cup K_{11}$
12	6	5	3	3	3	2	2	2	2	2	2	1	$K_{33333} \cup K_{12}$
4	4	4	4	3	3	2	2	2	2	2	2	1	
5	4	4	4	3	3	2	2	2	2	2	2	1	
6	4	4	4	3	3	2	2	2	2	2	2	1	
7	4	4	4	3	3	2	2	2	2	2	2	1	
8	4	4	4	3	3	2	2	2	2	2	2	1	
5	5	4	4	3	3	2	2	2	2	2	2	1	
6	5	4	4	3	3	2	2	2	2	2	2	1	
7	5	4	4	3	3	2	2	2	2	2	2	1	
8	5	4	4	3	3	2	2	2	2	2	2	1	
9	5	4	4	3	3	2	2	2	2	2	2	1	
10	5	4	4	3	3	2	2	2	2	2	2	1	
6	6	4	4	3	3	2	2	2	2	2	2	1	
7	6	4	4	3	3	2	2	2	2	2	2	1	
8	6	4	4	3	3	2	2	2	2	2	2	1	
9	6	4	4	3	3	2	2	2	2	2	2	1	
10	6	4	4	3	3	2	2	2	2	2	2	1	
11	6	4	4	3	3	2	2	2	2	2	2	1	
12	6	4	4	3	3	2	2	2	2	2	2	1	
7	7	4	4	3	3	2	2	2	2	2	2	1	
8	7	4	4	3	3	2	2	2	2	2	2	1	
9	7	4	4	3	3	2	2	2	2	2	2	1	
10	7	4	4	3	3	2	2	2	2	2	2	1	
11	7	4	4	3	3	2	2	2	2	2	2	1	
12	7	4	4	3	3	2	2	2	2	2	2	1	

Table C.1 (continued): Sequences satisfying the conditions of Theorem 3.13.

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	Graph
5	5	5	4	3	3	2	2	2	2	2	2	1	
6	5	5	4	3	3	2	2	2	2	2	2	1	
7	5	5	4	3	3	2	2	2	2	2	2	1	
8	5	5	4	3	3	2	2	2	2	2	2	1	
9	5	5	4	3	3	2	2	2	2	2	2	1	
10	5	5	4	3	3	2	2	2	2	2	2	1	
6	6	5	4	3	3	2	2	2	2	2	2	1	
7	6	5	4	3	3	2	2	2	2	2	2	1	
8	6	5	4	3	3	2	2	2	2	2	2	1	
9	6	5	4	3	3	2	2	2	2	2	2	1	
10	6	5	4	3	3	2	2	2	2	2	2	1	
11	6	5	4	3	3	2	2	2	2	2	2	1	
12	6	5	4	3	3	2	2	2	2	2	2	1	
7	7	5	4	3	3	2	2	2	2	2	2	1	
8	7	5	4	3	3	2	2	2	2	2	2	1	
9	7	5	4	3	3	2	2	2	2	2	2	1	
10	7	5	4	3	3	2	2	2	2	2	2	1	
11	7	5	4	3	3	2	2	2	2	2	2	1	
12	7	5	4	3	3	2	2	2	2	2	2	1	
13	7	5	4	3	3	2	2	2	2	2	2	1	K_{13}

Table C.1 (continued): Sequences satisfying the conditions of Theorem 3.13.

Appendix D

On the Placement of Strings in a Collection of Hats

In this appendix a summary is given of an article by Burger and Van Vuuren [23] that is used in §4.6 to determine an upper bound on the maximum degree chromatic number of a complete balanced multipartite graph.

D.1 Article Problem Description

The following problem is considered in a paper titled “On the placement of a number of strings in a collection of hats” by Burger and Van Vuuren [23]: “Suppose k strings of unit length are to be distributed amongst $x < k$ hats. If cuts in the strings are allowed, how should the strings (or parts thereof) be distributed amongst the hats so that, if the shortest string in each hat is discarded, the remaining (combined) string length in the hat with the most string is as small as possible?”

A partitioning of all k strings (or parts thereof) amongst the x hats is referred to as a **distribution** of the strings into the hats. The total string length in a specific hat or in a distribution after the strings have been partitioned into the hats is called the **weight** of the hat or of the distribution. **Purging** is the removal of the shortest string from each hat, while the total length of string removed from a specific hat, or from all the hats in a distribution, is called the **purging weight** of the hat in question or of the particular distribution. After purging, the total string length remaining in a specific hat or in a distribution is referred to as the **residual weight** of the hat or of the distribution. A distribution achieving the minimum value of the maximum residual weight, is called an **optimal distribution**. Denote this minimum value of the maximum residual weight by $\alpha(x, k)$. Finally, hats with a residual weight equal to the maximum value in a distribution (which need not be optimal), are referred to as **residually optimal**.

When small pieces of all strings are cut off and placed in an empty hat, this hat achieves a non-zero residual weight, while the maximum residual weight of the distribution will decrease. Therefore, there will never be an empty hat in an optimal distribution. This implies that each hat in a distribution contains at least one string (or part thereof) so that the process of purging in the problem description is a feasible process.

In order to minimize the maximum residual weight one would first like to ensure that all residual weights are as small as possible. This implies that the strings should be distributed amongst the hats in such a way that the residual weights are more or less equal. Secondly, one would prefer to purge longer rather than shorter pieces of string from each hat so that the residual weight in each hat is as small as possible. This implies that the shortest piece of string in each hat must be as long as possible. Therefore, the strings should be cut so as to be distributed evenly among the hats, but the strings should not be cut too much, since each cut introduces shorter pieces of string, which decreases the length of the shortest string in some hats. This trade-off between cutting and not cutting is illustrated in the following example.

Example D.1 Suppose $x = 5$ and $k = 7$. The following three possible distributions provide upper bounds on $\alpha(5, 7)$ given in the right-most column of the table.

	Hat 1	Hat 2	Hat 3	Hat 4	Hat 5	Bound
Distribution 1:	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$	$\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}$	$\alpha(5, 7) \leq \frac{6}{5}$
Residual weight:	$\frac{6}{5}$	$\frac{6}{5}$	$\frac{6}{5}$	$\frac{6}{5}$	$\frac{6}{5}$	
Distribution 2:	1, 1	1, 1	1	1	1	$\alpha(5, 7) \leq 1$
Residual weight:	1	1	0	0	0	
Distribution 3:	$\frac{5}{6}, \frac{5}{6}$	$\frac{5}{6}, \frac{5}{6}$	$\frac{5}{6}, \frac{5}{6}$	1	$\frac{1}{6}, \frac{1}{6}, \frac{1}{6}$ $\frac{1}{6}, \frac{1}{6}, \frac{1}{6}$	$\alpha(5, 7) \leq \frac{5}{6}$
Residual weight:	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	0	$\frac{5}{6}$	

In Distribution 1 all the strings were cut into five parts of equal length and one piece of each string was then placed in each hat, so that the residual weights of all hats are equal. From the table above it is clear that this strategy of cutting all strings into parts of equal length is not necessarily the best strategy. No cutting was performed in Distribution 2 in order to purge the longest possible piece of string from each hat. Again, the table above shows that this is also not necessarily the best strategy. Distribution 3, where some cutting, but not equal cutting for all strings, was performed, seems to be a better strategy than the other two strategies. ■

One might still intuitively expect that cutting strings such that equal residual weights result, is a necessary condition for optimality, but this is not the case. Burger and Van Vuuren [23, Corollary 1] proved that it is optimal *not* to cut strings if and only if $x|k$ or $x|(k+1)$. If $x|k$ it is optimal to distribute the uncut strings equally amongst the hats. This case results in equal residual weights in each hat. However, if $x|(k+1)$ the uncut strings may be distributed amongst the hats as equally as possible, except for one hat containing one string fewer than the other hats. In this case the residual lengths in each hat are *not* equal. For all other cases one can perform a balancing function by cutting off small pieces of strings in hats that are residually optimal and by placing these string pieces in hats that are not residually optimal, so as to lower the overall maximum residual weight.

D.2 Upper and Lower Bounds on the Parameter $\alpha(x, k)$

Placing at most $\lceil k/x \rceil$ strings in each hat, without any cutting, an uncut string may be removed from each hat, resulting in a general upper bound of $\lceil k/x \rceil - 1$ on $\alpha(x, k)$. Distribution 2 in Example D.1 is an illustration of achieving this upper bound. Considering the (possibly hypothetical) situation where an uncut string may still be purged from each hat, but in this case all hats have equal residual weights, a general lower bound of $(k-x)/x = k/x - 1$ on $\alpha(x, k)$ is obtained. Thus, the rather tight general bounds

$$\frac{k}{x} - 1 \leq \alpha(x, k) \leq \left\lceil \frac{k}{x} \right\rceil - 1 \quad (\text{D.2.1})$$

are established [23, Proposition 2]. Clearly, for the ideal situation where $x|k$, the lower and upper bounds in (D.2.1) are equal and the exact value of $\alpha(x, k)$ is obtained. The main results of Burger and Van Vuuren [23, Theorem 2], reproduced in Theorem D.1 below, are a number of (recursive in some cases) upper bounds on $\alpha(x, k)$ that constitute an improvement on the upper bound in (D.2.1) for the non-ideal situations. The theorem is given here without proof; however, it is followed by a discussion of the different non-ideal situations addressed by each case of the theorem.

Theorem D.1 Let s be the number of (uncut) strings placed in one hat, say the “last” hat, in a distribution where this hat contains fewer strings than the rest of the hats, i.e. $s \equiv k \pmod{\lceil \frac{k}{x} \rceil}$ and let x' be the number of empty hats after placing $\lceil \frac{k}{x} \rceil$ uncut strings in each of the first $x - x'$ hats, except maybe for the last hat that may be filled with only s strings, i.e. $x' = x - \lceil k/\lceil \frac{k}{x} \rceil \rceil$.

(a) If $s = \lceil \frac{k}{x} \rceil - 1$, then $\alpha(x, k) \leq \alpha(x - 1, k - s)$.

(b) If $x' = 0$ and $s \neq 0, \lceil \frac{k}{x} \rceil - 1$, then $\alpha(x, k) \leq \frac{(k-1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}$.

(c) If $x' > 0$ and $s \neq \lceil \frac{k}{x} \rceil - 1$, then $\alpha(x, k) \leq 1 / (\frac{1}{\alpha(x-x', k)} + \frac{1}{\alpha(x', k)})$. ■

In Theorem D.1(a) the last hat that has been filled contains exactly 1 uncut string fewer than the other hats that have been filled with uncut strings. In this case, as stated before, these $s = \lceil \frac{k}{x} \rceil - 1$ strings should not be cut. The remaining $k - s$ strings should then be distributed optimally amongst the remaining $x - 1$ hats, since the residual string weight of the hat containing the s strings will be less than the maximum residual string weights of the other hats after an optimal distribution of $k - s$ strings amongst $x - 1$ hats has been made.

Since $x = \lceil k/\lceil \frac{k}{x} \rceil \rceil$ in Theorem D.1(b), one may place $\lceil \frac{k}{x} \rceil$ uncut strings in each of the first $x - 1$ hats, and the remaining s strings in the last hat. In this case a portion y is cut from each string in the first $x - 1$ hats, and placed in the last hat, where the portion y is determined such that the residual weight of the last hat is equal to the residual weights of the first $x - 1$ hats after the cuts.

In Theorem D.1(c) x' hats remain empty after $\lceil \frac{k}{x} \rceil$ uncut strings are placed in each hat. In this case pieces of equal length are cut from each string and distributed optimally amongst the x' empty hats. Note that the residual weight of any hat is directly proportional to the string length (i.e., if any string length is changed by a factor ℓ , then $\alpha(x, k)$ will change by a factor ℓ). Therefore, two optimal distributions of k strings into x' and $x - x'$ hats respectively, may be combined by choosing the correct proportions of string from each optimal solution, i.e. proportions y_1 and y_2 is sought such that $y_1 + y_2 = 1$ and $y_1\alpha(x - x', k) = y_2\alpha(x', k)$.

A large body of work in the article is then dedicated to prove that the result of Theorem D.1(b) is in fact exact, i.e. if $k = \lceil k/x \rceil(x - 1) + s$ and $1 \leq s \leq \lceil k/x \rceil - 2$, then

$$\alpha(x, k) = \frac{(k - 1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}. \tag{D.2.2}$$

All cases may now be summarised as in Table D.1 [23, Table 5.1], with the case $x' = 0$ and $s = 0$ the ideal situation where $x|k$ and the lower and upper bounds in (D.2.1) are equal. The case $x' = 0$ and $0 < s < \lceil k/x \rceil - 1$ refer to (D.2.2). Finally, the two recursive upper bounds in Table D.1 refer to Theorem D.1(a) and (c), for the case $s = \lceil k/x \rceil - 1$ and the case $x' > 0$ and $0 \leq s < \lceil k/x \rceil - 1$, respectively.

	$s = 0$	$0 < s < \lceil \frac{k}{x} \rceil - 1$	$s = \lceil \frac{k}{x} \rceil - 1$
$x' = 0$	$\alpha(x, k) = \frac{k}{x} - 1$	$\alpha(x, k) = \frac{(k - 1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}$	$\alpha(x, k) \leq \alpha(x - 1, k - s)$
$x' > 0$	$\alpha(x, k) \leq \frac{1}{\frac{1}{\alpha(x-x', k)} + \frac{1}{\alpha(x', k)}}$		

TABLE D.1: Values of and upper bounds on $\alpha(x, k)$, depending on the values of $s \equiv k \pmod{\lceil \frac{k}{x} \rceil}$ and $x' = x - \lceil k/\lceil \frac{k}{x} \rceil \rceil$.

D.3 A Recursive Algorithmic Solution Approach

Using the cases in Table D.1, a recursive algorithm [23, Algorithm 5], given in pseudo-code in Algorithm 14, may be developed to determine a good upper bound $\bar{\alpha}(x, k)$ on $\alpha(x, k)$. As indicated in Table D.1, the upper bound $\bar{\alpha}(x, k)$ is exact in some cases.

Algorithm 14 Algorithm to determine $\bar{\alpha}(x, k)$

Input: The number of hats x and the number of strings k .

Output: An upper bound $\bar{\alpha}(x, k)$ on $\alpha(x, k)$.

```

1: if  $x \mid k$  then
2:   stop return  $\frac{k}{x} - 1$ 
3: end if
4:  $x' \leftarrow x - \lceil k / \lceil \frac{k}{x} \rceil \rceil$ ,  $s \leftarrow k \pmod{\lceil \frac{k}{x} \rceil}$ 
5: if  $s = \lceil \frac{k}{x} \rceil - 1$  then
6:   return  $\bar{\alpha}(x - 1, k - s)$ 
7: else
8:   if  $x' > 0$  then
9:     return  $1 / (\frac{1}{\bar{\alpha}(x - x', k)} + \frac{1}{\bar{\alpha}(x', k)})$ 
10:  else
11:    stop return  $\frac{(k-1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}$ 
12:  end if
13: end if

```

If k is not a multiple of x , a number of recursive calls may be made during the execution of Algorithm 14, some of which are terminating calls (Steps 2 or 11). Suppose the total number of terminating calls made during the execution of Algorithm 14 is $\Omega(x, k)$ and denote the values of x , x' and k during terminating call i of Algorithm 14 by x_i , x'_i and k_i respectively for all $i = 1, \dots, \Omega(x, k)$, with the convention that $x_1 = x$ and $k_1 = k$. Furthermore, let

$$a_i = \left\lceil \frac{k_i}{x_i} \right\rceil - 1 \quad \text{for all } i = 0, \dots, \Omega(x, k). \quad (\text{D.3.1})$$

Then $a_i + 1$ is the number of larger string proportions in each of the $x_i - 1$ hats distributed during the i -th recursive call of Algorithm 14. Values of $\bar{\alpha}(x, k)$, as computed by Algorithm 14, are shown in Table D.2 [23, Table 5.2] for $1 \leq x \leq 17$ and $2 \leq k \leq 18$. Finally, Burger and Van Vuuren proved that the worst-case time complexity of Algorithm 14 is $\Omega(x, k) = \mathcal{O}(\log \log x)$.

x	k																	
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
2	-	1 [†]	1	2 [†]	2	3 [†]	3	4 [†]	4	5 [†]	5	6 [†]	6	7 [†]	7	8 [†]	8	
3	-	-	3 [‡] / ₄	1 [†]	1	12 [*] / ₇	2 [†]	2	27 [*] / ₁₀	3 [†]	3	48 [*] / ₁₃	4 [†]	4	75 [*] / ₁₆	5 [†]	5	
4	-	-	-	3 [†] / ₄	5 [‡] / ₆	1 [†]	1	8 [‡] / ₅	9 [*] / ₅	2 [†]	2	18 [*] / ₇	39 [*] / ₁₄	3 [†]	3	32 [*] / ₉	34 [*] / ₉	
5	-	-	-	-	2 [‡] / ₃	5 [†] / ₆	7 [‡] / ₈	1 [†]	1	8 [†] / ₅	22 [‡] / ₁₃	24 [*] / ₁₃	2 [†]	2	5 [‡] / ₂	8 [*] / ₃	17 [*] / ₆	
6	-	-	-	-	-	2 [†] / ₃	3 [‡] / ₄	7 [†] / ₈	9 [‡] / ₁₀	1 [†]	1	8 [‡] / ₅	22 [†] / ₁₃	7 [‡] / ₄	15 [*] / ₈	2 [†]	2	
7	-	-	-	-	-	-	2 [‡] / ₃	3 [†] / ₄	4 [‡] / ₅	9 [†] / ₁₀	11 [‡] / ₁₂	1 [†]	1	14 [‡] / ₉	5 [‡] / ₃	7 [†] / ₄	34 [‡] / ₁₉	
8	-	-	-	-	-	-	-	2 [†] / ₃	27 [‡] / ₃₇	4 [†] / ₅	5 [‡] / ₆	11 [†] / ₁₂	13 [‡] / ₁₄	1 [†]	1	14 [†] / ₉	8 [‡] / ₅	
9	-	-	-	-	-	-	-	-	9 [‡] / ₁₄	27 [†] / ₃₇	3 [‡] / ₄	5 [†] / ₆	6 [‡] / ₇	13 [†] / ₁₄	15 [‡] / ₁₆	1 [†]	1	
10	-	-	-	-	-	-	-	-	-	9 [†] / ₁₄	2 [‡] / ₃	3 [†] / ₄	4 [‡] / ₅	6 [†] / ₇	7 [‡] / ₈	15 [†] / ₁₆	17 [‡] / ₁₈	
11	-	-	-	-	-	-	-	-	-	-	22 [‡] / ₃₅	2 [†] / ₃	39 [†] / ₅₃	4 [†] / ₅	75 [‡] / ₉₁	7 [†] / ₈	8 [‡] / ₉	
12	-	-	-	-	-	-	-	-	-	-	-	22 [†] / ₃₅	2 [‡] / ₃	39 [†] / ₅₃	3 [‡] / ₄	75 [†] / ₉₁	5 [‡] / ₆	
13	-	-	-	-	-	-	-	-	-	-	-	-	22 [‡] / ₃₅	2 [†] / ₃	5 [‡] / ₇	3 [†] / ₄	34 [‡] / ₄₃	
14	-	-	-	-	-	-	-	-	-	-	-	-	-	22 [†] / ₃₅	15 [‡] / ₂₃	5 [†] / ₇	17 [‡] / ₂₃	
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5 [‡] / ₈	15 [†] / ₂₃	2 [‡] / ₃	
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5 [†] / ₈	34 [‡] / ₅₃	
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	8 [‡] / ₁₃	

TABLE D.2: Values of $\bar{\alpha}(x, k)$ for $1 \leq x \leq 17$ and $2 \leq k \leq 18$. Values of $\bar{\alpha}(x, k)$ below the jagged line are for the cases where $a_1 = 1$, while the values above the line are for the cases where $a_1 > 1$. [†]By Theorem D.1(a). ^{*}By Theorem D.1(b). [‡]By Theorem D.1(c).

Appendix E

The Tabu Search Methodology

The tabu search technique (TS), proposed by F. Glover [49] in 1986, is a metaheuristic that has been applied with great success to various combinatorial optimization problems. As a local search technique, the tabu search technique explores the search space by continually moving from a current solution to a neighbouring solution through the use of a memory structure in order to optimize an objective function. This memory structure keeps track of recent decisions forcing the search algorithm to explore new areas of the search space, where non-improving solutions may be accepted in order to escape from local optima [95, p 346]. These recent decisions are classified as **tabu** (forbidden) and are avoided when making decisions about moving to a neighbouring solution, called a **candidate solution**. The system of tabu classifications prevents cycling between the same solutions [93, p 11]. All tabu moves, *i.e.* the reverse move of a decision recently made, are recorded in what is called a **tabu list**. The nature of the tabu list is problem specific and is usually implemented in the form of a **FIFO (First-In-First-Out) list**, where the addition of new elements force expulsion of the last (oldest) elements from the list in order to maintain a constant list length. The length of a tabu list, referred to as a **tabu tenure**, describes how many of the past moves should be remembered and is also problem specific [101, p 370]. The tabu tenure should be selected carefully, since small values of the tabu tenure may create cycling and larger values may not improve the procedure while increasing the computation time [64].

It is, of course, possible to encounter a situation where all non-tabu moves result in a far worse solution than the current solution, or where no non-tabu moves exist. For this reason a so-called **aspiration criterion** is incorporated into the tabu search, where the most common aspiration criterion used is to override the tabu status of a move if the resulting neighbouring solution improves on the best solution obtained thus far [95, p 347]. The reader is referred to [84, p 130] for other aspiration criteria. In contrast to the short-term **recency-based** memory structures described thus far, a long-term **frequency-based** memory structure may be implemented by the use of some frequency measure. Frequency measures are usually the number of occurrences of a particular event (*e.g.* the number of times a particular vertex is moved to a specific colour class during the search for a feasible colouring of a graph) in relation to the total number of iterations performed thus far [93, p 16]. (For other examples of frequency measures, see [93, p 16].) Frequency-based memory structures may be used to implement a so-called **diversification strategy** where new areas of the search space are explored [84, p 129]. On the other hand, during **intensification strategies** good solutions are exploited by concentrating the local search around their neighbourhoods, since neighbourhood solutions often have correlated objective function values [93, p 62].

Classical tabu search is basically deterministic and operates, therefore, without reference to randomization, except perhaps in the case of the initial solution. However, in recent adaptations of the classical tabu search method some probabilistic elements are often incorporated into the search [84, p 125].

Traditionally, tabu search techniques explore the entire neighbourhood of the current solution and evaluates all these neighbourhood solutions (candidate solutions) in order to select the next solution. If the specific problem instance is very large, this may lead to intensive computations. For this reason, modern tabu search methods, called **candidate list strategies**, exist in which case only a proportion of the candidate solutions is examined. In this case, only a subset of the candidate solutions are listed in a **can-**

didate list from which the next solution is selected [95, p 360]. Again, the length of the candidate list is problem specific. Finally, a **stopping criterion** should be defined in order to terminate the execution of the tabu search. Several stopping criteria may be implemented, including a given number of maximum iterations of the tabu search algorithm, or as soon as a specific value of the objective function has been obtained, or after a given number of iterations without any improvement on the objective function value have been executed [93, p 137]. For additional information on the tabu search technique, the reader is referred to the comprehensive text by Glover and Laguna [50].

Appendix F

Benchmark Instances for the $\Delta(d, x)$ -Colouring Algorithms

This appendix contains the graph instances used to test and compare the four $\Delta(d, x)$ -colouring algorithms described in Chapter 5. Although benchmark instances for algorithms determining the classical chromatic number of a graph G , $\chi(G)$, exist, no such benchmark instances exist, to the knowledge of the author, for algorithms determining the maximum degree chromatic number, $\chi_d^\Delta(G)$, for values of $d \geq 1$. The graphs used to test and compare the $\Delta(d, x)$ -colouring algorithms were divided into four groups, namely (1) small graphs (listed in §F.1), (2) graphs from the structure classes considered in Chapter 4 (listed in §F.2), (3) a selection of graphs from the set of benchmark graphs for proper colourings (listed in §F.3) and (4) randomly generated graphs (listed in §F.4). The adjacency matrices of all these graphs are given in Mathematica format on the CD accompanying this dissertation (see Appendix G).

F.1 Certain Graphs of Order at Most 10

Since there exist a total of 11 989 764 connected graphs of order at most 10 [94, p 4], a small and hopefully representative selection from amongst these graphs was made for the purposes of testing and comparing the algorithms of Chapter 5. All 31 pairwise non-isomorphic connected graphs of order at most five are listed in Table F.1. For reference purposes, each of the graphs in Table F.1 was named according to the corresponding graph in the graph atlas by Read and Wilson [94, p 8]. For all these graphs the corresponding Δ -chromatic sequences were determined by hand and listed in the relevant columns in Table F.1.

Graph	order	Δ -chromatic sequence	Graph	order	Δ -chromatic sequence	Graph	order	Δ -chromatic sequence
G_1	1	1	G_{30}	5	2 2 2 1	G_{43}	5	3 2 2 1
G_3	2	2 1	G_{31}	5	2 2 1	G_{44}	5	2 2 2 1
G_6	3	2 2 1	G_{34}	5	3 2 2 2 1	G_{45}	5	4 2 2 2 1
G_7	3	3 2 1	G_{35}	5	3 2 2 1	G_{46}	5	3 2 2 2 1
G_{13}	4	2 2 2 1	G_{36}	5	3 2 2 1	G_{47}	5	3 2 2 2 1
G_{14}	4	2 2 1	G_{37}	5	2 2 2 1	G_{48}	5	3 2 2 1
G_{15}	4	3 2 2 1	G_{38}	5	3 2 1	G_{49}	5	4 2 2 2 1
G_{16}	4	2 2 1	G_{40}	5	3 2 2 2 1	G_{50}	5	3 3 2 2 1
G_{17}	4	3 2 2 1	G_{41}	5	3 2 2 1	G_{51}	5	4 3 2 2 1
G_{18}	4	4 2 2 1	G_{42}	5	3 2 2 2 1	G_{52}	5	5 3 2 2 1
G_{29}	5	2 2 2 2 1						

TABLE F.1: The Δ -chromatic sequences of all connected graphs of order at most 5.

For graphs of orders six, seven or eight, a selection of r -regular graphs, as listed in Table F.2, was made from the graph atlas by Read and Wilson [94]. This selection includes all seven connected 3-regular (cubic) graphs of orders six and eight named Ci in Table F.2 [94, p 127], the nine connected 4-regular (quartic) graphs of orders six, seven and eight named Qi in Table F.2 [94, p 145], the three connected 5-regular (quintic) graphs of order eight named Fi in Table F.2 [94, p 154], as well as the only connected 6-regular (sextic) graph of order eight named $X2$ in Table F.2 [94, p 156]. For all these graphs the corresponding Δ -chromatic sequences were determined by hand.

Graph	order	r	Δ -chromatic sequence	Graph	order	r	Δ -chromatic sequence
$C2$	6	3	3 2 2 1	$Q5$	8	4	3 3 2 2 1
$C3$	6	3	2 2 2 1	$Q6$	8	4	4 3 2 2 1
$C4$	8	3	3 2 2 1	$Q7$	8	4	3 2 2 2 1
$C5$	8	3	3 2 2 1	$Q8$	8	4	4 2 2 2 1
$C6$	8	3	3 2 2 1	$Q9$	8	4	4 2 2 2 1
$C7$	8	3	3 2 2 1	$Q10$	8	4	2 2 2 2 1
$C8$	8	3	2 2 2 1	$F2$	8	5	4 2 2 2 2 1
$Q2$	6	4	3 3 2 2 1	$F3$	8	5	4 3 2 2 2 1
$Q3$	7	4	4 3 2 2 1	$F4$	8	5	4 3 2 2 2 1
$Q4$	7	4	3 2 2 2 1	$X2$	8	6	4 4 2 2 2 2 1

TABLE F.2: The Δ -chromatic sequences of connected r -regular graphs of orders 6, 7 and 8.

Finally, in the category of small graphs, vertex-transitive graphs of order 10 were chosen as listed in Table F.3. In particular, the three connected 3-regular (cubic) vertex-transitive graphs of order ten named Cti in Table F.3 [94, p 161] and the three connected 4-regular (quartic) vertex-transitive graphs of order ten named Qti in Table F.3 [94, p 161], were selected. Once again, the Δ -chromatic sequences of these graphs were determined by hand.

Graph	order	r	Δ -chromatic sequence	Graph	order	r	Δ -chromatic sequence
$Ct6$	10	3	3 2 2 1	$Qt10$	10	4	4 3 2 2 1
$Ct7$	10	3	2 2 2 1	$Qt11$	10	4	2 2 2 2 1
$Ct8$	10	3	3 2 2 1	$Qt12$	10	4	3 3 2 2 1

TABLE F.3: The Δ -chromatic sequences of connected vertex-transitive, r -regular graphs of order 10.

F.2 Graphs from Various Structure Classes

Since it is very simple to determine the Δ -chromatic number of any tree, only a few small trees were used to test the algorithms of Chapter 5. There exist a total of 23 pairwise non-isomorphic trees of order eight [94, p 64] — these trees are listed in Table F.4. Again, the names of the trees in Table F.4

Graph	Δ -chromatic sequence	Graph	Δ -chromatic sequence	Graph	Δ -chromatic sequence
$T26$	2 2 2 2 2 2 1	$T34$	2 2 2 2 1	$T42$	2 2 2 1
$T27$	2 2 2 2 2 2 1	$T35$	2 2 2 1	$T43$	2 2 2 1
$T28$	2 2 2 2 2 1	$T36$	2 2 2 2 1	$T44$	2 2 2 1
$T29$	2 2 2 2 1	$T37$	2 2 2 2 1	$T45$	2 2 2 1
$T30$	2 2 2 2 2 1	$T38$	2 2 2 2 1	$T46$	2 2 2 1
$T31$	2 2 2 2 2 1	$T39$	2 2 2 1	$T47$	2 2 2 1
$T32$	2 2 2 2 1	$T40$	2 2 2 1	$T48$	2 2 1
$T33$	2 2 2 2 1	$T41$	2 2 2 1		

TABLE F.4: The Δ -chromatic sequences of all trees of order 8.

Graph	order	size	Upper bounds obtained by the heuristic method of §4.6
$K_{4 \times 4}$	16	96	<u>4</u> <u>4</u> <u>4</u> <u>3</u> 2 2 2 2 2 2 2 1
$K_{4 \times 7}$	28	294	<u>4</u> <u>4</u> <u>4</u> <u>4</u> <u>4</u> <u>3</u> 2 2 2 2 2 2 2 2 2 2 2 2 1
$K_{4 \times 11}$	44	726	<u>4</u> <u>4</u> <u>4</u> <u>4</u> <u>4</u> <u>4</u> <u>4</u> <u>3</u> 3 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1
$K_{7 \times 4}$	28	336	<u>7</u> <u>7</u> <u>7</u> <u>6</u> <u>4</u> <u>4</u> <u>3</u> <u>3</u> <u>3</u> 3 2 2 2 2 2 2 2 2 2 2 1
$K_{7 \times 11}$	77	2541	<u>7</u> <u>7</u> <u>7</u> <u>7</u> <u>7</u> <u>7</u> <u>6</u> <u>6</u> 5 4 4 4 4 4 4 4 <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> 2 ... (31 2's) ... 2 1
$K_{11 \times 4}$	44	880	<u>11</u> <u>11</u> <u>11</u> <u>9</u> <u>6</u> <u>6</u> <u>5</u> <u>4</u> <u>4</u> <u>4</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> <u>3</u> 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1

TABLE F.7: The selection of complete balanced multipartite graphs used to compare the results of the $\Delta(d, x)$ -colouring algorithms of Chapter 5 with the upper bounds obtained via the heuristic method of §4.6. The overlined values are upper bounds and the values without an overline are the exact values in the Δ -chromatic sequence of the specific graph listed on the left-handside.

F.3 Proper Colouring Benchmark Graphs

During the second DIMACS implementation challenge (cliques, colouring and satisfiability) in 1992–1993 [72], a number of benchmark problems for the (classical) graph colouring problem were identified. These benchmark graphs may be obtained from the DIMACS website [72] or from the website by MA Trick [106]. Unfortunately, the classical chromatic number is not known for all of these graphs. In selecting graphs from the DIMACS benchmark graphs to test the $\Delta(d, x)$ -colouring algorithms of Chapter 5, only graphs for which the classical chromatic number is known were chosen. These graphs were chosen from the following three groups:

Queen Graphs Given an $n \times n$ chessboard, a queen graph is a graph on n^2 vertices, each corresponding to a square of the board. Two vertices are adjacent if the corresponding squares are in the same row, column or diagonal [106]. The selected queen graphs are listed in Table F.8 as *queen i* , using the same names as on the DIMACS website [72].

Mycielski Graphs As mentioned in §2.3, Mycielski [87] established the existence of an x -chromatic, *triangle-free* graph for every positive integer x . From a triangle-free graph G , a **Mycielski construction**, as illustrated in Figure F.1, produces a triangle-free graph G' containing G . Let $V(G) = \{v_1, \dots, v_n\}$. Then $V(G') = V(G) \cup U \cup \{w\}$, where $U = \{u_1, \dots, u_n\}$. To conclude the construction, $E(G') = E(G) \cup \{u_i v_j \mid v_j \in N_G(v_i) \text{ for all } i = 1, \dots, n\} \cup \{u_i w \mid u_i \in U \text{ for all } i = 1, \dots, n\}$ [112, p 205]. The selected Mycielski graphs are listed in Table F.8 as *myciel i* , using the same names as on the DIMACS website [72].

Book Graphs Given a work of literature, a graph is created where each vertex represents a character in the book. Two vertices are adjacent if the corresponding characters encounter each other in the book [106]. Three book graphs were selected and listed in Table F.8 using the original names on the DIMACS website [72]. These graphs include *david* from Charles Dickens’s *David Copperfield*, *huck* from Mark Twain’s *Huckleberry Finn*, and *jean* from Victor Hugo’s *Les Misérables*.

Graph	order	size	χ	Graph	order	size	χ
queen5_5	25	160	5	huck	74	301	11
queen6_6	36	290	7	jean	80	254	10
queen7_7	49	476	7	david	87	406	11
queen8_8	64	728	9				
myciel4	11	20	4				
myciel5	23	71	5				
myciel6	47	236	6				
myciel7	95	755	7				

TABLE F.8: The selection from the DIMACS benchmark graphs for classical colouring.

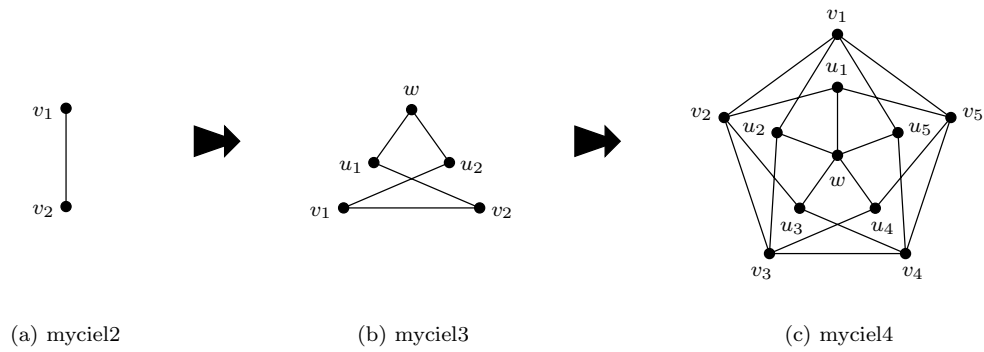


FIGURE F.1: A Mycielski construction

F.4 Random Graphs

In the case of randomly generated graphs, the trend set by Johnson *et al.* [71] was followed, where a random graph $G_{n,p}$ has n vertices and is obtained by letting a pair of vertices be adjacent with probability p , independently for each pair, *i.e.* $G_{n,p}$ has density p . The number of random graphs generated in this way are summarised in Table F.9.

density p	Number of different random instances $G_{n,p}$				
	order n of $G_{n,p}$				
	20	35	50	65	85
0.2	5	5	5	5	5
0.5	5	5	5	5	5
0.8	5	5	5	5	5

TABLE F.9: The number of different instances of each type of randomly generated graph.

Appendix G

The CD Accompanying this Dissertation

In this appendix the contents of the CD accompanying this dissertation are discussed, *i.e.* source codes of computer implementations referred to in this dissertation as well as input and output files for the various algorithmic implementations. The files are divided into two major folders, namely **DELTA ALGORITHMS** and **DELTA CHROMATIC SEQUENCES**. The contents of these two folders are described next.

- **DELTA ALGORITHMS**: This folder contains all the relevant files used in Chapter 5 ($\Delta(d, x)$ -colouring algorithms). The folder is subdivided into the following folders:
 - **SOURCE**: The MATHEMATICA source codes of the $\Delta(d, x)$ -colouring algorithms used in Chapter 5, namely **kleurgraad.nb** (the colour degree heuristic described in §5.1.1), **tabu.nb** (the tabu search $\Delta(d, x)$ -colouring heuristic described in §5.1.2), **exact.nb** (the irredundant χ_d^Δ -colouring algorithm described in §5.2.1) and **critical.nb** (the critical χ_d^Δ -colouring algorithm described in §5.2.2), are given here.
 - **INPUT**: This folder contains the benchmark instances used to test the $\Delta(d, x)$ -colouring algorithms listed in Appendix F. All these graphs are given in MATHEMATICA format. The relevant files are **SmallGInput.nb** (all the graphs of order at most 10 listed in §F.1), **TreeInput.nb** (all the trees given in Table F.4), **StrucClassInput.nb** (all the graphs from various structure classes listed in §F.2, excluding trees which are contained in a separate file), **BMarkInput.nb** (all the proper colouring benchmark instances given in Table F.8), and finally, **Random20en35Input.nb**, **Random50en65Input.nb** and **Random85Input.nb** (all the random graphs described in §F.4).
- **DELTA CHROMATIC SEQUENCES**: The relevant source code, as well as input and output files used in the JAVA program to search through the 2- and 3-partitions of certain circulant graphs as described in §6.4.3, are given in this folder. The files are subdivided into the following three folders:
 - **SOURCE**: This folder contains the JAVA source code used to search through the 2- and 3-partitions of certain circulant graphs as described in §6.4.3, namely **GraphProject.java**.
 - **INPUT**: This folder contains the input files to the JAVA program used to search through the 2- and 3-partitions of the specific circulants in the particular file. The circulants in these files either have chromatic number 4 or are those circulants for which the chromatic number could not be obtained within the set time limit by the MATHEMATICA program, as described in §6.4.3. These input files are named **xx0.txt** or **xx4.txt**, where **xx** represents the order of the circulants in the particular file; 0 and 4, respectively, indicate that the file contains

those circulants of order xx for which the chromatic number could not be obtained within the time limit by the MATHEMATICA program, and those circulants of order xx with chromatic number 4.

- **OUTPUT:** All the output files of the JAVA program used to search through the 2- and 3-partitions of the specific circulants in the particular input file are given in this folder. The first three characters in a file name are the same as the input file name. The rest of the file name indicates which filters were used to process that particular input file. After the execution of each filter was completed, an output file of the form **$xx\mathbf{y}_i.txt$** was generated, where xy is the file name of the input file and i denotes the filter that had just been executed. For example, if xx is divisible by 3, but not by 2, then Filters 1 and 2 will have been processed on the circulants in that particular input file, but not Filter 3, and the output file would characteristically be named **$xx4_1_2_4_5.txt$** .