Mixing Graph Colourings

Luis Cereceda

A thesis submitted for the degree of
Doctor of Philosophy

Department of Mathematics
London School of Economics
and Political Science

December 2007

Declaration

I certify that the thesis I have presented for examination for the MPhil/PhD degree of the London School of Economics and Political Science is solely my own work other than where I have clearly indicated that it is the work of others (in which case the extent of any work carried out jointly by me and any other person is clearly identified in it).

I warrant that this authorisation does not, to the best of my knowledge, infringe the rights of any third party.

Abstract

This thesis investigates some problems related to graph colouring, or, more precisely, graph *re*-colouring. Informally, the basic question addressed can be phrased as follows. Suppose one is given a graph G whose vertices can be properly k-coloured, for some $k \geq 2$. Is it possible to transform any k-colouring of G into any other by recolouring vertices of G one at a time, making sure a proper k-colouring of G is always maintained? If the answer is in the affirmative, G is said to be k-*mixing*. The related problem of deciding whether, given two k-colourings of G, it is possible to transform one into the other by recolouring vertices one at a time, always maintaining a proper k-colouring of G, is also considered.

These questions can be considered as having a bearing on certain mathematical and 'real-world' problems. In particular, being able to recolour any colouring of a given graph to any other colouring is a necessary pre-requisite for the method of sampling colourings known as Glauber dynamics. The results presented in this thesis may also find application in the context of frequency reassignment: given that the problem of assigning radio frequencies in a wireless communications network is often modelled as a graph colouring problem, the task of *re*-assigning frequencies in such a network can be thought of as a graph recolouring problem.

Throughout the thesis, the emphasis is on the algorithmic aspects and the computational complexity of the questions described above. In other words, how easily, in terms of computational resources used, can they be answered? Strong results are obtained for the $k = 3$ case of the first question, where a characterisation theorem for 3-mixing graphs is given. For the second question, a dichotomy theorem for the complexity of the problem is proved: the problem is solvable in polynomial time for $k \leq 3$ and PSPACE-complete for $k \geq 4$. In addition, the possible length of a shortest sequence of recolourings between two colourings is investigated, and an interesting connection between the tractability of the problem and its underlying structure is established.

Some variants of the above problems are also explored.

To my parents,

for all their support.

Contents

Acknowledgements **9**

1 Introduction **10**

1.1 Preliminaries 12

1.2 Background and motivation 16

1.3 Outline of the thesis 20

2 First results on mixing **22**

2.1 Basic properties of mixing 22

2.2 Mixing k-colourings in k-chromatic graphs 30

3 Mixing 3-colourings **37**

3.1 Characterising 3-mixing graphs 37

3.2 The complexity of 3-MIXING 42

3.3 A polynomial time algorithm for 3-MIXING for planar graphs 53

4 Paths between 3-colourings **59**

4.1 A polynomial time algorithm for 3-COLOUR PATH 60

4.2 Distances between 3-colourings 70

5 Paths between k-colourings **74**

5.1 Preliminaries 75

5.2 PSPACE-completeness of k-COLOUR PATH 79

5.3 Distances between k-colourings 84

5.4 Tractability of k-COLOUR PATH and distances between k-colourings . . . 93

6 Miscellaneous results about recolouring **96**

6.1 Recolouring using extra colours 96

6.2 The complexity of finding alternative colourings 99

7 Conclusion **104**

7.1 Related work 104

7.2 Discussion and open problems 109

Appendix **113**

Bibliography **116**

Acknowledgements

This thesis would not exist without the help and collaboration of a great number of people. I would like to acknowledge some particular contributions.

Firstly, I am truly indebted and very thankful to Jan van den Heuvel for his excellent supervision throughout the development of this thesis. He has been a constant source of inspiration and fruitful mathematical questions; his patience and encouragement have been second to none.

Secondly, I owe thanks to my coauthors—Paul Bonsma, Jan van den Heuvel and Matthew Johnson—for all their help and generosity in sharing and communicating ideas, as well as to Hajo Broersma for putting forward the seminal question that led to the development of this thesis. For other questions and remarks that led to further results I am also grateful to Bill Jackson, Mark Jerrum, Gary MacGillivray, Steve Noble, Moshe Vardi and Peter Winkler.

Thanks also to everyone at the Mathematics Department at the London School of Economics for making my time there so enjoyable. In particular, I would like to thank Graham Brightwell and Bernhard von Stengel for their interest and helpful discussions, and Jackie Everid, Simon Jolly and Dave Scott for their unstinting support in all matters non-mathematical. I would also like to acknowledge the London School of Economics itself and the Engineering and Physical Sciences Research Council for their financial support.

Lastly, and in some sense most importantly, thanks to all my friends and family for all their love and support during the past four years.

1

Introduction

Graph theory deals with the abstract study of connections between objects. It is a fundamental branch of combinatorial mathematics with a very wide range of applications. Its origin is usually attributed to Leonhard Euler's solution of the Seven Bridges of Königsberg Problem in 1735. The city of Königsberg in Prussia (now called Kaliningrad, and situated in Russia), set on the river Pregel, included two large islands connected to each other and the mainland by seven bridges. Allegedly, the residents had long asked themselves whether it was possible to tour the city crossing each bridge exactly once, ending up at the point from which one had started. Euler proved, in what is widely accepted to be the first paper in the history of graph theory [21], that no such tour is possible.

Roughly speaking, a graph is a set of vertices—which may be thought of as representing objects—and a set of edges between pairs of vertices—which may be thought of as connections between pairs of objects. As a basic way of representing the connectivity properties of a set of objects, graphs are used to model, for example, road and railway networks, components on an electrical circuit board, flows through a system of pipes, the structure of molecules, computer networks, and the Internet. In all these contexts, many interesting problems can be cast in graph-theoretic terms, and can therefore be attacked employing the tools of graph theory. But graphs do not serve just as models for physical connections between objects. Many other, more abstract problems, such as how best to timetable a set of exams, or how best to assign a set of jobs to a given set of people, can also be explored using graph-theoretic techniques.

This thesis concentrates on the area of graph theory known as graph colouring. The origins of graph colouring can be traced back to the middle of the 19th century when, in 1852, Francis Guthrie asked whether four colours are enough to colour the regions of any map drawn in the plane in such a way that regions with a common boundary receive different colours. It was not until 1976 that this question was settled in the affirmative by Appel and Haken [2, 3], though some researchers argue that their proof is not completely satisfactory. Part of Appel and Haken's proof relies heavily on the use of a computer for extensive case-analysis, and the part that is supposedly hand-checkable is still extraordinarily complicated. Some twenty years later, another, simpler and more easily verifiable proof (indeed independently verified)—though still relying on the use of a computer—was provided by Robertson, Sanders, Seymour and Thomas [56]. Ever since this first graph colouring problem was posed a century and a half ago, the subject has grown continually and is now vast. The fact that many non-mathematicians know of Guthrie's question or of the subsequent Four Colour Theorem is testament to its status.

Indeed graph colouring now occupies a central position in discrete mathematics: it has developed into an elegant theory with many applications, sometimes surfacing in unexpected areas. It deals with the basic problem of partitioning a set of objects according to certain prescribed rules or constraints. Typically, the constraints specify, for each pair of objects, whether both objects are allowed in the same class or not. Sequencing and scheduling problems are important applications which fall into this category. As a basic example, consider the following problem. Suppose we wish to construct a timetable for a set of exams, taking care to use the smallest number of time-slots as possible. This problem can be modelled as a graph colouring problem by letting each exam be represented by a vertex, and joining two exams by an edge if there is some student sitting both exams (which therefore require different time-slots). If we think of time-slots as 'colours', an assignment of colours to the vertices of the graph that gives vertices joined by an edge different colours—a colouring—yields a timetable. Hence a colouring using the minimum possible number of colours will yield the desired timetable. Another important application of graph colouring, which we will examine in some detail later in this chapter, is the task of assigning radio frequencies in a wireless communications network.

The field is still a very active area of research, and many important questions remain unresolved. An abundance of graph colouring open problems, together with detailed annotations, historical notes and references can be found in the monograph of Jensen and Toft [37].

The results in this thesis can more precisely be described as concentrating not on graph colouring, but on graph *re*-colouring. Basically, we investigate the following two problems.

1. Given a graph, is it possible to recolour any colouring of the graph to any other by recolouring vertices one at a time, always maintaining a colouring of the graph?

2. Given a graph and two colourings of the graph, is it possible to recolour one colouring to the other by recolouring vertices one at a time, always maintaining a colouring?

Our primary focus is on the algorithmic aspects of these questions. In particular, we study the computational complexity of the decision problems associated with them: that is, how easily, in terms of computational resources used, can we answer them?

We also examine related issues. For example, we provide some answers to the following questions. Can we characterise the graphs for which the answer to the first question is 'yes'? Is there anything remarkable or particular about the colourings of such graphs? For the second question, if, for a particular instance, we know that the answer is in the affirmative, how easily can we find a sequence of recolourings that achieves the transformation? How long is such a sequence, and how long is a shortest possible sequence? On the other hand, if we know the answer is in the negative, can we achieve the transformation by using relatively few extra colours? How many are actually necessary?

Before proceeding, in the rest of this chapter, to provide some motivation for studying these problems and to give an overview of the thesis, we describe our basic terminology and notation, together with some fundamental concepts and definitions.

1.1 Preliminaries

Most of our mathematical terminology and notation is standard. Let us point out some particulars. The cardinality of a set X is denoted by $|X|$, and the set-theoretic difference between X and any other set Y by $X \setminus Y$. We do not count 0 as a natural number, and, for $k \in \mathbb{R}$, $\lfloor k \rfloor$ is the largest integer less than or equal to k.

We assume familiarity with the basic concepts of graph theory and computational complexity theory. For an introduction to the former we refer the reader to any standard textbook on graph theory such as, for example, Diestel [17] or West [61]; for an introduction to the latter, see Garey and Johnson [24] or Papadimitriou [53]. The reader should also find definitions for concepts and terminology not defined here in these references. We presently revise some of the basics of graph theory and describe some particular conventions used in this thesis. Following this, we will give some definitions necessary for a precise description of our results.

Basic graph-theoretic concepts and conventions used in this thesis

We denote the set of vertices of a graph G by $V(G)$ and its set of edges by $E(G)$. When there is no danger of confusion, or the graph in question is given implicitly, we will write V and E. Throughout this thesis we consider only finite graphs with no loops or multiple edges. Thus for any graph G, V will be a finite set and E will be a set of unordered pairs of elements of V, where elements in any given pair are distinct. We will often write n for $|V|$ and m for $|E|$. For simplicity and ease of reference, we will also often deliberately confuse a graph with its set of vertices.

We denote an edge between vertices u and v by uv (or, equivalently, by vu), saying vertices u and v are *adjacent*, or *neighbours*, and that the vertex u is *incident* with, or an *end-vertex* of, the edge uv. We write $d(v)$ for the *degree* of v, which is the number of edges incident with v. If we need to distinguish the graph G in which the degree is measured, we will write $d_G(v)$. The maximum and minimum degree of G are respectively denoted by $\Delta(G)$ and $\delta(G)$. If it is clear from the context which graph is under consideration, we will simply write Δ or δ.

A *path* between vertices u and v—a (u, v)-path—is a sequence of distinct vertices, starting at u and ending at v, such that pairs of consecutive vertices in the sequence constitute an edge of the graph. The *distance* between vertices v and w, denoted $d(v, w)$, is the number of edges in a shortest path between v and w; if there is no path between v and w we say that the distance between them is infinite. If we need to distinguish the graph G in which the distance is measured, we will write $d_G(v, w)$. The *diameter* of G, $\mathrm{diam}(G)$, is defined as $\max\{d(u, v) \mid u, v \in V\}$.

For a subset X of V, we denote by $G - X$ the graph that has $V \setminus X$ as its vertex set and whose edges are the edges of G that have both end-vertices in $V \setminus X$. For X a subset of E, $G - X$ denotes the graph with vertex set V and edge set $E \setminus X$.

A graph G is said to be *connected* if any two of its vertices are linked by path. It is *k-connected* if it has at least $k + 1$ vertices and for every set $X \subset V$ with $|X| < k$, $G - X$ is connected.

If H is a subgraph of G, we write $H \subseteq G$. The *degeneracy* of G, $\deg(G)$, is defined as the largest minimum degree of any subgraph of G. That is, $\deg(G) = \max\{\delta(H) \mid H \subseteq G\}$. This quantity is also known as the *colouring number* or *maximin degree* of G. It is an easy exercise to verify that a graph has degeneracy r if and only if there is an ordering $v_1, v_2, \ldots, v_n$ of its vertices such that for $1 \leq i \leq n$, the vertex v_i has at most r neighbours v_j with $j < i$. Such a graph is described as *r-degenerate*, as is any associated vertex-ordering.

A *drawing* of a graph G on a surface $\mathcal{S}$ (that is, a compact 2-dimensional manifold without boundary) is a graphical representation of G on $\mathcal{S}$, with each vertex assigned a distinct point on $\mathcal{S}$, and curves joining points which correspond to vertices forming an edge. The drawing is said to be an *embedding* if no two curves intersect (other than at vertex points), and G is *embeddable* on $\mathcal{S}$ if there exists an embedding of G on $\mathcal{S}$. If a graph is embeddable on the sphere it is said to be *planar*, since the plane is homeomorphic to a sphere with a point removed.

Two graphs G_1 and G_2 are said to be isomorphic, written $G_1 \cong G_2$, if there exists a bijection $\varphi : V(G_1) \to V(G_2)$ such that $uv \in E(G_1)$ if and only if $\varphi(u)\varphi(v) \in E(G_2)$.

We denote the cycle on n vertices (or n-cycle) by C_n, and the complete graph on n vertices by K_n. The graph $C_3 \cong K_3$ is known as the *triangle*. Quite often we will describe a cycle C_n by just listing its vertices $v_1, v_2, \ldots, v_n$, with the edges $v_1v_2, v_2v_3, \ldots, v_{n-1}v_n, v_nv_1$ being read implicitly.

Colourings and recolouring: the colour graph

In this section we recall some basic definitions about colouring, and formalise our notions about recolouring graph colourings.

All colourings considered in this thesis are proper vertex colourings. That is, for a natural number $k \geq 2$, we define a *k-colouring* of a graph G as a function $\alpha : V \to \{1, 2, \ldots, k\}$ such that $\alpha(u) \neq \alpha(v)$ for all $uv \in E$. If G has a k-colouring, we say it is *k-colourable*. (We insist that $k \geq 2$ in order to avoid trivialities—there is not much to say about 1-colourable graphs.) For $1 \leq i \leq k$, the preimages $\alpha^{-1}(i)$ are termed *colour classes*. The smallest k for which a k-colouring of G exists is called the *chromatic number* of G, $\chi(G)$. We will generally use lower case Greek letters $\alpha, \beta, \ldots$ to denote specific colourings, and we will often describe a k-colouring of a path or cycle by just listing the colours as they appear on consecutive vertices.

Definition 1.1
Let G be a k-colourable graph. The *k-colour graph* of G, denoted $\mathcal{C}_k(G)$, is the graph that has the k-colourings of G as its vertex set, with two k-colourings joined by an edge in $\mathcal{C}_k(G)$ if they differ in colour on precisely one vertex of G. If $\mathcal{C}_k(G)$ is connected, we say that G is *k-mixing*.

The colour graph allows us to talk about recolourings and possible sequences of recolourings in a graph-theoretic language: we may now meaningfully talk of adjacency, paths and distances between colourings.

A k-colouring of G that forms an isolated node in $\mathcal{C}_k(G)$ is said to be *frozen*. Note that the existence of a frozen k-colouring of a graph immediately implies that the graph is not k-mixing. If G has a k-colouring α, then we say that we can, from α, *recolour* G *with* β if $\alpha\beta$ is an edge of $\mathcal{C}_k(G)$. If v is the unique vertex on which α and β differ, then we also say that we can *recolour* v. Given a k-colouring α, a colour is *available* for a vertex v if neither v nor any of its neighbours are assigned that colour. If there is a path between α and β in $\mathcal{C}_k(G)$ we will say that we can *recolour* α *to* β.

We will sometimes describe recolourings explicitly, and sometimes implicitly. In either case, it will often be useful to think of a sequence of recolourings as a list of ordered pairs (v, c) where, at any stage in the sequence, v is the vertex to be recoloured with colour c.

Decision problems about recolouring

We are now in a position to formally state the decision problems corresponding to the two questions stated at the beginning of this introduction, and whose computational complexity will be the central question addressed in this thesis.

Corresponding to the first question (given a graph, is it possible to recolour any colouring of the graph to any other by recolouring vertices one at a time, always maintaining a colouring of the graph?) we have the problem k-MIXING.

k-MIXING
Instance: A connected graph G.
Question: Is G k-mixing? That is, is $\mathcal{C}_k(G)$ connected?

Corresponding to the second question (given a graph and two colourings of the graph, is it possible to recolour one colouring to the other by recolouring vertices one at a time, always maintaining a colouring?) we have the problem k-COLOUR PATH.

k-COLOUR PATH
Instance: A connected graph G together with two k-colourings of G, α and β.
Question: Is there a path between α and β in $\mathcal{C}_k(G)$?

Note that k is never part of the input. In other words, we have two *classes* of problems, each consisting of an infinite number of problems parametrised by k. Note also that we always insist that the instance graph G is connected. If G is not connected, then it is easy to see that G is k-mixing if and only if H is k-mixing for every connected component H

of G. Similarly, there is a path between k-colourings α and β of G if and only if, for every connected component H of G, there is a path between the colourings induced by α and β on H. Thus we may always reduce our problems to connected graphs, and will therefore, as a general rule, take graphs to be connected.

1.2 Background and motivation

Our main motivation for studying the problems described above is for their own sake. The questions are simple and fairly natural, and lead to, in the author's opinion, some interesting mathematics. However, the questions can certainly be regarded as being motivated by other lines of research, or indeed as having applications. We proceed to outline two such motivating applications.

Sampling colourings via Glauber dynamics

The question of when the k-colour graph is connected is not new. It has been looked at, as a subsidiary issue, by researchers in the statistical physics community studying the Glauber dynamics of an anti-ferromagnetic Potts model at zero temperature. Associated with that research is the work on rapid mixing of Markov chains used to obtain efficient algorithms for almost uniform sampling of k-colourings of a given graph. We give a brief description of the basic ideas involved in these areas of research.

Randomness plays an important role in many parts of combinatorics and theoretical computer science. Indeed results from probability theory have led to major developments in both fields. It is therefore unsurprising that researchers are often interested in obtaining random samples of particular combinatorial structures. For example, much attention has been devoted to the problem of sampling from an exponential number of structures (exponential in the size of the object over which the structures are defined) in time polynomial in this quantity. One of the reasons for this is that being able to sample almost uniformly from a set of combinatorial structures is enough to be able to approximately count such structures—see [38] for an example illustrating the method in the context of graph colourings, and [39] for full details.

Quite often, the sampling is done via the simulation of an appropriately defined Markov chain. Here the important point is that the Markov chain should be *rapidly mixing*. This means, loosely speaking, that it should converge to a close approximation of the stationary distribution in time polynomial in the size of the problem instance. For a precise description of this concept and further details we refer the reader to [39].

In the context of the particular Markov Chain used for sampling k-colourings of a graph known as *Glauber dynamics*—originally defined for the anti-ferromagnetic Potts model at zero temperature (see below)—we have the following. For a particular graph G and value of k, let us denote the Glauber dynamics for the k-colourings of G by $\mathcal{M}_k(G) = (X_t)_{t=0}^{\infty}$. The state space of $\mathcal{M}_k(G)$ is the set of k-colourings of G, the initial state X_0 is an arbitrary colouring, and its transition probabilities are determined by the following procedure.

1. Select a vertex v of G uniformly at random.
2. Select a colour $c \in \{1, 2, \ldots, k\}$ uniformly at random.
3. If recolouring vertex v with colour c yields a proper colouring, then set X_{t+1} to be this new colouring. Otherwise, set $X_{t+1} = X_t$.

The relation of $\mathcal{M}_k(G)$ to the k-colour graph of G should be obvious: a simulation of the chain corresponds to a walk in $\mathcal{C}_k(G)$, since two k-colourings α, β of G form an edge of $\mathcal{C}_k(G)$ if and only if $\Pr(X_{t+1} = \beta \,|\, X_t = \alpha) > 0$, in which case

$$\Pr(X_{t+1} = \beta \,|\, X_t = \alpha) = \frac{1}{k|V|}.$$

Clearly $\mathcal{M}_k(G)$ is irreducible if and only if G is k-mixing. Thus the fact that a graph is k-mixing is a necessary condition for its Glauber dynamics Markov chain to be rapidly mixing. (This should go some way to explaining our choice of terminology for describing a graph with a connected k-colour graph!) Let us remark, however, that a graph being k-mixing is not sufficient for its Glauber dynamics Markov chain to be rapidly mixing. An example showing this is given by the stars $K_{1,m}$, which are k-mixing for any $k \geq 3$ (see Theorem 2.7 in Section 2.1) but whose Glauber dynamics is not rapidly mixing for $k \leq m^{1-\varepsilon}$, for fixed $\varepsilon > 0$ (proved in [44]).

We turn to a brief and informal description of the *Potts model*. This is a statistical mechanics model for studying the interaction of spins (intrinsic angular momenta) of the particles in a crystalline lattice. It is used as a theoretical description for ferromagnetism and other phenomena of solid-sate physics. In the ferromagnetic case, like spins of neighbouring particles are encouraged by a certain lowering of the total energy of the system for every neighbouring pair with like spins. In the anti-ferromagnetic case, neighbouring particles are encouraged to have different spins. The temperature of the system reflects the extent of 'encouragement': the lower the temperature, the more the energy of the system is lowered by a given neighbouring pair of particles having like/unlike spins. At zero temperature, this encouragement becomes an inviolable requirement. Thus the zero-temperature anti-ferromagnetic k-state Potts model of a particular lattice—where each particle has one of k possible spins, and neighbouring particles cannot have the same spin—has as its set of configurations the set of k-colourings of the graph corresponding to the lattice. The Glauber dynamics of this model describes the transitions between the spin states of the system in precisely the same manner as described above.

Let us point out that much of the work on rapid mixing of the Glauber dynamics Markov chain (as well as that of its many generalisations and variants) has concentrated on specific graphs, or on values of k so large that the connectedness of the k-colour graph is guaranteed. In particular, because of its focus on crystalline structures, the Potts model has been widely studied on very regular and highly symmetric graphs such as integer grids. In contrast, we address the question of the irreducibility of the chain in a wider sense, asking what can be said in general, for any graph and relatively small values of k.

Radio frequency reassignment

Besides its use in sequencing and scheduling, another important application of graph colouring is that of modelling the assignment of frequencies in radio-communication networks. The basic aim of the Frequency Assignment Problem (FAP) is to assign frequencies to users of a wireless network, minimising the interference between them and taking care to use the smallest possible range of frequencies. Because the radio spectrum is a naturally limited resource with a constantly growing demand for the services that rely on it, it has become increasingly important to use it as efficiently as possible. As a result, and because of the inherent difficulty of the problem, the subject of frequency assignment is huge. For an introduction and survey of different approaches and results we refer the reader to [43] and [48].

The FAP was first considered as a graph colouring problem by Hale in [28]. In this setting, we think of the available frequencies (discretised and appropriately spaced in the spectrum) as colours, transmitters as vertices of a graph, and we add edges between transmitters that must be assigned different frequencies. In order to better capture the subtleties of the 'real-world' problem, this basic model has been generalised in a multitude of different ways. Typically this might involve taking into account the fact that radio waves decay with distance obeying an inverse-square law. For instance, numerical weights can be placed on the edges of the graph to indicate that frequencies assigned to the end-vertices of an edge must differ by the amount given by the particular edge-weight. Another example is provided by the well-known $L(2,1)$-labelling problem: this asks for the smallest k such that the vertices of a graph can be labelled with values from the set $\{1, 2, \ldots, k\}$ in a way such that labels on adjacent vertices differ by at least 2, and labels on vertices at distance two differ by at least 1. See [36] for a survey of graph colouring and labelling techniques applied to the FAP.

One of the the major factors contributing to the growth in demand for use of the radio spectrum has been the dramatic increase, in recent years, of mobile telecommunication systems. In such systems, where new transmitters are continually added to meet increases in demand, an optimal or near-optimal assignment of frequencies will in general not remain so for long. On the other hand, it might just be the case that, because of the difficulty of finding optimal assignments, a sub-optimal assignment is to be replaced with a recently-found better one. It thus becomes necessary to think of the assignment of frequencies as a dynamic process, where one assignment is to be replaced with another. In order to avoid interruptions to the running of the system, it is desirable to avoid a complete re-setting of the frequencies used on the whole network. In a graph colouring framework, this leads naturally to our problems.

Not much attention seems yet to have been devoted to the problem of reassigning frequencies in a network. Some first results can be found in [4, 6, 29, 47]. The work in [4, 6, 29] describes some specific heuristic approaches to the problem, as well as some associated computational simulations.

A more general approach, with a theoretical bent which gives rise to problems similar to the ones we study, can be found in [47]. Here the authors describe a problem they call *colour switching*: given a graph G and two proper vertex colourings of G, the colour switching problem asks for a sequence of vertex recolourings that transforms the first colouring into the second, with all intermediate colourings being proper. This looks remarkably similar to the problem k-COLOUR PATH, but is quite different. Firstly, colour switching is a combinatorial problem (it asks for a sequence of recolourings) while k-COLOUR PATH is a decision problem (it asks for a yes or no answer). Thus colour switching is always possible (by using enough extra colours), while the question in k-COLOUR PATH might well be answered in the negative. This is because of the more important fundamental difference between the problems: for k-COLOUR PATH we insist that all colourings considered (both the input colourings as well as all intermediate colourings) are k-colourings, while in colour switching no such restriction is imposed.

A tight bound on the minimum number of extra colours necessary to guarantee one can always find a solution to colour switching is given in [47]. We have obtained the same result independently, with very similar examples illustrating tightness. We will describe this result in Chapter 6. The authors of [47] also consider the question of finding bounds on the number of recolourings necessary to transform one colouring into another when the number of extra colours used is minimal, or nearly so. Indeed this is an issue we have also addressed (for k-COLOUR PATH, where no extra colours are allowed), and to which we devote our attention in the latter sections of Chapters 4 and 5.

1.3 Outline of the thesis

In Chapter 2 we prove some basic results about k-colour graphs and the k-mixing properties of graphs. We first look for values of k that ensure a graph will be k-mixing, considering possible bounds in terms of the chromatic number and the degeneracy. We also examine the case $k = \chi(G)$, showing that if $k = \chi(G)$ is 2 or 3, then G is not k-mixing. On the other hand, we show that for all $k \geq 4$ there are k-chromatic graphs that are k-mixing, and k-chromatic graphs that are not k-mixing.

Chapter 3 addresses the computational complexity of deciding whether a given graph is 3-mixing. Given that 3-chromatic graphs are never 3-mixing, we focus our attention on bipartite graphs. We give two equivalent characterisations of 3-mixing bipartite graphs and prove that deciding if a given bipartite graph is 3-mixing is coNP-complete. We also prove that for *planar* bipartite graphs the problem is decidable in polynomial time.

Chapter 4 gives a polynomial time algorithm for 3-COLOUR PATH. The algorithm can be used to exhibit a path between the two 3-colourings, if this exists. It also allows us to deduce that the connected components of $\mathcal{C}_3(G)$ always have diameter at most quadratic in the size of the graph.

In Chapter 5 we examine the complexity of k-COLOUR PATH for values of $k \geq 4$, proving that in this regime the problem is PSPACE-complete. We also show, by means of explicit construction, that in these cases the distance between colourings can be superpolynomial in the size of the graph.

Chapter 6 describes some miscellaneous results. In particular, we provide an answer to the following question: given any graph G together with two k-colourings, what is the least number of extra colours necessary to guarantee that it is possible to recolour the first k-colouring to the second? We show that the answer to this question is $\chi(G) - 1$. We also examine the complexity of finding some particular types of k-colouring of a given k-colourable graph.

We close in Chapter 7 with a discussion of our work. We also describe related work and mention some possibilities for further research.

Most of the work presented in this thesis is the result of joint work. Some parts of it have been published and other parts are in the process of being accepted for publication. The main results of Chapter 2 are to be found in [10]; Chapter 3 corresponds entirely to [11], Chapter 4 to [12], and Chapter 5 to [7]. The results of Section 6.1 are also joint work with Jan van den Heuvel.

Note

It has recently come to the author's attention that very similar results to those presented in Chapter 5 appear in [35], a dissertation which is otherwise unpublished. In that thesis, the problem of deciding whether two given colourings of a graph are connected, where the number of colours k is part of the input, is proved to be PSPACE-complete. The reduction in [35] also proves the existence of graphs with colourings at superpolynomial distance, but not by means of any explicit construction. The result showing that $\chi(G)-1$ extra colours are always enough to recolour a given k-colouring of a graph to a second given k-colouring of the graph, and that this bound is best possible- presented in Chapter 6, and proved independently in [47]—can also to be found in [35]. We will make a comparative study of all these results in Chapter 7.

2

First results on mixing

In this chapter we prove some first results about the mixing properties of graphs. After making some preliminary observations, we look for values of k that will ensure a graph is k-mixing; we consider possible bounds in terms of various graph invariants including the chromatic number and the degeneracy of a graph. We also study this question for graphs embeddable on a particular surface. In Section 2.2 we examine the case $k = \chi(G)$, showing that if G is a graph with chromatic number $k \in \{2, 3\}$, then G is not k-mixing. On the other hand, we prove that for all $k \geq 4$ there exist k-chromatic graphs that are k-mixing, as well as k-chromatic graphs that are not k-mixing.

2.1 Basic properties of mixing

The k-colour graph of a given graph G is a complex structure containing much information about G. Indeed it turns out that for $k > \chi(G)$ it actually determines G, in the sense that non-isomorphic χ-chromatic graphs have non-isomorphic k-colour graphs, as long as $k > \chi$, [32]. Clearly if G is not k-colourable, $\mathcal{C}_k(G)$ is just the null graph (though note that, strictly speaking, we have defined the k-colour graph only for k-colourable graphs). In general, $\mathcal{C}_k(G)$ will have exponential size, with $|V(\mathcal{C}_k(G))| = P_G(k)$, where P_G is the chromatic polynomial of G.

Let us make some simple observations. Notice that for any k and any graph G, $\mathcal{C}_k(G)$ is an induced subgraph of $\mathcal{C}_{k+1}(G)$, since a k-colouring of G can be regarded as a (non-surjective) $(k+1)$-colouring, and any possible recolouring in $\mathcal{C}_k(G)$ is also possible in $\mathcal{C}_{k+1}(G)$.

Let $Q_p(k)$ be the generalised p-dimensional cube. This graph has vertex set $\{1, 2, \ldots, k\}^p$, the set of all sequences of length p with entries from $\{1, 2, \ldots, k\}$, and an edge between any two sequences that differ in precisely one entry. If we denote the empty graph on n vertices (that is, the graph consisting of n isolated vertices) by U_n, we have $\mathcal{C}_k(U_n) \cong Q_n(k)$. On the other hand, for the complete graph on n vertices, K_n, we have $\mathcal{C}_n(K_n) \cong U_{n!}$.

In what follows we investigate the relationship between the mixing properties of a graph and two of the most important graph invariants relating to colouring: the chromatic number and the degeneracy of a graph. This leads naturally to the exploration of the mixing properties of a graph embeddable on a certain surface.

Mixing and chromatic number

Let us briefly consider the 2-mixing properties of 2-chromatic graphs. A connected 2-chromatic graph has exactly two frozen 2-colourings, so its 2-colour graph consists of two isolated vertices. If G is a disconnected 2-chromatic graph (so G is bipartite and contains at least one edge), then there is a path between a pair of 2-colourings of G if and only if the colourings agree on every connected component that contains more than one vertex. It is an easy exercise to show that if such a G has p isolated vertices and q other connected components, then $\mathcal{C}_2(G)$ has 2^q connected components, each of which is isomorphic to the p-dimensional cube $Q_p(2)$. To see this, observe that from any given 2-colouring of G, only isolated vertices may be recoloured, and that they may be recoloured freely (by which we mean that any isolated vertex may be recoloured at any time). Thus the set $\{1,2\}^p$ can be thought of as representing the 2^p possible colourings of these p isolated vertices, with adjacent colourings differing in precisely one entry. Because each of the other q connected components has two possible 2-colourings (which are frozen), we see that $\mathcal{C}_2(G)$ consists of 2^q disjoint copies of $Q_p(2)$. In any case, whether G is connected or not, we have the following result.

Proposition 2.1

Let G be a graph with chromatic number 2. Then G is not 2-mixing.

Note that these observations immediately render the decision problems 2-MIXING and 2-COLOUR PATH trivial. We will examine the k-mixing properties of k-chromatic graphs for $k \geq 3$ later in this chapter, in Section 2.2.

At first one might expect that if k is sufficiently large compared with the chromatic number of a graph, then the graph will be k-mixing. We now show that no such result is possible.

For $m \geq 3$, let L_m be the graph obtained from the balanced complete bipartite graph $K_{m,m}$ by removing the edges of a perfect matching in $K_{m,m}$. More formally, we have the

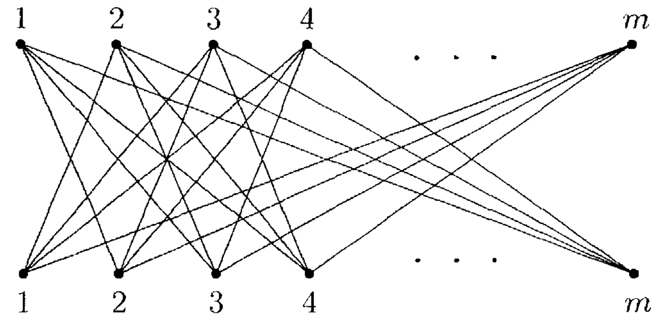

Figure 2.1 The graph L_m together with a frozen m-colouring.

following definition.

Definition 2.2
Let $m \geq 3$. The graph L_m has

- vertex set $V(L_m) = \{v_1, v_2, \ldots, v_m, w_1, w_2, \ldots, w_m\}$, and
- edge set $E(L_m) = \{v_i w_j \mid 1 \leq i, j \leq m,\ i \neq j\}$.

Note that L_m is 2-chromatic. Since $m \geq 3$, it is obvious that there are many ways to colour L_m with m colours. But suppose that we colour the vertices in each part of the bipartition of L_m with the colours $1, 2, \ldots, m$, where vertices in opposite parts that were originally connected by an edge from the removed perfect matching are given the same colour. For example, we could set $\kappa(v_i) = \kappa(w_i) = i$, for $1 \leq i \leq m$. The graph L_m together with this m-colouring is shown in Figure 2.1. This m-colouring is clearly an isolated node in the k-colour graph $\mathcal{C}_m(L_m)$, and so L_m is not m-mixing. This proves the following.

Proposition 2.3
There is no expression $\varphi(\chi)$ in terms of the chromatic number χ, so that for all graphs G and integers $k \geq \varphi(\chi(G))$, G is k-mixing.

It is interesting and worth observing that the graphs L_m are mixing for all other values of $k \geq 3$.

Proposition 2.4

For any fixed $m \geq 3$, the graph L_m is k-mixing if and only if $k \geq 3$ and $k \neq m$.

Proof. We have observed that L_m is not m-mixing; because it is 2-chromatic, neither is it 2-mixing. We show that for all other k it is k-mixing, distinguishing the cases $k < m$ and $k > m$.

Let L_m have vertex bipartition $\{X, Y\}$ and consider a k-colouring of L_m with $3 \leq k \leq m - 1$. Since X contains m vertices, there is at least one colour c_1 that appears on more than one vertex of X. But this means that no vertex in Y is coloured with c_1. Hence it is possible to recolour all vertices in X with c_1. Once this is done, we can choose a second colour $c_2 \neq c_1$ and recolour every vertex in Y with c_2. Thus we have shown that any k-colouring of L_m is connected to some 2-colouring of L_m. It is an easy exercise to show that if $k \geq 3$, all 2-colourings of L_m are connected in $\mathcal{C}_k(L_m)$. This can be seen by observing that if it is not possible to directly recolour a given 2-colouring of L_m to another distinct 2-colouring by recolouring all vertices in one part of the bipartition to their required colour, followed by recolouring all vertices in the other part—then this must be because the two 2-colourings use the same two colours. But then recolouring all vertices in X, say, with a third colour (possible since $k \geq 3$) allows us to recolour all vertices in Y to their target colour and finally reach the target 2-colouring by recolouring all vertices in X. This proves that $\mathcal{C}_k(L_m)$ is connected for $3 \leq k \leq m - 1$.

If we colour L_m with $k \geq m + 1$ colours, then again we have that a certain colour is not used on Y. By a similar argument to that in the case above, it follows that $\mathcal{C}_k(L_m)$ is connected for $k \geq m + 1$. □

Proposition 2.4 also allows us to deduce that, unlike colouring, mixing is not a monotone property; a fact which might seem, at first glance, a little surprising.

Proposition 2.5

There exist graphs G for which there exist numbers $k_1 < k_2$ such that G is k_1-mixing but not k_2-mixing.

Even though a particular graph G may not be k-mixing for k arbitrarily larger than its chromatic number, it is obvious that there always exists a value k' such that G is guaranteed to be k-mixing for all $k \geq k'$. We can take $k' = |V(G)| + 1$, for example. A better bound on such a value k' is to be found via the maximum degree of G, a fact first observed by Jerrum [38] in the context of sampling colourings via Glauber dynamics.

Proposition 2.6 (Jerrum [38])
For any graph G and integer $k \geq \Delta(G) + 2$, G is k-mixing.

We omit the proof of this proposition due to its similarity to that of Theorem 2.7 below, which in fact refines the result. Observe that the bound on k is best possible: the complete graphs K_n, which have maximum degree $n-1$, are not n-mixing since every n-colouring is a frozen colouring. Similarly, the graphs L_m have maximum degree $m-1$ but are not m-mixing.

Mixing and degeneracy

The degeneracy is a particularly useful invariant for studying the colouring properties of a graph. We will find it is also highly relevant to the mixing properties of a graph.

Let us recall that a graph G with degeneracy r can always be coloured with at most $r+1$ colours. Such a colouring can easily be found by following an r-degenerate ordering of the vertices of G, colouring each successive vertex with the first available colour (that is, the lowest colour not appearing on any of the neighbours of the vertex to be coloured). At most $r+1$ colours will be necessary because, at any stage in the process, a vertex to be coloured will have at most r neighbours that have already been coloured.

In contrast with the chromatic number, we find that if k is sufficiently large compared with the degeneracy of a graph, then the graph will be k-mixing. The following result is proved in [20] as a lemma leading to a further result on the colouring of random graphs. We give a proof for completeness.

Theorem 2.7 (Dyer, Flaxman, Frieze and Vigoda [20])
For any graph G and integer $k \geq \deg(G) + 2$, G is k-mixing.

Proof. We use induction on the number of vertices of G. The result is obviously true for the graph with one vertex, so suppose G has two or more vertices. Let v be a vertex with degree $d_G(v) \leq \deg(G)$, and set $G' = G - \{v\}$. Note that $\deg(G') \leq \deg(G)$, hence we also have $k \geq \deg(G') + 2$. By induction we can assume that $\mathcal{C}_k(G')$ is connected.

Take two k-colourings α and β of G, and let α', β' be the k-colourings of G' induced by α, β. Since $\mathcal{C}_k(G')$ is connected, there exists a sequence $\alpha' = \gamma'_0, \gamma'_1, \ldots, \gamma'_N = \beta'$ of k-colourings of G' so that for $i = 1, \ldots, N$, γ'_{i-1} and γ'_i differ in the colour of exactly one vertex of G'. Denote this vertex by v_i and denote the new colour $\gamma'_i(v_i)$ by c_i. We now try to take the same recolouring steps to recolour G, starting from α. If for some i it is not possible to recolour vertex v_i, this must be because v_i is adjacent to v and v at that moment has colour c_i. But because v has degree at most $\deg(G) \leq k-2$, there is a colour $c \neq c_i$ that does not appear on any of the neighbours of v. Hence we can first recolour v to c, then recolour v_i to c_i and continue.

In this way we find a sequence of k-colourings of G, starting at α, and ending in a colouring in which all the vertices except possibly v will have the same colour as in β. But then, if necessary, we can also recolour v to give it the colour from β. This gives a path between α and β in $\mathcal{C}_k(G)$, completing the proof. □

Since for any graph G, $\deg(G) \leq \Delta(G)$, Theorem 2.7 immediately refines Proposition 2.6. There are many graphs that show the bound in Theorem 2.7 is best possible. For example, the graphs L_m have degeneracy $m-1$ and are not m-mixing, and the graphs K_n have degeneracy $n-1$ and are not n-mixing.

We mention that the best known lower bound on the number of colours needed for rapid mixing is $\frac{11}{6}\,\Delta(G)$, proved by Vigoda [59]. We also observe that the expression in terms of the degeneracy that guarantees mixing cannot guarantee rapid mixing of the Glauber dynamics Markov chain. For instance, the stars $K_{1,m}$ have degeneracy $\deg(K_{1,m}) = 1$, but it is shown in [44] that the Glauber dynamics Markov chain for these graphs is not rapidly mixing for $k \leq m^{1-\varepsilon}$, for fixed $\varepsilon > 0$.

Mixing on surfaces

We examine what can be said about the mixing properties of a graph if we know it is embeddable on a certain surface.

Let us begin by recalling some basic definitions and facts about surfaces. For concepts not defined here, as well as for a thorough exploration of the topic of embeddings of graphs on surfaces, we refer the reader to the monograph by Mohar and Thomassen [52]. A *surface* $\mathcal{S}$ is a compact 2-dimensional manifold without boundary. Every surface is either homeomorphic to an *orientable surface* S_g of genus $g \geq 0$ or to a *non-orientable surface* N_g of non-orientable genus $g \geq 1$. The *genus* of the surface S_g can be considered as the number of handles added to a sphere, and the *non-orientable genus* of the surface N_g as the number of cross-caps added to a sphere. Thus the surface S_0 is the sphere and S_1 is the torus; N_1 is the projective plane and N_2 is the Klein bottle. The *Euler genus* $\varepsilon(\mathcal{S})$ of $\mathcal{S} = S_g$ is $2g$ and that of $\mathcal{S} = N_g$ is g. The Euler genus of a surface, together with its orientability, determine the surface up to homeomorphism.

We examine first the case of the sphere. Let G be a planar graph with n vertices, m edges and f faces. Euler's formula asserts that $n - m + f = 2$. One can easily deduce from

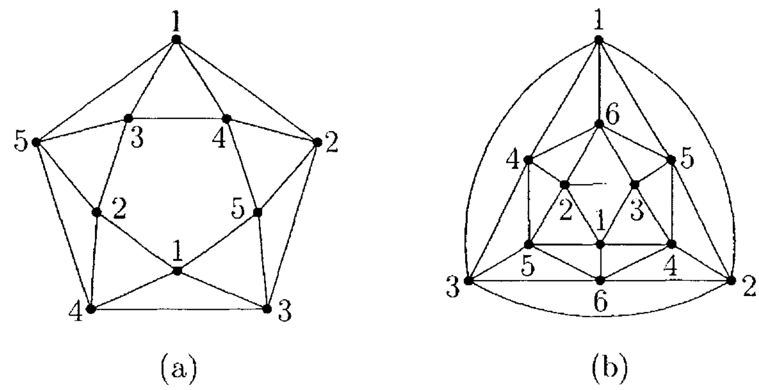

Figure 2.2 (a) a planar graph with a frozen 5-colouring, and (b) a planar graph with a frozen 6-colouring.

this that G must contain a vertex of degree at most 5 (a planar graph has at most $3n-6$ edges, so its average degree, $\frac{2m}{n}$, is strictly less than 6), and from this it follows that $\deg(G) \leq 5$. Theorem 2.7 then tells us that for any $k \geq 7$, a planar graph is k-mixing. This bound is tight: for every $k \leq 6$, there exists a planar graph that is not k-mixing. In fact, a stronger statement is true: for every $k \leq 6$, there exists a planar graph with a frozen k-colouring. For $k \leq 4$, this follows trivially from the fact that the complete graphs K_2, K_3 and K_4 are planar. For $k = 5$ and $k = 6$ we need to look harder: K_5 and K_6 are not planar, and neither are the graphs L_5 and L_6, other graphs which we have observed to have frozen 5- and 6-colourings. Examples of the required graphs and colourings are shown in Figure 2.2 (the graph in (b) is actually the icosahedron). This means we have a sharp threshold in the value of k which guarantees that any given planar graph will be k-mixing.

Given a surface $\mathcal{S}$, let us define the *mixing number* of $\mathcal{S}$ as the smallest integer $\mu(\mathcal{S})$ such that for any graph G embeddable on $\mathcal{S}$ and any $k \geq \mu(\mathcal{S})$, G is guaranteed to be k-mixing. Thus we have just seen that the mixing number of the sphere is $\mu(\mathcal{S}_0) = 7$. Can we say anything about the mixing number of other surfaces? Before providing an answer, let us review some facts about the colouring of graphs embeddable on a certain surface. The minimum number of colours $\gamma(\varepsilon)$ necessary to guarantee that any graph embeddable on a surface $\mathcal{S}$ of Euler genus ε can be coloured with $\gamma(\varepsilon)$ colours is called the *chromatic number of* $\mathcal{S}$. Clearly $\gamma(\varepsilon) = \max\{\chi(G) \mid G \text{ is embeddable on } \mathcal{S}\}$.

For both orientable and non-orientable surfaces, Euler's formula generalises to the Euler-Poincaré formula $n - m + f = 2 - \varepsilon$. From this it is possible to deduce that for any graph G embeddable on a surface $\mathcal{S}$ with $\varepsilon \geq 1$, we have $\deg(G) \leq H(\varepsilon) - 1$, where $H(\varepsilon)$

is the *Heawood number*, given by

$$H(\varepsilon) = \left\lfloor \frac{7 + \sqrt{1 + 24\varepsilon}}{2} \right\rfloor .$$

A proof of this result may be found in [52]; specifically, see Theorem 8.3.1 on p. 230.

In 1890, Heawood [31] conjectured that $\gamma(\varepsilon) = H(\varepsilon)$. (He in fact conjectured this only for orientable surfaces, but the conjecture easily generalises and, as we shall see shortly, is *nearly* as true for a non-orientable surface as for an orientable one.) It is clear that $\gamma(\varepsilon) \leq H(\varepsilon)$. For $\varepsilon \geq 1$, this follows from the fact that $\deg(G) \leq H(\varepsilon) - 1$ for any graph G embeddable on $\mathcal{S}$. For $\varepsilon < 1$, we are in fact dealing with the sphere. In this case equality follows from the Four Colour Theorem [2, 3, 56], which asserts that $\gamma(0) = 4$.

It was not until 1968 that Ringel and Youngs (see [55] and references therein) managed to complete the proof that $\gamma(\varepsilon) \geq H(\varepsilon)$ holds for all surfaces with $\varepsilon \geq 1$ except the Klein bottle. They did this by showing how a complete graph on $H(\varepsilon)$ vertices embeds on any $\mathcal{S} \neq N_2$ of Euler genus ε. (In fact, it is also true that any graph with chromatic number $H(\varepsilon)$ embeddable on $\mathcal{S} \neq N_2$ with $\varepsilon \geq 1$ contains a complete graph on $H(\varepsilon)$ vertices as a subgraph—this was proved by Dirac [18, 19] for the torus and $\varepsilon \geq 4$, and by Albertson and Hutchinson [1] for $\varepsilon = 1$ and $\varepsilon = 3$.) Franklin [23] showed that for the Klein bottle we do not have a maximum chromatic number of $H(2) = 7$ but of 6. (He also showed that there are 6-chromatic graphs embeddable on the Klein bottle that do not contain a K_6.) Thus Heawood's conjecture is true for all surfaces except the Klein bottle; in particular, it is true for all orientable surfaces.

All these results imply the following.

Theorem 2.8
Let $\mathcal{S}$ be any surface, excluding the sphere and the Klein bottle, and let $\mathcal{S}$ have Euler genus ε. Then $\mu(\mathcal{S})$, the mixing number of $\mathcal{S}$, is given by $\mu(\mathcal{S}) = H(\varepsilon) + 1$, where $H(\varepsilon)$ is the Heawood number of $\mathcal{S}$.

Any graph embeddable on the Klein bottle is guaranteed to be 8-mixing since such a graph is 6-degenerate. Franklin [23] also proved that K_7 is not embeddable on this surface, but that K_6 is; we thus have a non-6-mixing graph embeddable on the Klein bottle. Whether all graphs embeddable on the Klein bottle are 7-mixing or not remains an open question. We point out a result that hints at the fact that determining the mixing number of the Klein bottle is unlikely to be as straightforward as for all other surfaces. This result states that there is no 6-regular graph embeddable on the Klein bottle which has frozen

7-colourings, and is a consequence of the following result of Hliněný [33]. Let us say that a graph G is a *cover* of a graph H if there exists a surjection $\varphi : V(G) \to V(H)$ such that for every vertex v of G, φ maps the neighbours of v in G bijectively to the neighbours of $\varphi(v)$ in H. Thus a cover of a complete graph K_k is precisely a $(k-1)$-regular graph that has frozen k-colourings. Hliněný [33] proves, amongst other results, the following.

Theorem 2.9 (Hliněný [33])
The complete graph K_7 has no cover which is embeddable on the Klein bottle.

This means that if there is a graph embeddable on the Klein bottle which has frozen 7-colourings, it cannot be 6-regular. On the other hand, if there are no graphs embeddable on N_2 with frozen 7-colourings, and $\mu(N_2) = 8$, proving that this is the correct number—that is, proving that there are non-7-mixing graphs embeddable on the Klein bottle—will in all likelihood require some ingenuity. Similarly, if it happens that $\mu(N_2) = 7$, proving this will require an argument beyond the simple recolouring procedure provided by following a degenerate ordering.

2.2 Mixing k-colourings in k-chromatic graphs

We have seen that 2-chromatic graphs are not 2-mixing. What about the k-mixing properties of k-chromatic graphs for values of $k \geq 3$? In this section we prove that 3-chromatic graphs are not 3-mixing, and that, for $k \geq 4$, a k-chromatic graph may or may not be k-mixing.

Graphs with chromatic number 3

Let G be a 3-colourable graph. To *orient* a cycle in G means to orient each edge on the cycle so that a directed cycle is obtained. If C is a cycle, then by $\vec{C}$ we denote the cycle with one of the two possible orientations. Given a 3-colouring α, the *weight of an edge $e = uv$ oriented from u to v* is

$$w(\overrightarrow{uv}, \alpha) = \begin{cases} +1, & \text{if } \alpha(u)\alpha(v) \in \{12, 23, 31\}; \\ -1, & \text{if } \alpha(u)\alpha(v) \in \{21, 32, 13\}. \end{cases} \tag{2.1}$$

The *weight $W(\vec{C}, \alpha)$ of an oriented cycle $\vec{C}$* is the sum of the weights of its oriented edges:

$$W(\vec{C},\alpha) = \sum_{\vec{uv}\in E(\vec{C})} w(\vec{uv},\alpha).$$

Lemma 2.10

Let α and β be 3-colourings of a graph G that contains a cycle C. If α and β are in the same component of $\mathcal{C}_3(G)$, then $W(\vec{C},\alpha) = W(\vec{C},\beta)$.

Proof. Let α and α' be 3-colourings of G that are adjacent in $\mathcal{C}_3(G)$, and suppose the two 3-colourings differ on vertex v. If v is not on C, then we certainly have $W(\vec{C},\alpha) = W(\vec{C},\alpha')$.

If v is a vertex of C, then all its neighbours must have the same colour in α, for otherwise we would not be able to recolour v. If we denote the in-neighbour of v on $\vec{C}$ by v_i and its out-neighbour by v_o, then this means that $w(\vec{v_iv},\alpha)$ and $w(\vec{vv_o},\alpha)$ have opposite sign, hence $w(\vec{v_iv},\alpha) + w(\vec{vv_o},\alpha) = 0$. Recolouring vertex v will change the signs of the weights of the oriented edges $\vec{v_iv}$ and $\vec{vv_o}$, but they will remain opposite. Therefore $w(\vec{v_iv},\alpha') + w(\vec{vv_o},\alpha') = 0$, and it follows that $W(\vec{C},\alpha) = W(\vec{C},\alpha')$.

From the above we immediately obtain that the weight of an oriented cycle is constant on all 3-colourings in the same component of $\mathcal{C}_3(G)$. □

We observe that the converse of Lemma 2.10 is not true. Given a 3-colouring of an oriented 3-cycle, consider a second 3-colouring obtained by changing the colour on each vertex to that of its unique out-neighbour in the original colouring. The two colourings are not connected—they are in fact both frozen—but the weight of the cycle is the same for each.

Lemma 2.11

Let α be a 3-colouring of a graph G that contains a cycle C. If $W(\vec{C},\alpha) \neq 0$, then G is not 3-mixing.

Proof. Let β be the 3-colouring of G obtained by setting, for each vertex v of G:

$$\beta(v) = \begin{cases} 1, & \text{if } \alpha(v) = 2; \\ 2, & \text{if } \alpha(v) = 1; \\ 3, & \text{if } \alpha(v) = 3. \end{cases}$$

It is easy to check that for each edge e of C, $w(\vec{e},\alpha) = -w(\vec{e},\beta)$, which gives $W(\vec{C},\alpha) = -W(\vec{C},\beta)$. Since $W(\vec{C},\alpha) \neq 0$, we must have $W(\vec{C},\alpha) \neq W(\vec{C},\beta)$, and so, by Lemma 2.10, α and β belong to different components of $\mathcal{C}_3(G)$. □

Theorem 2.12

Let G be a graph with chromatic number 3. Then G is not 3-mixing.

Proof. As G has chromatic number 3, it contains a cycle C of odd length. Let α be a 3-colouring of G, and note that as the weight of each edge in $\vec{C}$ is $+1$ or -1, $W(\vec{C}, \alpha) \neq 0$. We are done by Lemma 2.11. □

Given this result, one may now ask about the 3-mixing properties of bipartite graphs. We study this question in detail in the following chapter.

Graphs with chromatic number at least 4

For any $k \geq 4$, it is easy to find graphs with chromatic number k that are not k-mixing. For example, the complete graph K_k or any k-chromatic graph that contains it as an induced subgraph is not k-mixing. We now show that, in contrast to the results we have seen for graphs with chromatic number 2 or 3, for $k \geq 4$, there exist graphs with chromatic number k that are k-mixing. The following definition describes examples of such graphs.

Definition 2.13

Let $m \geq 4$. The graph H_m has

- vertex set $V(H_m) = \{u, v_1, v_2, \ldots, v_{m-1}, w_1, w_2, \ldots, w_{m-1}\}$, and
- edge set $E(H_m) = \{v_iv_j \,|\, 1 \leq i < j \leq m-1\} \cup \{w_iw_j \,|\, 1 \leq i < j \leq m-1\}$
$\cup \{uv_i \,|\, 2 \leq i \leq m-1\} \cup \{uw_i \,|\, 2 \leq i \leq m-1\} \cup \{v_1w_1\}$.

It is easy to verify that the graphs H_m are m-chromatic. This actually follows from the fact that H_m is obtained from two copies of K_m using Hajos' construction; see, for example, [17, pp. 117–118]. This also means that it is *m-critical,* which means that removing any vertex or edge from H_m will yield a graph with chromatic number less than m. We observe that the two set of vertices $\{v_2, v_3, \ldots, v_{m-1}\}$ and $\{w_2, w_3, \ldots, w_{m-1}\}$ induce two complete graphs isomorphic to K_{m-2}. This allows for a simple representation of H_m, as the sketch in Figure 2.3 shows. Note that the degeneracy of H_m is $m-1$ and so by Theorem 2.7, H_m is k-mixing for all $k \geq m+1$. In fact:

Theorem 2.14

For every fixed $m \geq 4$, the graph H_m is m-mixing.

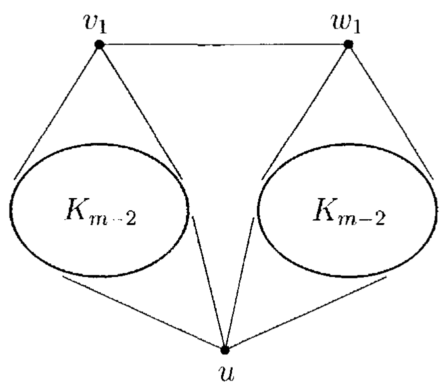

Figure 2.3 The graph H_m.

We shall prove Theorem 2.14 via the following sequence of claims, first giving some definitions. Let us divide the m-colourings of H_m into classes according to the colour of v_1 and w_1. An m-colouring α is a (c, c')*-colouring* if $\alpha(v_1) = c$ and $\alpha(w_1) = c'$. If also $\alpha(u) = c$, we call α a *standard* (c, c')-colouring.

We will show that H_m is m-mixing by showing that

- every m-colouring is connected to a standard colouring;
- for any pair c, c', the set of all standard (c, c')-colourings is connected; and
- for any two pairs c, c' and d, d', each standard (c, c')-colouring is connected to a standard (d, d')-colouring.

Claim 2.15
Let c and c' be distinct colours. Let α be a (c, c')-colouring of H_m where $\alpha(u) = c''$. Then there is a path from α to a standard (c, c')-colouring or to a standard (c'', c')-colouring of H_m in $C_m(H_m)$.

Proof. Let us assume $c \neq c''$, for else we are done. Note that as $\alpha(v_1) = c$, $\alpha(v_i) \neq c$ for $2 \leq i \leq m-1$. If it is not possible to immediately recolour u with c to obtain a standard (c, c')-colouring, then there must be a vertex w_j, $j \in \{2, \ldots, m-1\}$, such that $\alpha(w_j) = c$.

If $c'' = c'$, then, as two of the $m - 1$ neighbours of w_j are coloured c', there is some colour d not used on either w_j or any of its neighbours. Recolour w_j with d and then u with c to obtain a standard (c, c')-colouring.

If $c'' \neq c'$, then no neighbour of v_1 is coloured c''. By recolouring v_1 with c'', we immediately obtain a standard (c'', c')-colouring. □

Claim 2.16

For each distinct pair of colours c and c', all standard (c, c')-colourings belong to the same connected component of $\mathcal{C}_m(H_m)$.

Proof. Let α and β be distinct standard (c, c')-colourings and let x be the first vertex in the ordering $v_2, \ldots, v_{m-1}, w_2, \ldots, w_{m-1}$ at which α and β disagree. To prove the claim, we show that from α we can recolour to obtain a standard (c, c')-colouring that agrees with β on x and all vertices prior to it in the ordering.

Suppose that $x = v_i$ for some $i \in \{2, \ldots, m-1\}$. We simply recolour v_i with $\beta(v_i)$ unless there is a vertex v_j such that $\alpha(v_j) = \beta(v_i)$; in which case, by the choice of x, $j > i$. Note that a total of $m-1$ colours are used on $u, v_1, \ldots, v_{m-1}$ in any standard (c, c')-colouring, so there is a colour d available for v_j. Recolour v_j with d and then recolour v_i with $\beta(v_i)$.

The other possibility is that $x = w_i$ for some $i \in \{2, \ldots, m-1\}$. Much as before, recolour w_i with $\beta(w_i)$ unless there is a vertex w_j, $j > i$, such that $\alpha(w_j) = \beta(w_i)$. In this case, however, there is no colour available for w_j. Hence we find, if necessary, a vertex $v_l \in \{v_2, \ldots, v_{m-1}\}$ coloured c' and recolour it with its available colour. In any case, u can now be recoloured c' and so c is now available at w_j. Finally we perform the following sequence of recolourings: w_j with c, w_i with $\beta(w_i)$, w_j with $\alpha(w_i)$, u with c and, if such a vertex was found, v_l with $\alpha(v_l)$. □

Claim 2.17

Let α be a standard (c, c')-colouring of H_m. Then there is a path from α to a standard (c', c'')-colouring of H_m for any $c'' \notin \{c, c'\}$.

Proof. From α, we describe a sequence of recolourings that lead to a standard (c', c'')-colouring. First, if one of $v_2, \ldots, v_{m-1}$ is coloured c', it is recoloured with its available colour. Then u is recoloured c'. Next, if one of $w_2, \ldots, w_{m-1}$ is coloured c'', it is recoloured c. Then w_1 is recoloured c'' and v_1 is recoloured c'. □

Proof of Theorem 2.14. Let α and β be two m-colourings of H_m; we must show that they are connected. By Claim 2.15, we can assume that they are standard colourings. So suppose that α is a standard (c, c')-colouring and that β is a standard (d, d')-colouring.

By Claim 2.16, it is sufficient to find a path from α to any standard (d, d')-colouring. There are a number of cases to consider.

Suppose that $d = c'$. If $d' \neq c$, then the theorem follows immediately from Claim 2.17. If $d' = c$, then let b and b' be distinct colours not in $\{c, c'\}$. (As $m \geq 4$, such colours can be found. This need to have four colours available, explains, in essence, why the theorem is not correct for smaller m.) Now we repeatedly apply Claim 2.17: from α we can find a path to a standard (c', b)-colouring, then to a standard (b, b')-colouring, then a standard (b', c')-colouring and finally a standard (c', c)-colouring.

Suppose that $d = c$. Then if $d' = c'$ the result follows from Claim 2.16. Otherwise, applying Claim 2.17, we find a path from α to a standard (c', b)-colouring (for some colour b distinct from c, c' and d'), then to a standard (b, c)-colouring, and then to the required standard (c, d')-colouring.

If $d \notin \{c, c'\}$ and $d' \neq c'$, then Claim 2.17 gives a path from α to a standard (c', d)-colouring and then to a standard (d, d')-colouring. Otherwise, for $d' = c'$, we can recolour α to β via a standard (c', b)-colouring and a standard (b, d)-colouring, for some b distinct from c, c' and d, as above. This completes the proof. $\square$

Graphs that are mixing for specified values only

The results proved in this chapter allow us to characterise all positive integers l and sets F with $\min F \geq l$ such that there exist graphs G with $\chi(G) = l$ that are k-mixing if and only if $k \notin F$.

Theorem 2.18
Let $l \geq 2$ be an integer, and F a set of integers with $\min F \geq l$, if $F \neq \varnothing$. Then the following two statements are equivalent.

(i) There exists a graph G with chromatic number l such that for all $k \geq l$, G is k-mixing if and only if $k \notin F$.

(ii) The set F is finite, and if $l \in \{2, 3\}$, then $l \in F$.

Proof. By Theorem 2.7, a graph can be non-k-mixing for a finite number of k only. By Proposition 2.1 and Theorem 2.12, a graph with chromatic number $l \in \{2, 3\}$ cannot be l-mixing. Hence statement (i) implies statement (ii).

Before proving the converse, let us make some basic observations and recollections. If a graph G is the disjoint union of graphs $G_1, \ldots, G_s$, then we obviously have that $\chi(G)$ is

$\max\{\, \chi(G_i) \mid i = 1, \ldots, s\,\}$, and G is k-mixing if and only each G_i, $1 \le i \le s$, is k-mixing. We have just seen that for $m \ge 4$, H_m has chromatic number m and is k-mixing if and only if $k \ge m$. Similarly, the complete graph K_m is m-chromatic and is k-mixing only for $k \ge m+1$, since $\deg(K_m) = m-1$. Let us also recall the graphs L_m from Definition 2.2: for every $m \ge 3$, L_m has chromatic number 2 and is k-mixing if and only if $k \ge 3$ and $k \ne m$.

Now let l and F be as in the theorem and suppose that statement (ii) holds. If $F = \varnothing$, we are in the case $l \ge 4$ and the graph H_l will do the trick for (i). Hence we can assume that F is not empty and finite. Let us write $F = \{p_1, \ldots, p_t\}$, with $p_1 = \min F$. Then if $l \in F$ (so $p_1 = l$) the disjoint union of $K_l, L_{p_2}, \ldots, L_{p_t}$ has chromatic number l, and for $k \ge l$, the graph is k-mixing if and only if $k \notin F$. Otherwise, if $l \notin F$, we must have $p_1 > l \ge 4$, and then the disjoint union of $H_l, L_{p_1}, \ldots, L_{p_t}$ yields a graph for which (i) holds. $\square$

3

Mixing 3-colourings

In this chapter we investigate what can be said about the 3-mixing properties of a given graph. Having already considered some facts about 3-colourings (of 3-chromatic graphs) in Chapter 2, we will refer to definitions and results from the relevant section, Section 2.2. Recall that we saw in Theorem 2.12 that if G is a 3-chromatic graph, then G is not 3-mixing. For this reason we focus exclusively on bipartite graphs in this chapter.

In Section 3.1 we give two equivalent characterisations of a 3-mixing bipartite graph; one in terms of the possible 3-colourings it may have, the other in terms of its structure. In Section 3.2 we consider the problem of deciding whether a given bipartite graph is 3-mixing, and prove that this problem is coNP-complete. In the final section, Section 3.3, we describe an algorithm that answers this question for bipartite planar graphs in polynomial time.

3.1 Characterising 3-mixing graphs

Let us make some preliminary observations about the 3-mixing properties of some specific graphs, noting in particular that there exist 3-mixing bipartite graphs as well as non-3-mixing bipartite graphs. By Theorem 2.7 we know that any 1-degenerate graph is 3-mixing. Hence all trees are 3-mixing. The cycle on four vertices, C_4, is also 3-mixing—this is easily verified by hand after noting that any 3-colouring of a 4-cycle has a pair of vertices at distance two which are coloured the same. All other even cycles, however, are not 3-mixing. Given a cycle C_{2m} with $2m \geq 6$, it is easy to construct a 3-colouring α of C_{2m} so that $W(\overrightarrow{C}_{2m}, \alpha) \neq 0$: just use the colour pattern $1, 2, 3, 1, 2, 3, \ldots$ for as long as possible, making sure that the final vertices are properly coloured. Lemma 2.11 then guarantees that the cycle C_{2m} is not 3-mixing.

In Theorem 3.1 below we distinguish between 3-mixing and non-3-mixing bipartite graphs in terms of their structure and the possible 3-colourings they may have. Before being able to state it we need two simple definitions.

If v and w are vertices of a bipartite graph G at distance two, then we define a *pinch* on v and w as the identification of v and w (together with the replacement of all double edges by single edges). We say that G is *pinchable* to a graph H if there exists a sequence of pinches that transforms G into H.

Theorem 3.1
Let G be a connected bipartite graph. Then the following statements are equivalent.

(i) The graph G is not 3-mixing.

(ii) There exists a cycle C in G and a 3-colouring α of G with $W(\overrightarrow{C}, \alpha) \neq 0$.

(iii) The graph G is pinchable to the 6-cycle C_6.

To prove Theorem 3.1, we need some definitions and technical lemmas. For the rest of this section, let $G = (V, E)$ denote a connected bipartite graph with vertex bipartition X, Y.

Given a 3-colouring α of G, let us define a *height function for α with base X* as a function $h : V \to \mathbb{Z}$ satisfying the following three conditions. (See [5, 25] for other, similar definitions and uses of height functions.)

(H1) For all $v \in X$, $h(v) \equiv 0 \pmod 2$; and for all $v \in Y$, $h(v) \equiv 1 \pmod 2$.

(H2) For all $uv \in E$, $|h(v) - h(u)| = 1$.

(H3) For all $v \in V$, $h(v) \equiv \alpha(v) \pmod 3$.

If $h : V \to \mathbb{Z}$ satisfies conditions (H2), (H3) and also

(H1′) For all $v \in X$, $h(v) \equiv 1 \pmod 2$; and for $v \in Y$, $h(v) \equiv 0 \pmod 2$,

then h is said to be a height function for α with base Y.

Observe that for a particular colouring of a given G, a height function might not exist. An example of this is the 6-cycle C_6 coloured 1-2-3-1-2-3.

Conversely, however, a function $h : V \to \mathbb{Z}$ satisfying conditions (H1) and (H2) induces a 3-colouring of G: the unique $\alpha : V \to \{1, 2, 3\}$ satisfying condition (H3), and so h is in fact a height function for this α. Observe also that if h is a height function for α with base X, then so are $h + 6$ and $h - 6$; while $h + 3$ and $h - 3$ are height functions for α

with base Y. Because we will be concerned solely with the question of *existence* of height functions, we assume henceforth that for a given G, all height functions have base X. Thus we let $\mathcal{H}_X(G)$ be the set of height functions with base X corresponding to some 3-colouring of G, and, following [25], we define a metric m on $\mathcal{H}_X(G)$ by setting

$$m(h_1, h_2) = \sum_{v \in V} |h_1(v) - h_2(v)|,$$

for $h_1, h_2 \in \mathcal{H}_X(G)$. Note that condition (H1) above implies that $m(h_1, h_2)$ is always even.

For a given height function h, $h(v)$ is said to be a *local maximum* (respectively, *local minimum*) if $h(v)$ is larger than (respectively, smaller than) $h(u)$ for all neighbours u of v. Again following [25], we define the following *height transformations* on h.

An *increasing height transformation* takes a local minimum $h(v)$ of h and transforms h into the height function h' given by

$$h'(x) = \begin{cases} h(x) + 2, & \text{if } x = v; \\ h(x), & \text{if } x \neq v. \end{cases}$$

A *decreasing height transformation* takes a local maximum $h(v)$ of h and transforms h into the height function h' given by

$$h'(x) = \begin{cases} h(x) - 2, & \text{if } x = v; \\ h(x), & \text{if } x \neq v. \end{cases}$$

Note that these height transformations give rise to transformations between the corresponding colourings. Specifically, if we let α' be the 3-colouring corresponding to h', an increasing transformation yields $\alpha'(v) = \alpha(v) - 1$ and $\alpha'(x) = \alpha(x)$ for all $x \neq v$, while a decreasing transformation yields $\alpha'(v) = \alpha(v) + 1$ and $\alpha'(x) = \alpha(x)$ for all $x \neq v$, where addition is modulo 3.

The following lemma, a simple extension of the range of applicability of a similar lemma appearing in [25], shows that all colourings with height functions are connected in $C_3(G)$.

Lemma 3.2 (Goldberg, Martin and Paterson [25])
Let α, β be two 3-colourings of G with corresponding height functions h_α, h_β. Then there is a path between α and β in $C_3(G)$.

Proof. We use induction on $m(h_\alpha, h_\beta)$. The lemma is trivially true when $m(h_\alpha, h_\beta) = 0$, since in this case α and β are identical.

Suppose therefore that $m(h_\alpha, h_\beta) > 0$. We show that there is a height transformation transforming h_α into some height function h with $m(h, h_\beta) = m(h_\alpha, h_\beta) - 2$, from which the lemma follows.

Without loss of generality, let us assume that there is some vertex $v \in V$ with $h_\alpha(v) > h_\beta(v)$, and let us choose v with $h_\alpha(v)$ as large as possible. We show that such a v must be a local maximum of h_α. Let u be any neighbour of v. If $h_\alpha(u) > h_\beta(u)$, then it follows that $h_\alpha(v) > h_\alpha(u)$, since v was chosen with $h_\alpha(v)$ maximum, and $|h_\alpha(v) - h_\alpha(u)| = 1$. If, on the other hand, $h_\alpha(u) \leq h_\beta(u)$, we have $h_\alpha(v) \geq h_\beta(v) + 1 \geq h_\beta(u) \geq h_\alpha(u)$, which in fact means $h_\alpha(v) > h_\alpha(u)$.

Thus $h_\alpha(v) > h_\alpha(u)$ for all neighbours u of v, and we can apply a decreasing height transformation to h_α at v to obtain h. Clearly $m(h, h_\beta) = m(h_\alpha, h_\beta) - 2$. □

In the same way that we consider weights of oriented cycles in a 3-coloured graph, let us consider weights of oriented paths. For a path P in a graph G, let $\overrightarrow{P}$ denote one of the two possible oriented paths obtainable from P. If G is 3-coloured with α, we define the *weight* $W(\overrightarrow{P}, \alpha)$ *of* $\overrightarrow{P}$ as the sum of the weights of its oriented edges:

$$W(\overrightarrow{P}, \alpha) = \sum_{\overrightarrow{uv} \in E(\overrightarrow{P})} w(\overrightarrow{uv}, \alpha),$$

where $w(\overrightarrow{uv}, \alpha)$ takes values as defined in equation (2.1).

The next lemma tells us that for a given 3-colouring, non-zero weight cycles are, in some sense, the obstructing configurations forbidding the existence of a corresponding height function.

Lemma 3.3
Let α be a 3-colouring of G with no corresponding height function. Then G contains a cycle C for which $W(\overrightarrow{C}, \alpha) \neq 0$.

Proof. Let us observe that if a 3-colouring of a certain graph does have a height function, it is possible to construct one by fixing a vertex x of the graph, giving x an appropriate height (satisfying conditions (H1) and (H3)) and then assigning heights to all vertices in the graph by following a breadth-first ordering from x.

Whenever we attempt to construct a height function h for α in such a fashion, we must come to a stage in the ordering where we attempt to give some vertex v a height $h(v)$ and find ourselves unable to because v has a neighbour u with a previously assigned height $h(u)$ and $|h(u) - h(v)| > 1$. Letting P be a path between u and v formed by vertices that have been assigned a height, and choosing the appropriate orientation of P, we have $W(\overrightarrow{P}, \alpha) = |h(u) - h(v)|$. The lemma now follows by letting C be the cycle formed by P and the edge uv. □

The following lemma is obvious.

Lemma 3.4

Let u and v be vertices on a cycle C in a graph G, and suppose there is a path P between u and v in G internally disjoint from C. Let α be a 3-colouring of G. Let C' and C'' be the two cycles formed from P and edges of C, and let $\overrightarrow{C'}, \overrightarrow{C''}$ be the orientations of C', C'' induced by an orientation $\overrightarrow{C}$ of C (so the edges of P have opposite orientations in $\overrightarrow{C'}$ and $\overrightarrow{C''}$). Then $W(\overrightarrow{C}, \alpha) = W(\overrightarrow{C'}, \alpha) + W(\overrightarrow{C''}, \alpha)$.

Note this tells us that $W(\overrightarrow{C}, \alpha) \neq 0$ implies $W(\overrightarrow{C'}, \alpha) \neq 0$ or $W(\overrightarrow{C''}, \alpha) \neq 0$.

Proof of Theorem 3.1. Let G be a connected bipartite graph.

(i) $\Longrightarrow$ (ii). Suppose $\mathcal{C}_3(G)$ is not connected. Take two 3-colourings of G, α and β, in different components of $\mathcal{C}_3(G)$. By Lemma 3.2 we know at least one of them, say α, has no corresponding height function, and by Lemma 3.3, there is a cycle C in G with $W(\overrightarrow{C}, \alpha) \neq 0$.

(ii) $\Longrightarrow$ (iii). Let G contain a cycle C with $W(\overrightarrow{C}, \alpha) \neq 0$ for some 3-colouring α of G. Because $W(\overrightarrow{C_4}, \beta) = 0$ for any 3-colouring β of C_4, it follows that $C = C_n$ for some even $n \geq 6$. If $G = C$, then it is easy to find a sequence of pinches that will yield C_6. If G is C plus some chords, then Lemma 3.4 tells us that there is a smaller cycle C' with $W(\overrightarrow{C'}, \alpha) \neq 0$ and we can again easily find a sequence of pinches that will yield C_6. Thus if $G \neq C$, we can assume that $V(G) \neq V(C)$, and we describe how to pinch a pair of vertices so that (ii) remains satisfied (for a specified cycle with G replaced by the graph created by the pinch and α replaced by its restriction to that graph, also denoted α); by repetition, we can obtain a graph that is a cycle and, by the previous observations, the implication is proved.

Note that we shall choose vertices coloured alike to pinch so that the restriction of α to the graph obtained is well-defined and proper. If C has three consecutive vertices u, v, w with $\alpha(u) = \alpha(w)$, pinching u and w yields a graph containing a cycle $C' = C_{n-2}$ with $W(\overrightarrow{C'}, \alpha) = W(\overrightarrow{C}, \alpha)$. Otherwise C is coloured 1-2-3-$\cdots$-1-2-3. We can choose u, v, w to be three consecutive vertices of C, such that there is a vertex $x \notin V(C)$ adjacent to v. Suppose, without loss of generality, that $\alpha(x) = \alpha(u)$, and pinch x and u to obtain a graph in which $W(\overrightarrow{C}, \alpha)$ is unchanged.

(iii) $\Longrightarrow$ (i). Suppose G is pinchable to C_6. Take two 3-colourings of C_6 not connected by a path in $\mathcal{C}_3(C_6)$: 1-2-3-1-2-3 and 1-2-1-2-1-2, for example. Considering the appropriate orientation of C_6, note that the first colouring has weight 6 and the second has weight 0. We construct two 3-colourings of G not connected by a path in $\mathcal{C}_3(G)$ as follows. Consider the reverse sequence of pinches that gives G from C_6. Following this sequence, for each colouring of C_6, give every pair of new vertices introduced by an 'unpinching' the same colour as the vertex from which they originated. In this manner we obtain two 3-colourings of G, α and β, say. Observe that every unpinching maintains a cycle in G which has weight 6 with respect to the colouring induced by the first colouring of C_6 and weight 0 with respect to the second induced colouring. This means G will contain a cycle C for which $W(\overrightarrow{C},\alpha) = 6$ and $W(\overrightarrow{C},\beta) = 0$, showing that α and β cannot possibly be in the same connected component of $\mathcal{C}_3(G)$.

This completes the proof of the theorem. □

3.2 The complexity of 3-Mixing

Let us now turn our attention to the computational complexity of deciding whether or not a 3-colourable graph G is 3-mixing. From Theorem 2.12 we know that we can restrict our attention to bipartite graphs, so the case $k = 3$ of our decision problem k-Mixing we formally define as follows.

3-Mixing
Instance : A connected bipartite graph G.
Question : Is G 3-mixing?

Observing that Theorem 3.1 gives us two polynomial time verifiable certificates for when G is *not* 3-mixing, we immediately obtain that 3-Mixing is in the complexity class coNP. By the same theorem, the following decision problem is the complement of 3-Mixing.

Pinchability-to-C_6
Instance : A connected bipartite graph G.
Question : Is G pinchable to C_6?

We will prove the following result.

Theorem 3.5
The decision problem 3-MIXING *is* coNP-*complete.*

Our proof will in fact show that PINCHABILITY-TO-C_6 is NP-complete. We will obtain a reduction from the following decision problem.

RETRACTABILITY-TO-C_6
Instance : A connected bipartite graph G with an induced 6-cycle S.
Question : Is G *retractable* to S? That is, does there exist a homomorphism $r : V(G) \to V(S)$ such that $r(v) = v$ for all $v \in V(S)$?

It is mentioned in [60], without references, that Tomás Feder and Gary MacGillivray have independently proved that RETRACTABILITY-TO-C_6 is NP-complete by a reduction from 3-COLOURABILITY. For completeness, we give a sketch proof.

3-COLOURABILITY
Instance : A connected graph G.
Question : Is G 3-colourable?

Theorem 3.6 (Feder; MacGillivray; see [60])
The decision problem RETRACTABILITY-TO-C_6 *is* NP-*complete.*

Proof. That RETRACTABILITY-TO-C_6 is in NP is clear.

Given a connected graph G, construct a new graph G' as follows: subdivide every edge uv of G by inserting a vertex y_{uv} between u and v. Also add new vertices a, b, c, d, e together with edges za, ab, bc, cd, de, ez, where z is a particular vertex of G (any one will do). The graph G' is clearly connected and bipartite, and the vertices z, a, b, c, d, e induce a 6-cycle S. We will prove that G is 3-colourable if and only if G' retracts to the induced 6-cycle S.

Assume that G is 3-colourable and take a 3-colouring τ of G with $\tau(z) = 1$. From τ we construct a 6-colouring σ of G'. For this, first set $\sigma(x) = \tau(x)$, if $x \in V(G)$. For the new vertices y_{uv} set

$$\sigma(y_{uv}) = \begin{cases} 4, & \text{if } \tau(u) = 1 \text{ and } \tau(v) = 2, \\ 5, & \text{if } \tau(u) = 2 \text{ and } \tau(v) = 3, \\ 6, & \text{if } \tau(u) = 3 \text{ and } \tau(v) = 1, \end{cases}$$

and for the cycle S take $\sigma(a) = 4$, $\sigma(b) = 2$, $\sigma(c) = 5$, $\sigma(d) = 3$ and $\sigma(e) = 6$. Now define $r : V(G') \to V(S)$ by setting

$$r(x) = \begin{cases} z, & \text{if } \sigma(x) = 1, \\ a, & \text{if } \sigma(x) = 4, \\ b, & \text{if } \sigma(x) = 2, \\ c, & \text{if } \sigma(x) = 5, \\ d, & \text{if } \sigma(x) = 3, \\ e, & \text{if } \sigma(x) = 6. \end{cases}$$

It is easy to check that r is a retraction of G' to S.

Conversely, suppose G' retracts to S. We can use this retraction to define a 6-colouring of G' in a similar way to that in which we defined r from σ in the preceding paragraph. The restriction of this 6-colouring to G yields a proper 3-colouring of G, completing the proof. □

Our proof of Theorem 3.5 follows [60], where, as a special case of the main result of that paper, the following problem is proved to be NP-complete.

COMPACTABILITY-TO-C_6
Instance: A connected bipartite graph G.
Question: Is G *compactable* to C_6? That is, does there exist an edge-surjective homomorphism $c : V(G) \to V(C_6)$?

If an edge-surjective homomorphism $c : V(G) \to V(C_6)$ exists, we call it a *compaction*.

In [60] a polynomial time reduction from RETRACTABILITY-TO-C_k to COMPACTABILITY-TO-C_k, with $k \geq 6$ even, is given. We will use exactly the same transformation (for $k = 6$) to prove that PINCHABILITY-TO-C_6 is NP-complete.

Proof of Theorem 3.5. As mentioned before, we will show that 3-MIXING is coNP-complete by showing that PINCHABILITY-TO-C_6 is NP-complete. This we do by giving a polynomial time reduction from RETRACTABILITY-TO-C_6 to PINCHABILITY-TO-C_6.

Consider an instance of RETRACTABILITY-TO-C_6: a connected bipartite graph G with an induced 6-cycle S. From G we construct, in time polynomial in the size of G, an instance G' of PINCHABILITY-TO-C_6 such that

$$G \text{ retracts to } S \text{ if and only if } G' \text{ is pinchable to } C_6. \qquad (*)$$

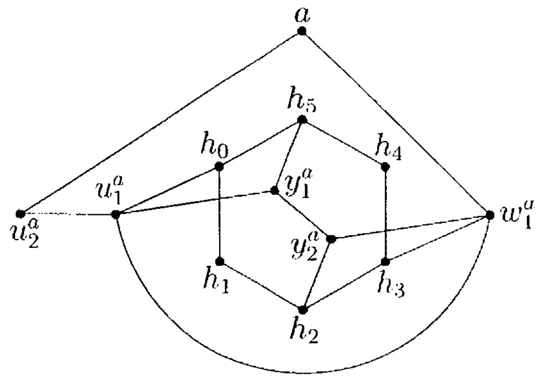

Figure 3.1 The subgraph of G' added around a vertex $a \in G_A \setminus S_A$, together with the 6-cycle S.

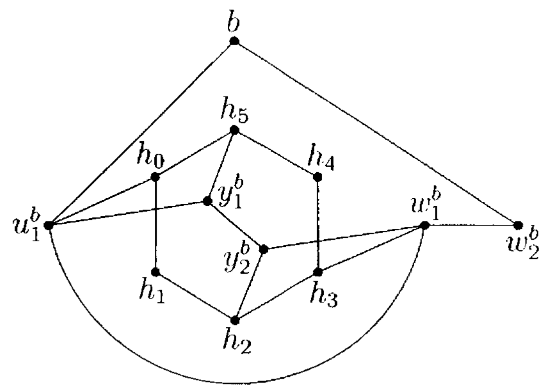

Figure 3.2 The subgraph of G' added around a vertex $b \in G_B \setminus S_B$, together with the 6-cycle S.

Assume G has vertex bipartition (G_A, G_B). Let $V(S) = S_A \cup S_B$, where $S_A = \{h_0, h_2, h_4\}$ and $S_B = \{h_1, h_3, h_5\}$, and assume $E(S) = \{\, h_0h_1, \ldots, h_4h_5, h_5h_0 \,\}$.

The construction of G' is as follows.

- For every vertex $a \in G_A \setminus S_A$, add to G new vertices $u_1^a, u_2^a, w_1^a, y_1^a, y_2^a$, together with edges $u_1^a h_0, au_2^a, w_1^a h_3, aw_1^a, u_1^a w_1^a, y_1^a h_5, y_2^a h_2, u_1^a y_1^a, w_1^a y_2^a, u_1^a u_2^a, y_1^a y_2^a$.
- For every vertex $b \in G_B \setminus S_B$, add to G new vertices $u_1^b, w_1^b, w_2^b, y_1^b, y_2^b$, together with edges $u_1^b h_0, bu_1^b, w_1^b h_3, bw_2^b, u_1^b w_1^b, y_1^b h_5, y_2^b h_2, u_1^b y_1^b, w_1^b y_2^b, w_1^b w_2^b, y_1^b y_2^b$.
- For every edge $ab \in E(G) \setminus E(S)$, with $a \in G_A \setminus S_A$ and $b \in G_B \setminus S_B$, add two new vertices: x_a^{ab} adjacent to a and u_1^a; and x_b^{ab} adjacent to b, w_1^b and x_a^{ab}.

From the construction it is clear that G' is connected and bipartite. Note that G' contains G as an induced subgraph, and note also that the subgraphs constructed around a vertex $a \in G_A \setminus S_A$ and a vertex $b \in G_B \setminus S_B$ are isomorphic. These graphs are depicted in Figures 3.1 and 3.2.

We will prove $(*)$ via a sequence of claims.

Claim 3.7

Suppose G retracts to S. Then G is pinchable to C_6.

Proof. The fact that G retracts to S means we have a homomorphism $r : V(G) \to V(S)$ such that $r(v) = v$ for all $v \in V(S)$. Define a partition $\{R_i \mid i = 0, 1, \ldots, 5\}$ of $V(G)$ by setting $v \in R_i \iff r(v) = h_i$. Because r is a homomorphism, we know any edge $e \in E(G)$ has one vertex in R_j and another in R_{j+1}, for some j, where subscript addition is modulo 6. Using this partition of $V(G)$, we show that G is pinchable to a 6-cycle—to S, in fact. We describe how to pinch a pair of vertices such that the resulting (smaller) graph still has S as an induced subgraph; by repetition, this will eventually yield S. Supposing $V(G) \neq V(S)$ (for else we are done), let $E^- = E(G) \setminus E(S)$. Because G is connected, there must be an edge $uv \in E^-$ with $u \in V(S)$ and $v \in V(G) \setminus V(S)$. Suppose $v \in R_j$, for some $j \in \{0, 1, \ldots, 5\}$. Pinch v with h_j, and note that the resulting graph remains bipartite, connected and contains S as an induced subgraph. Denote the resulting graph by G and repeat. □

We now prove the 'only if' part of $(*)$.

Claim 3.8

Suppose G retracts to S. Then G' is pinchable to C_6.

Proof. By Claim 3.7, G is pinchable to C_6. In fact, by the proof of Claim 3.7, we know G is pinchable to S. Because G is an induced subgraph of G', we can follow, in G', the sequence of pinches that gives S from G. We now show how, after following this sequence of pinches, we can choose some further pinches that will leave us with S. For a vertex $v \in V(G) \setminus V(S)$, we will pinch into S all vertices introduced to G' on account of v, yielding a smaller graph still containing S as an induced subgraph. By repetition, we will eventually end up with just S.

First let us consider where a vertex $a \in G_A \setminus S_A$ with no neighbours in $G_B \setminus S_B$ might have been pinched to, and how we could continue pinching. There are three possibilities.

1. The vertex a has been pinched with h_1. In that case pinch y_1^a with h_0, y_2^a with h_1, u_1^a with h_1, u_2^a with h_0, and w_1^a with h_2.

2. The vertex a has been pinched with h_3. In that case pinch y_1^a with h_4, y_2^a with h_3, u_1^a with h_5, u_2^a with h_4, and w_1^a with h_4.

3. The vertex a has been pinched with h_5. In that case pinch y_1^a with h_4, y_2^a with h_3, u_1^a with h_5, u_2^a with h_0, and w_1^a with h_4.

Similarly, let us consider where a vertex $b \in G_B \setminus S_B$ with no neighbours in $G_A \setminus S_A$ might have been pinched to, and how we could continue pinching. Again, there are three possibilities.

1. The vertex b has been pinched with h_0. In that case pinch y_1^b with h_0, y_2^b with h_1, u_1^b with h_1, w_1^b with h_2, and w_2^b with h_1.

2. The vertex b has been pinched with h_2. In that case pinch y_1^b with h_0, y_2^b with h_1, u_1^b with h_1, w_1^b with h_2, and w_2^b with h_3.

3. The vertex b has been pinched with h_4. In that case pinch y_1^b with h_4, y_2^b with h_3, u_1^b with h_5, w_1^b with h_4, and w_2^b with h_3.

Now let us consider the case where a vertex $a \in G_A \setminus S_A$ is adjacent to a vertex $b \in G_B \setminus S_B$. There are six cases to consider, corresponding to the six edges of S to which ab might have been pinched. Often there will be a choice of pinches—for each case we give just one.

1. The edge ab has been pinched to h_1h_2. We can use the previous case analyses to conclude that u_1^a must be pinched with h_1 and w_1^b with h_2. Now we must deal with x_a^{ab} and x_b^{ab}. Pinching x_a^{ab} with h_2 and x_b^{ab} with h_1 gives us what we require.

2. The edge ab has been pinched to h_1h_0. Then we conclude u_1^a must be pinched with h_1 and w_1^b with h_2. Now pinch x_a^{ab} with h_0 and x_b^{ab} with h_1.

3. The edge ab has been pinched to h_3h_4. Then u_1^a must be pinched with h_5 and w_1^b with h_4. Now pinch x_a^{ab} with h_4 and x_b^{ab} with h_3.

4. The edge ab has been pinched to h_3h_2. Then u_1^a must be pinched with h_5 and w_1^b with h_2. Now pinch x_a^{ab} with h_4 and x_b^{ab} with h_3.

5. The edge ab has been pinched to h_5h_0. Then u_1^a must be pinched with h_5 and w_1^b with h_2. Now pinch x_a^{ab} with h_0 and x_b^{ab} with h_1.

6. The edge ab has been pinched to h_5h_4. Then u_1^a must be pinched with h_5 and w_1^b with h_4. Now pinch x_a^{ab} with h_4 and x_b^{ab} with h_5.

This completes the proof of the claim. □

We must now prove the 'if' part of $(*)$. We do this via the next three claims.

Claim 3.9

Suppose G' is pinchable to C_6. Then G' is compactable to C_6.

Proof. The fact that G' is pinchable to the 6-cycle C_6 means there exists a homomorphism $c : V(G') \to V(C_6)$. In order to make this precise, let $V(C_6) = \{k_0, k_1, k_2, k_3, k_4, k_5\}$ and $E(C_6) = \{k_0k_1, \dots, k_4k_5, k_5k_0\}$. Let us also define sets P_i, for $i = 0, 1, \dots, 5$, as follows. Initially, set $P_i = \{k_i\}$. Now let us consider the reverse sequence of 'unpinchings' that yields G' from C_6. Following this sequence, suppose a vertex $v \in P_j$ is unpinched. Delete v from P_j and add to P_j the two vertices that were identified to give v in the original pinch. Repeat this until G' is obtained, and now define c by setting, for $v \in V(G')$, $c(v) = k_i \iff v \in P_i$. Clearly the sets P_i form a partition of $V(G')$ and so c is well-defined. In addition, by the way the sets P_i have been constructed, it is clear that any edge $uv \in E(G')$ has one end-vertex in P_j and the other in P_{j+1}, for some $j \in \{0, 1, \dots, 5\}$. This means $c(u)c(v) \in E(C_6)$ and so c is a homomorphism. Moreover, it is edge-surjective: the P_is are all non-empty and there is at least one edge between every pair P_i, P_{i+1}. □

The proof of the following claim is the same as the proof in [60] that shows that if G' is compactable to C_6, then G' retracts to S.

We need some further notation. As usual, for a set S and a function f, we let $f(S) = \{f(s) \mid s \in S\}$. Recalling that we denote the distance between vertices u and v in a graph H by $d_H(u, v)$, let us write, for a vertex u and set of vertices S of H, $d_H(S, u) = \min\{d_H(v, u) \mid v \in S\}$.

Claim 3.10 (Vikas [60])

Suppose G' is pinchable to C_6. Then G' retracts to S.

Proof. By Claim 3.9 we know there exists a compaction $c : V(G') \to V(C_6)$. (Recall that a compaction is just an edge-surjective homomorphism.) We prove that c is in fact a retraction to S. To do this, we must show that for all $v \in V(S)$, $c(v) = v$.

For convenience, we now use the same notation for C_6 and S; that is, we let $V(C_6) = \{h_0, h_1, \ldots, h_5\}$ and $E(C_6) = \{\, h_0h_1, \ldots, h_4h_5, h_5h_0 \,\}$.

Let $U = \{u_1^v \mid v \in V(G) \setminus V(S)\} \cup \{h_0, h_1, h_5\}$ and $W = \{w_1^v \mid v \in V(G) \setminus V(S)\} \cup \{h_2, h_3, h_4\}$. Because both these vertex sets induce subgraphs of diameter 2 in G', $c(U)$ and $c(W)$ must each induce a path of length 1 or 2 in C_6. We prove they each induce a path of length 2.

Suppose that $c(U)$ has only two vertices, adjacent in C_6. Thus we let $c(U) = \{h_0, h_1\}$, with $c(h_0) = h_0$. (Due to the symmetry of C_6, we can, if necessary, redefine c in this way.) Let $U^- = U \setminus \{h_0\}$. Because h_0 is adjacent to every other vertex in U, $c(U^-) = \{h_1\}$. It is easy to check that for any $g \in G'$, $d_{G'}(U^-, g) \leq 2$. But we have $d_{C_6}(c(U^-), h_4) = d_{C_6}(h_1, h_4) = 3$, which means no $g \in G'$ can be mapped to h_4 under c, contradicting the fact that c is a compaction.

Hence $c(U)$ induces a path on three vertices. By a similar argument, the same applies to $c(W)$. By the symmetry of C_6, we can without loss of generality take $c(U) = \{h_1, h_0, h_5\}$. This means that $c(h_0) = h_0$. We now prove that $c(h_3) = h_3$.

Let gg' be an edge of G' that is mapped to h_3h_2 or h_3h_4, with $c(g) = h_3$, and $c(g') = h_2$ or $c(g') = h_4$. Note that h_3 is at distance 2 from $c(U)$ in C_6 while h_2 and h_4 are at distance 1 from $c(U)$ in C_6. This means that $d_{G'}(U, g) \geq 2$ and $d_{G'}(U, g') \geq 1$. Earlier we noted that the distance between U^- and any vertex of G' is at most 2, which means that $d_{G'}(U, g) \leq 2$, so in fact $d_{G'}(U, g) = 2$. Because G' is bipartite, $d_{G'}(U, g') = 1$. Hence g is one of $a, x_b^{ab}, h_3, y_2^a, y_2^b, w_2^b$, and g' is one of $b, x_a^{ab}, u_2^a, h_2, h_4, y_1^a, y_1^b, w_1^a, w_1^b$, for some $a \in G_A \setminus S_A$, $b \in G_B \setminus S_B$. Given that $c(h_0) = h_0$, we cannot have $c(h_3) = h_2$ or $c(h_3) = h_4$. Aiming for a contradiction, let us suppose that $c(h_3) \neq h_3$. Then no edge of G' with h_3 as an endpoint covers h_3h_2 or h_3h_4. Hence gg' must be one of the following: $ax_a^{ab}, ab, au_2^a, aw_1^a$, $x_b^{ab}x_a^{ab}, x_b^{ab}b, x_b^{ab}w_1^b$, $y_2^ay_1^a, y_2^aw_1^a, y_2^ah_2, y_2^by_1^b, y_2^bw_1^b, y_2^bh_2$, $w_2^bw_1^b, w_2^bb$. If ah_2 or ah_4 is an edge of G', then we also need to consider such an edge as a possible candidate for gg'. By previous assumptions, we have $c(h_3) = h_1$ or $c(h_3) = h_5$. We now prove that $c(h_3) \neq h_3$ is impossible as follows. We first assume $c(h_3) = h_1$ and show that no possible edge for gg' covers h_3h_4, and then assume $c(h_3) = h_5$ and show that no possible edge for gg' covers h_3h_2. Thus let us assume $c(h_3) = h_1$.

Let us suppose that for some $v \in V(G) \setminus V(S)$, $y_2^vw_1^v$ covers h_3h_4, so $c(y_2^v) = h_3$ and $c(w_1^v) = h_4$. But $c(h_3) = h_1$, and since h_3 an w_1^v are adjacent, we must have $c(w_1^v) = h_0$ or $c(w_1^v) = h_2$, a contradiction.

By exactly the same argument, we come to the conclusion that none of the edges aw_1^a, $w_2^bw_1^b$, $x_b^{ab}w_1^b$ can cover the edge h_3h_4. A similar argument applies to $y_2^vh_2$.

Suppose that for some $v \in V(G) \setminus V(S)$, $y_2^v y_1^v$ covers h_3h_4, so $c(y_2^v) = h_3$ and $c(y_1^v) = h_4$. Now $c(u_1^v) = h_1$ or $c(u_1^v) = h_5$, but since u_1^v and y_1^v are adjacent we must have $c(u_1^v) = h_5$. Because $c(w_1^v)$ must be adjacent to $c(y_2^v) = h_3$, $c(w_1^v) = h_2$ or $c(w_1^v) = h_4$. But u_1^v is adjacent to w_1^v, so $c(w_1^v) = h_4$. This means $y_2^v w_1^v$ covers h_3h_4, which we have already seen is impossible.

Now suppose that for some $b \in G_B \setminus S_B$, $w_2^b b$ covers h_3h_4, so $c(w_2^b) = h_3$ and $c(b) = h_4$. If $c(b) = h_4$, we must have $c(u_1^b) = h_3$ or $c(u_1^b) = h_5$. But $c(h_0) = h_0$ means $c(u_1^b) = h_1$ or $c(u_1^b) = h_5$, so $c(u_1^b) = h_5$. This implies, since $c(w_1^b) = h_2$ or $c(w_1^b) = h_4$, that $c(w_1^b) = h_4$. But this means that $w_2^b w_1^b$ covers h_3h_4, which we have already excluded as a possibility.

Assume that for some $a \in G_A \setminus S_A$, au_2^a covers h_3h_4, so $c(a) = h_3$ and $c(u_2^a) = h_4$. Because u_1^a and u_2^a are adjacent, $c(u_1^a) = h_3$ or $c(u_1^a) = h_5$, but since u_1^a is adjacent to h_0 and $c(h_0) = h_0$, we have $c(u_1^a) = h_5$. Similarly, $c(w_1^a) = h_2$ or $c(w_1^a) = h_4$, but since w_1^a and u_1^a are adjacent, we have $c(w_1^a) = h_4$. Hence aw_1^a covers h_3h_4, but we have already seen this is impossible.

Now assume that for some $a \in G_A \setminus S_A$, ax_a^{ab} covers h_3h_4, so $c(a) = h_3$ and $c(x_a^{ab}) = h_4$. Now $c(u_1^a) = h_1$ or $c(u_1^a) = h_5$, but since u_1^a and x_a^{ab} are adjacent, we have $c(u_1^a) = h_5$. Because $c(u_2^a)$ must be adjacent to $c(a) = h_3$ as well as $c(u_1^a) = h_5$, we have $c(u_2^a) = h_4$. Hence au_2^a covers h_3h_4, but we have already seen this is impossible.

Suppose that for some $b \in G_B \setminus S_B$, $x_b^{ab} b$ covers h_3h_4, so $c(x_b^{ab}) = h_3$ and $c(b) = h_4$. Now $c(u_1^b) = h_1$ or $c(u_1^b) = h_5$, but since b and u_1^b are adjacent, we must have $c(u_1^b) = h_5$. Because $c(w_1^b)$ must be adjacent to $c(x_b^{ab}) = h_3$, we have $c(w_1^b) = h_2$ or $c(w_1^b) = h_4$. But u_1^b and w_1^b are adjacent, so $c(w_1^b) = h_4$. This means $x_b^{ab} w_1^b$ covers h_3h_4, which we have already ruled out as a possibility.

Now suppose that for some $a \in G_A \setminus S_A$ and some $b \in G_B \setminus S_B$, ab covers h_3h_4, so $c(a) = h_3$ and $c(b) = h_4$. Since u_2^a is adjacent to a and we have seen au_2^a does not cover h_3h_4, we must have $c(u_2^a) = h_2$. Now $c(u_1^a) = h_1$ or $c(u_1^a) = h_5$, but since u_1^a and u_2^a are adjacent, we must have $c(u_1^a) = h_1$. Also, $c(x_a^{ab})$ must be adjacent to $c(u_1^a) = h_1$ and $c(a) = h_3$, so $c(x_a^{ab}) = h_2$. Similarly, $c(x_b^{ab})$ must be adjacent to $c(x_a^{ab}) = h_2$ and $c(b) = h_4$, so $c(x_b^{ab}) = h_3$. But this means $x_b^{ab} b$ covers h_3h_4, which we have already seen is impossible.

Suppose that for some $a \in G_A \setminus S_A$ and some $b \in G_B \setminus S_B$, $x_b^{ab} x_a^{ab}$ covers h_3h_4, so $c(x_b^{ab}) = h_3$ and $c(x_a^{ab}) = h_4$. Since a is adjacent to x_a^{ab} and we have seen ax_a^{ab} does not cover h_3h_4, we must have $c(a) = h_5$. Because $c(b)$ must be adjacent to $c(a) = h_5$ and $c(x_b^{ab}) = h_3$, we have $c(b) = h_4$. But then $x_b^{ab} b$ covers h_3h_4, and we have seen this is impossible.

Lastly, if ah_2 (or ah_4) is an edge of G', assuming $c(a) = h_3$ and $c(h_2) = h_4$ (or $c(a) = h_3$ and $c(h_4) = h_4$) immediately leads us to a contradiction, since $c(h_3) = h_1$.

From all this we obtain that assuming $c(h_3) = h_1$ leads us to the conclusion that no edge of G' covers h_3h_4, contradicting the fact that c is a compaction.

We now show that assuming $c(h_3) = h_5$ leads us to the conclusion that no edge of G' covers h_2h_3.

Let us suppose that for some $v \in V(G) \setminus V(S)$, $y_2^v w_1^v$ covers h_3h_2, so $c(y_2^v) = h_3$ and $c(w_1^v) = h_2$. But $c(h_3) = h_5$, and since h_3 an w_1^v are adjacent, we must have $c(w_1^v) = h_0$ or $c(w_1^v) = h_4$, a contradiction.

By exactly the same argument, we come to the conclusion that none of the edges aw_1^a, $w_2^b w_1^b$, $x_b^{ab} w_1^b$ can cover the edge h_3h_2. A similar argument applies to $y_2^v h_2$.

Suppose that for some $v \in V(G) \setminus V(S)$, $y_2^v y_1^v$ covers h_3h_2, so $c(y_2^v) = h_3$ and $c(y_1^v) = h_2$. Now $c(u_1^v) = h_1$ or $c(u_1^v) = h_5$, but since u_1^v and y_1^v are adjacent we must have $c(u_1^v) = h_1$. Because $c(w_1^v)$ must be adjacent to $c(y_2^v) = h_3$, $c(w_1^v) = h_2$ or $c(w_1^v) = h_4$. But u_1^v is adjacent to w_1^v, so $c(w_1^v) = h_2$. This means $y_2^v w_1^v$ covers h_3h_2, which we have already seen is impossible.

Now suppose that for some $b \in G_B \setminus S_B$, $w_2^b b$ covers h_3h_2, so $c(w_2^b) = h_3$ and $c(b) = h_2$. If $c(b) = h_2$, we must have $c(u_1^b) = h_3$ or $c(u_1^b) = h_1$. But $c(h_0) = h_0$ means $c(u_1^b) = h_1$ or $c(u_1^b) = h_5$, so $c(u_1^b) = h_1$. This implies, since $c(w_1^b) = h_2$ or $c(w_1^b) = h_4$, that $c(w_1^b) = h_2$. But this means that $w_2^b w_1^b$ covers h_3h_2, which we have already excluded as a possibility.

Assume that for some $a \in G_A \setminus S_A$, au_2^a covers h_3h_2, so $c(a) = h_3$ and $c(u_2^a) = h_2$. Because u_1^a and u_2^a are adjacent, $c(u_1^a) = h_3$ or $c(u_1^a) = h_1$, but since u_1^a is adjacent to h_0 and $c(h_0) = h_0$, we have $c(u_1^a) = h_1$. Similarly, $c(w_1^a) = h_2$ or $c(w_1^a) = h_4$, but since w_1^a and u_1^a are adjacent, we have $c(w_1^a) = h_2$. Hence aw_1^a covers h_3h_2, but we have already seen this is impossible.

Now assume that for some $a \in G_A \setminus S_A$, ax_a^{ab} covers h_3h_2, so $c(a) = h_3$ and $c(x_a^{ab}) = h_2$. Now $c(u_1^a) = h_1$ or $c(u_1^a) = h_5$, but since u_1^a and x_a^{ab} are adjacent, we have $c(u_1^a) = h_1$. Because $c(u_2^a)$ must be adjacent to $c(a) = h_3$ as well as $c(u_1^a) = h_1$, we have $c(u_2^a) = h_2$. Hence au_2^a covers h_3h_2, but we have already seen this is impossible.

Suppose that for some $b \in G_B \setminus S_B$, $x_b^{ab} b$ covers h_3h_2, so $c(x_b^{ab}) = h_3$ and $c(b) = h_2$. Now $c(u_1^b) = h_1$ or $c(u_1^b) = h_5$, but since b and u_1^b are adjacent, we must have $c(u_1^b) = h_1$. Because $c(w_1^b)$ must be adjacent to $c(x_b^{ab}) = h_3$, we have $c(w_1^b) = h_2$ or $c(w_1^b) = h_4$. But u_1^b and w_1^b are adjacent, so $c(w_1^b) = h_2$. This means $x_b^{ab} w_1^b$ covers h_3h_2, which we have already ruled out as a possibility.

Now suppose that for some $a \in G_A \setminus S_A$ and some $b \in G_B \setminus S_B$, ab covers h_3h_2, so $c(a) = h_3$ and $c(b) = h_2$. Since u_2^a is adjacent to a and we have seen au_2^a does not cover h_3h_2, we must have $c(u_2^a) = h_4$. Now $c(u_1^a) = h_1$ or $c(u_1^a) = h_5$, but since u_1^a and u_2^a are adjacent, we must have $c(u_1^a) = h_5$. Also, $c(x_a^{ab})$ must be adjacent to $c(u_1^a) = h_5$ and $c(a) = h_3$, so $c(x_a^{ab}) = h_4$. Similarly, $c(x_b^{ab})$ must be adjacent to $c(x_a^{ab}) = h_4$ and $c(b) = h_2$, so $c(x_b^{ab}) = h_3$. But this means $x_b^{ab}b$ covers h_3h_2, which we have already seen is impossible.

Suppose that for some $a \in G_A \setminus S_A$ and some $b \in G_B \setminus S_B$, $x_b^{ab}x_a^{ab}$ covers h_3h_2, so $c(x_b^{ab}) = h_3$ and $c(x_a^{ab}) = h_2$. Since a is adjacent to x_a^{ab} and we have seen ax_a^{ab} does not cover h_3h_2, we must have $c(a) = h_1$. Because $c(b)$ must be adjacent to $c(a) = h_1$ and $c(x_b^{ab}) = h_3$, we have $c(b) = h_2$. But then $x_b^{ab}b$ covers h_3h_2, and we have seen this is impossible.

Lastly, if ah_2 (or ah_4) is an edge of G', assuming $c(a) = h_3$ and $c(h_2) = h_2$ (or $c(a) = h_3$ and $c(h_4) = h_2$) immediately leads us to a contradiction, since $c(h_3) = h_5$.

From all this we obtain that assuming $c(h_3) = h_5$ leads us to the conclusion that no edge of G' covers h_3h_2, contradicting the fact that c is a compaction.

From all the above we obtain that $c(h_3) = h_3$, which means that $c(W) = \{h_2, h_3, h_4\}$.

Now we show $c(h_1) \neq c(h_5)$. To the contrary, assume $c(h_1) = c(h_5)$. Since $c(h_0) = h_0$, we have $c(h_1), c(h_5) \in \{h_1, h_5\}$. Due to symmetry, we can without loss of generality assume $c(h_1) = c(h_5) = h_1$. Since $c(U) = \{h_1, h_0, h_5\}$, it must be the case that $c(u_1^v) = h_5$ for some $v \in V(G) \setminus V(S)$. Now $c(w_1^v)$ and $c(h_2)$ must both be adjacent to $c(h_3) = h_3$, so $c(w_1^v), c(h_2) \in \{h_2, h_4\}$. Because $c(u_1^v) = h_5$ and u_1^v and w_1^v are adjacent, $c(w_1^v) = h_4$. Similarly, because $c(h_0) = h_0$ and h_1 and h_2 are adjacent, $c(h_2) = h_2$. Now $c(y_2^v)$ must be adjacent to $c(h_2) = h_2$ and $c(w_1^v) = h_4$, so $c(y_2^v) = h_3$. Also, $c(y_1^v)$ must be adjacent to $c(h_5) = h_1$ and $c(u_1^v) = h_5$, so $c(y_1^v) = h_0$. Thus we have that y_1^v and y_2^v are adjacent in G', but $c(y_1^v) = h_0$ and $c(y_2^v) = h_3$ are not adjacent in C_6, a contradiction.

Hence $c(h_1) \neq c(h_5)$. That is, $c(\{h_1, h_5\}) = \{h_1, h_5\}$. Without loss of generality, we can take $c(h_1) = h_1$ and $c(h_5) = h_5$. Since $c(h_3) = h_3$, we have $c(h_2), c(h_4) \in \{h_2, h_4\}$. Because h_1 and h_2 are adjacent in G' and the distance between $c(h_1) = h_1$ and h_4 in C_6 is 3, it must be that $c(h_2) \neq h_4$ and so $c(h_2) = h_2$. Similarly, because h_5 and h_4 are adjacent in G' and the distance between $c(h_5) = h_5$ and h_2 in C_6 is 3, it must be that $c(h_4) \neq h_2$, and so $c(h_4) = h_4$.

Thus $c(h_i) = h_i$ for all $i = 0, 1, \ldots, 5$, and $c : V(G') \to V(C_6)$ is a retraction. □

The last claim is a simple observation that completes the proof of $(*)$ and thus also of

Theorem 3.5.

Claim 3.11

Suppose G' is pinchable to C_6. Then G retracts to S.

Proof. By Claims 3.9 and 3.10 we know there exists a retraction $r : V(G') \to V(S)$. Because S is an induced subgraph of G, and G is an induced subgraph of G', restricting r to G gives us what we need. $\square$

3.3 A polynomial time algorithm for 3-MIXING for planar graphs

In this section, we prove the following.

Theorem 3.12

Restricted to planar *bipartite graphs, the decision problem* 3-MIXING *is in the complexity class* P.

To prove the theorem we need two lemmas.

Lemma 3.13

Let P be a shortest path between distinct vertices u and v in a connected bipartite graph H. Then H is pinchable to P.

Proof. Let P have vertices $u = v_0, v_1, \ldots, v_{k-1}, v_k = v$, and let T be a breadth-first spanning tree of H rooted at u that contains P (we can choose T so that it contains P since P is a shortest path). Now, working in T, pinch all vertices at distance one from u to v_1. Next pinch all vertices at distance two from u to v_2. Continue until all vertices at distance k from u are pinched to $v_k = v$. If necessary, arbitrary pinches on the vertices at distance at least $k + 1$ from u will yield P. $\square$

In the following, when we say some vertices of a graph are *properly precoloured*, we just mean that they are assigned colours in a way such that the subgraph induced by these vertices is properly coloured.

Lemma 3.14

Let H be a bipartite graph, and suppose the vertices of a 4-cycle in H are properly precoloured using colours from $\{1,2,3\}$. Then this 3-colouring can be extended to a proper 3-colouring of H.

Proof. Since any 3-colouring of a four cycle C_4 has two vertices with the same colour, we can without loss of generality assume the four vertices are coloured 1-2-1-2 or 1-2-1-3. In the first instance, since H is bipartite, we can extend the precolouring to a colouring of H using colours 1 and 2 only. For the second case, we can use the same colouring, except leaving the vertex coloured 3 as it is. $\square$

The sequence of claims that follows outlines an algorithm that, given a connected bipartite planar graph G as input, determines in polynomial time whether or not G is 3-mixing. We first show how we can take the input graph to be 2-connected.

Claim 3.15

Let G be a connected bipartite planar graph, and suppose that G has a cut-vertex v. Let H_1 be a component of $G - \{v\}$. Denote by G_1 the subgraph of G induced by $V(H_1) \cup \{v\}$, and let G_2 be the subgraph induced by $V(G) \setminus V(H_1)$. Then G is 3-mixing if and only if both G_1 and G_2 are 3-mixing.

Proof. If G is 3-mixing, then clearly so are G_1 and G_2. Conversely, if G is not 3-mixing, we know by Theorem 3.1 that there must exist a 3-colouring α of G and a cycle C in G such that $W(\overrightarrow{C}, \alpha) \neq 0$. But because C must lie completely in G_1 or G_2, we have that G_1 or G_2 is not 3-mixing. $\square$

Let us now consider an embedding of our 2-connected bipartite planar graph G in the plane, and let us identify G with this embedding. (Throughout the rest of this section, we will usually, for ease of reference, identify a planar graph with a given embedding of the graph in the plane.) Given a cycle D in G, denote by $\mathrm{Int}(D)$ and $\mathrm{Ext}(D)$ the sets of vertices inside and outside of D, respectively. Note that the vertices of D itself are not included in $\mathrm{Int}(D)$ nor in $\mathrm{Ext}(D)$. If both $\mathrm{Int}(D)$ and $\mathrm{Ext}(D)$ are non-empty, D is said to be *separating*. For D a separating cycle in G, let us write $G_{\mathrm{Int}}(D) = G - \mathrm{Ext}(D)$ and $G_{\mathrm{Ext}}(D) = G - \mathrm{Int}(D)$, and note that D is part of both these graphs.

We now consider the case that the planar embedding of G has a separating 4-cycle.

Claim 3.16

Let G be a 2-connected bipartite planar graph, and suppose that G has a planar embedding with a separating 4-cycle D. Then G is 3-mixing if and only if $G_{\mathrm{Int}}(D)$ and $G_{\mathrm{Ext}}(D)$ are both 3-mixing.

Proof. To prove necessity, we show that if one of $G_{\mathrm{Int}}(D)$ or $G_{\mathrm{Ext}}(D)$ is not 3-mixing, then G is not 3-mixing. Without loss of generality, suppose that $G_{\mathrm{Int}}(D)$ is not 3-mixing, so there exists a 3-colouring α of $G_{\mathrm{Int}}(D)$ and a cycle C in $G_{\mathrm{Int}}(D)$ with $W(\overrightarrow{C}, \alpha) \neq 0$. By Lemma 3.14, the 3-colouring of the vertices of the 4-cycle D can be extended to a 3-colouring of $G_{\mathrm{Ext}}(D)$. The combination of the 3-colourings of $G_{\mathrm{Int}}(D)$ and $G_{\mathrm{Ext}}(D)$ gives a 3-colouring of G with a non-zero weight cycle, showing that G is not 3-mixing.

To prove sufficiency, we show that if G is not 3-mixing, then at least one of $G_{\mathrm{Int}}(D)$ and $G_{\mathrm{Ext}}(D)$ must fail to be 3-mixing. Suppose that α is a 3-colouring of G for which there is a cycle C with $W(\overrightarrow{C}, \alpha) \neq 0$. If C is contained entirely within $G_{\mathrm{Int}}(D)$ or $G_{\mathrm{Ext}}(D)$ we are done, so let us assume that C has some vertices in $\mathrm{Int}(D)$ and some in $\mathrm{Ext}(D)$. Then applying Lemma 3.4 (repeatedly, if necessary) we can find a cycle C' contained entirely in $G_{\mathrm{Int}}(D)$ or $G_{\mathrm{Ext}}(D)$ for which $W(\overrightarrow{C'}, \alpha) \neq 0$, completing the proof. □

We need two further claims to complete the description of our algorithm. We call a face of G with k edges in its boundary a *k-face*, and a face with at least k edges in its boundary a *$\geq k$-face*. The number of ≥ 6-faces of G—which we can now assume has no separating 4-cycle—will in fact determine if G is 3-mixing.

Claim 3.17

Let G be a 2-connected bipartite planar graph. Suppose that G has a planar embedding with no separating 4-cycle, and suppose that every internal face of the embedding is a 4-face. Then G is 3-mixing.

Proof. Let α be any 3-colouring of G and let C be any cycle in G. We show $W(\overrightarrow{C}, \alpha) = 0$ by induction on the number of faces inside C. If there is just one face inside C, C is in fact a facial 4-cycle and $W(\overrightarrow{C}, \alpha) = 0$. For the inductive step, let C be a cycle with $r \geq 2$ faces in its interior. If, for two consecutive vertices u, v of C, we have vertices $a, b \in \mathrm{Int}(C)$ together with edges ua, ab, bv in G, let C' be the cycle formed from C by the removal of the edge uv and the addition of edges ua, ab, bv. If not, check whether for three consecutive vertices u, v, w of C, there is a vertex $a \in \mathrm{Int}(C)$ with edges ua, aw in G. If so, let C' be the cycle formed from C by the removal of the vertex v and the addition of the edges ua, aw. If neither of the previous two cases apply, we must have, for u, v, w, x four consecutive vertices of C, an edge ux inside C. In such a case, let C' be the cycle formed from C by the removal of vertices v, w and the addition of the edge ux. In all cases we have that C' has $r-1$ faces in its interior, so, by induction, we can assume $W(\overrightarrow{C'}, \alpha) = 0$. From Lemma 3.4 we then obtain $W(\overrightarrow{C}, \alpha) = 0$. □

Claim 3.18

Let G be a 2-connected bipartite planar graph. Suppose that G has a planar embedding with no separating 4-cycle, and suppose further that the embedding has an internal ≥ 6-face, and that the outer face is a ≥ 6-face. Then G is not 3-mixing.

Proof. We claim that G, under the given assumptions, is pinchable to C_6. Denote the internal ≥ 6-face by f, and the outer face by f_o. We call a cycle D in G *f-separating* if f lies inside D, where we include the possibility that edges on the boundary of f lie on the cycle D. (Note that the cycle bounding f_o is always an f-separating cycle, and thus an f-separating cycle need not be a separating cycle.) Obviously G contains no f-separating 4-cycle, since such a cycle would constitute a separating 4-cycle. We now claim that if G is not a cycle, then it is possible to find a sequence of one or more pinches so that the resulting graph is a planar graph that has an internal ≥ 6-face f', whose outer face is a ≥ 6-face, and without an f'-separating 4-cycle. (Note that bipartiteness is trivially maintained by pinching.) Repeating such a sequence of pinches will eventually transform G into a cycle of length at least six, proving that G is not 3-mixing.

Let C be the cycle that bounds f: we will initially attempt to pinch vertices into C. Let x, y, z be three consecutive vertices of C with y having degree at least 3; if there is no such vertex y, then G is simply a cycle of length at least six and we are done. Let a be a neighbour of y distinct from x and z, such that the edges ya and yz form part of the boundary of a face adjacent to f.

Suppose the result of pinching a and z introduces no f-separating 4-cycle. If so, we pinch a and z. Note that the resulting graph still contains the internal ≥ 6-face f, and is planar since the edges ya and yz form part of a common face. Note also that the outer face, though it might have decreased in size, remains a ≥ 6-face: if it did not—so the edges ya and yz were originally part of the boundary of f_o, which had length six—then we would have a contradiction to the fact that pinching a and z introduced no f-separating 4-cycle. We observe that pinching a and z might well introduce a cut-vertex into the graph, but that as long as such a vertex is not included twice on the boundary of the outer face, this is not a problem. (Note that such a situation cannot arise for the internal face f.) If we do find that the boundary of the outer face now includes a vertex v twice, then let us denote by G' the graph resulting from pinching a and z. Let us also denote by C'_o and C''_o the two distinct cycles formed by the boundary of the outer face, with $V(C'_o) \cap V(C''_o) = \{v\}$, and where $G'_{\text{Int}}(C'_o)$ is the subgraph of G' containing the internal face f (so C'_o must have length at least six, for otherwise we have introduced an f-separating 4-cycle). Now, considering an edge vw of C''_o, we pinch $G'_{\text{Int}}(C''_o)$ to vw (using Lemma 3.13 and the fact that vw is a shortest path between v and w). Using this same sequence of pinches in G', followed by pinching vw into C'_o, leaves us with a graph with the required invariants, and every vertex on the boundary of the outer face of the resulting graph distinct.

Suppose pinching a and z *does* result in the creation of an f-separating 4-cycle. If so, this must be because the path a, y, z forms part of an f-separating 6-cycle D. We now show how we can find alternative pinches which do not introduce an f-separating 4-cycle. The fact that D is f-separating means there is a path $P \subseteq D$ of length 4 between a and z. Note that P cannot contain y, for this would contradict the fact that G has no f-separating 4-cycle. Consider the graph $G' = G_{\text{Int}}(D) - \{yz\}$. We claim that the path $P' = P \cup \{y\}$ is a shortest path between y and z in G'. To see this, remember that G is bipartite, so any path between y and z in G has to have odd length. We cannot have another edge $yz \in E(G')$ since G is simple. Now note that any path between y and z in G', together with the edge yz, forms an f-separating cycle in G. Hence a path of length 3 between y and z would contradict the fact that G has no f-separating 4-cycle, and so P' is indeed a shortest path between y and z in G'. Using Lemma 3.13, we see that G' is pinchable to P'. Using the same sequence of pinches in G will pinch $G_{\text{Int}}(D)$ into D. Note this introduces no separating 4-cycle into the resulting graph, and note also that this graph is planar, since it is a subgraph of G. Moreover, note that the length of the cycle bounding the outer face remains the same, that the vertices of this cycle are all distinct, and that the cycle D now bounds an internal 6-face. It follows that this sequence of pinches is a sequence as required by the claim. This completes the proof. □

The algorithm that decides 3-MIXING runs as follows. Given a connected bipartite planar graph G with n vertices, we first find the blocks of G. (A *block* of G is a maximal connected subgraph of G with no cut-vertex.) These can be found by a standard depth-first search method (see, for example, [61, p. 157]) in time $O(n)$. Note that a block which is not 2-connected is either a K_1 or a K_2, which are both trivially 3-mixing.

Next, for each 2-connected component H of G, we perform the following procedure. Find an embedding of H in the plane. Let us recall that a planar embedding of a graph can be specified by a *combinatorial embedding* (a list of adjacencies for each vertex, with adjacencies listed as they are found in a clockwise order around the vertex) together with the specification of its outer face. There are fast algorithms, for instance the linear time algorithm of Mohar [50], to find such an embedding. We now check whether the embedding has a ≥ 6-face, by traversing the edges of H as they form faces, using the adjacency lists (this will take time at most $O(n_H^2)$, where n_H is the number of vertices of H). If the embedding has a ≥ 6-face, then we transform the embedding into an embedding in which this face is the outer face. This is done by reversing the order of vertices in each adjacency list and specifying the new face as the outer face, taking time $O(n_H)$. We now check whether or not the embedding of H has a separating 4-cycle. A naive approach, which checks all subsets of 4 elements of $V(H)$, runs as follows. First, check whether a given 4-tuple forms a cycle, using the adjacency lists. If so, we check whether or not it has an empty interior (note that it will always have non-empty exterior, where the outer ≥ 6-face is) by checking whether or not, for each vertex of the 4-cycle, the edges of the cycle are consecutive in the cyclic ordering of neighbours defining the embedding. This will take $O(n_H^5)$ time: $O(n_H^4)$ to enumerate all 4-tuples, multiplied by $O(n_H)$, the time needed to check whether, for a given 4-tuple, we have a cycle and whether this has a non-empty interior. If H does have a separating 4-cycle, we apply Claim 3.16 and recurse on two smaller problems. If it does not, then we check for a ≥ 6-face different from the outside face (adding $O(n_H^2)$ to the running time), and then either Claim 3.17 or 3.18 must apply to H. If at any stage in the process, for some H, Claim 3.18 applies, then the algorithm returns 'no'. Otherwise, the algorithm concludes that G is 3-mixing.

If we denote the running time of the procedure we are running on H by $T(n_H)$, the recursive call arising from finding a separating 4-cycle D leads to the recurrence relation $T(n_H) = T(n_{\text{Int}}) + T(n_{\text{Ext}}) + O(n_H^5)$, where n_{Int} and n_{Ext} are, respectively, the number of vertices of $H_{\text{Int}}(D)$ and $H_{\text{Ext}}(D)$. Noting that $n_H = n_{\text{Int}} + n_{\text{Ext}} - 4$, we may rewrite this as $T(n_H) = T(n_{\text{Int}}) + T(n_H - n_{\text{Int}} + 4) + O(n_H^5)$, and because we have $5 \leq n_{\text{Int}} \leq n_H - 1$, we see that we are in fact recursing on two smaller problems. After observing that $T(5) = c$ for some constant c, a simple inductive argument yields that $T(n_H)$ is $O(n_H^6)$. Because we have less than n blocks in G, the running time of the algorithm is bounded by $O(n^7)$. This completes the proof of Theorem 3.12.

4

Paths between 3-colourings

In this chapter we examine what can be said about possible sequences of recolourings between a given pair of 3-colourings. We determine how easy it is to find if a sequence exists, and also what its length may be if it does. Our main result is the following.

Theorem 4.1
The decision problem 3-COLOUR PATH *is in the complexity class* P.

We prove Theorem 4.1 in Section 4.1 by describing an algorithm that decides the problem in polynomial time. In doing so, we will see that in the case that two 3-colourings of a graph G belong to the same component of $\mathcal{C}_3(G)$, our algorithm can be used to exhibit a path of length $O(|V(G)|^2)$ between them. This proves the following.

Theorem 4.2
Let G be a 3-colourable graph with n vertices. Then the diameter of any component of $\mathcal{C}_3(G)$ is $O(n^2)$.

In Section 4.2 we turn our attention to what else can be said about the distance between a given pair of 3-colourings. We will prove that in many cases, the algorithm of Section 4.1 in fact returns a shortest path between two 3-colourings which are connected in $\mathcal{C}_3(G)$. We will also show that the quadratic bound on the number of recolourings can be met from below, constructing a class of instances G, α, β such that α and β are connected and at distance $\Omega(|V(G)^2|)$ in $\mathcal{C}_3(G)$.

4.1 A polynomial time algorithm for 3-COLOUR PATH

The algorithm that decides 3-COLOUR PATH stems from the proof of a characterisation of instances G, α, β where α and β belong to the same component of $\mathcal{C}_3(G)$. We will describe this characterisation in Theorem 4.6 below. Before doing so, we examine what can forbid the existence of a path between 3-colourings α and β of a graph G in $\mathcal{C}_3(G)$. The proof of the characterisation of connected pairs of 3-colourings is via an algorithm that, given G, α, β, either finds a sequence of recolourings between α and β, or exhibits a structure which proves that no such sequence exists. Thus this algorithm also decides 3-COLOUR PATH.

Obstructions to paths between 3-colourings

Let us examine what can stop us from being able to find a sequence of recolourings between a pair of 3-colourings α, β of a graph G. Informally, we call a structure in G, α, β forbidding the existence of a path between α and β in $\mathcal{C}_3(G)$ an *obstruction*. For the remainder of this section we assume that we are dealing with some fixed graph G.

We saw in Lemma 2.10 in Chapter 2 how a cycle C in G can act as an obstruction between α and β: if its weight $W(\vec{C}, \alpha)$ in α is different to that in β, $W(\vec{C}, \beta)$, then there can be no path between α and β in $\mathcal{C}_3(G)$.

A second obstruction is given by what we call *fixed vertices*. For a 3-colouring α, we define a vertex v as *fixed* if there is no sequence of recolourings from α which will allow us to recolour v. In other words, a vertex v is fixed if for every colouring β in the same component of $\mathcal{C}_3(G)$ as α we have $\beta(v) = \alpha(v)$. For example, if a cycle with 0 mod 3 vertices is coloured 1-2-3-1-2-3-$\cdots$-1-2-3, then every vertex on the cycle is fixed (as none can be the first to be recoloured); we call this a *fixed cycle* (with respect to the 3-colouring α). Similarly, a path coloured $\cdots$3-1-2-3-1-2-3-1-$\cdots$, both of whose end-vertices lie on fixed cycles, cannot be recoloured and is called a *fixed path*.

Given a 3-colouring α of G, we denote the set of fixed vertices of G by F_α. We shall shortly prove the following.

Proposition 4.3
Let α be a 3-colouring of G. Then every $v \in F_\alpha$ belongs to a fixed cycle or a fixed path.

The next lemma, which illustrates how fixed vertices may act as an obstruction, follows immediately from the definitions.

Lemma 4.4

Let α and β be two 3-colourings of G. Then if α and β belong to the same component of $\mathcal{C}_3(G)$, we must have $F_\alpha = F_\beta$ and $\alpha(v) = \beta(v)$ for each $v \in F_\alpha$. □

The following lemma, very similar to Lemma 2.10, shows a third type of obstruction.

Lemma 4.5

Let α and β be 3-colourings of G with $F_\alpha = F_\beta \neq \varnothing$ and $\alpha(v) = \beta(v)$ for all $v \in F_\alpha$, and suppose that G contains a path P with end-vertices u and w, where $u, w \in F_\alpha$. Then if α and β are in the same component of $\mathcal{C}_3(G)$, we must have $W(\vec{P}, \alpha) = W(\vec{P}, \beta)$.

Proof. Let α and α' be 3-colourings of G that are adjacent in $\mathcal{C}_3(G)$, and suppose the two 3-colourings differ on vertex v. Note that v cannot be a vertex in F_α, so neither can it be an end-vertex of P. If v is not on P, then we certainly have $W(\vec{P}, \alpha) = W(\vec{P}, \alpha')$.

If v is an internal vertex of P, then all its neighbours must have the same colour in α, for otherwise we would not be able to recolour v. If we denote the in-neighbour of v on $\vec{P}$ by v_i and its out-neighbour by v_o, then this means that $w(\overrightarrow{v_iv}, \alpha)$ and $w(\overrightarrow{vv_o}, \alpha)$ have opposite sign, hence $w(\overrightarrow{v_iv}, \alpha) + w(\overrightarrow{vv_o}, \alpha) = 0$. Recolouring vertex v will change the signs of the weights of the oriented edges $\overrightarrow{v_iv}$ and $\overrightarrow{vv_o}$, but they will remain opposite. Therefore $w(\overrightarrow{v_iv}, \alpha') + w(\overrightarrow{vv_o}, \alpha') = 0$, and it follows that $W(\vec{P}, \alpha) = W(\vec{P}, \alpha')$.

From the above we immediately obtain that the weight of an oriented path between fixed vertices is constant on all 3-colourings in the same component of $\mathcal{C}_3(G)$. □

Lemmas 2.10, 4.4 and 4.5 give necessary conditions for two 3-colourings α and β of a graph G to belong to the same component of $\mathcal{C}_3(G)$. From Lemmas 4.4 and 2.10 we obtain, respectively:

(C1) $F_\alpha = F_\beta$ and $\alpha(v) = \beta(v)$ for each $v \in F_\alpha$; and

(C2) for every cycle C in G, $W(\vec{C}, \alpha) = W(\vec{C}, \beta)$.

If for two 3-colourings α and β of G we take condition (C1) to be satisfied, Lemma 4.5 gives a third necessary condition for α and β to belong to the same component of $\mathcal{C}_3(G)$:

(C3) for every path P between fixed vertices, $W(\vec{P}, \alpha) = W(\vec{P}, \beta)$.

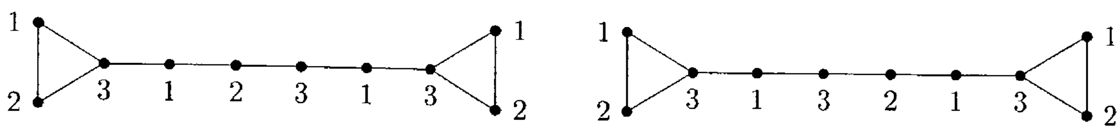

Figure 4.1 Two 3-colourings of a graph G not connected in $\mathcal{C}_3(G)$.

Bearing in mind that we are only considering condition (C3) if condition (C1) is already satisfied, let us observe that neither conditions (C1) and (C2) taken together, nor conditions (C1) and (C3) taken together, are sufficient to guarantee the existence of a path between 3-colourings α and β.

To see that conditions (C1) and (C2) are not sufficient, consider the graph and two 3-colourings shown in Figure 4.1. It is easy to check that (C1) and (C2) are satisfied (note that only vertices on the 3-cycles are fixed), but the two colourings are not connected: fix an orientation of the path between the two 3-cycles, and observe that the weight of this oriented path is $+3$ in one colouring and -3 in the other.

To see that conditions (C1) and (C3) are not sufficient, consider two 3-colourings α and β of a 5-cycle that differ only in that the colours 1 and 2 are swapped: (C1) and (C3) are satisfied (since $F_\alpha = F_\beta = \varnothing$), but there is no path between the two colourings as the 5-cycle has different weights in the two colourings.

We now prove that if all three conditions are satisfied by a pair of colourings α and β of G, then they are in the same component of $\mathcal{C}_3(G)$.

A characterisation of connected pairs of 3-colourings

The proof of the following characterisation of connected pairs of 3-colourings will yield a polynomial time algorithm for 3-COLOUR PATH, proving Theorem 4.1. We will also prove Theorem 4.2 in the process.

Theorem 4.6

Two 3-colourings α and β of a graph G belong to the same component of $\mathcal{C}_3(G)$ if and only if

(C1) $F_\alpha = F_\beta$ and $\alpha(v) = \beta(v)$ for each $v \in F_\alpha$;

(C2) for every cycle C in G, $W(\vec{C}, \alpha) = W(\vec{C}, \beta)$; and

(C3) for every path P between fixed vertices, $W(\overrightarrow{P},\alpha) = W(\overrightarrow{P},\beta)$.

The necessity of the three conditions has already been established. We prove that they are sufficient by outlining an algorithm whose input is a graph G and two 3-colourings α and β of G, and whose output is either a path in $\mathcal{C}_3(G)$ from α to β, or an obstruction that shows that (C1), (C2) or (C3) is not satisfied, so no such path exists.

The first step of the algorithm is to find F_α and F_β. We claim that the following procedure finds the fixed vertices of a graph G with 3-colouring α.

- Let S_1, S_2 and S_3 initially be the three colour classes induced by α.
- For $i \in \{1,2,3\}$, and for each vertex $v \in S_i$: let $S_i = S_i \setminus \{v\}$ unless v has neighbours in each of the other two sets.
- Repeat the previous step until no further changes are possible. Return $S = S_1 \cup S_2 \cup S_3$.

Claim 4.7

The above procedure returns $S = F_\alpha$.

Before proving the claim, let us give some definitions. Fix a vertex v of G and set $L_0^+ = L_0^- = \{v\}$. For $i = 1,2,\ldots$, let a vertex u belong to L_i^+ if u has a neighbour $w \in L_{i-1}^+$ and $\alpha(u) \equiv \alpha(w) + 1 \pmod 3$. (So, for example, if v is coloured 3, then L_1^+ contains all neighbours of v coloured 1, L_2^+ contains all vertices coloured 2 that have a neighbour in L_1^+, and so on.) For $j = 1,2,\ldots$, let a vertex u belong to L_j^- if u has a neighbour $w \in L_{j-1}^-$ and $\alpha(u) \equiv \alpha(w) - 1 \pmod 3$. We call these sets the *levels* of v, and the sets can be categorised as *positive* or *negative* according to their superscript.

Observe that v lies on a fixed cycle if and only if there is a vertex $u \in L_i^+ \cap L_j^-$, for some $i,j > 0$. To see this, note that if v lies on a fixed cycle C, then $v \in L_p^+(v)$ and $v \in L_p^-(v)$, where p is the number of edges of C. Hence there is a $u \in L_i^+ \cap L_j^-$, for some $i,j > 0$. Conversely, if we have a $u \in L_i^+ \cap L_j^-$ for some $i,j > 0$, then there is a path P^+ in G formed by vertices $v = p_0^+, p_1^+, \ldots, p_i^+ = u$, where $p_k^+ \in L_k^+(v)$ for $0 \le k \le i$, and there is also a path P^- in G formed by vertices $v = p_0^-, p_1^-, \ldots, p_j^- = u$, where $p_k^- \in L_k^-(v)$ for $0 \le k \le j$. Note that we can assume that u is distinct from all of $p_0^+, p_1^+, \ldots, p_{i-1}^+$ and $p_0^-, p_1^-, \ldots, p_{j-1}^-$; that is, we choose i and j to be as small as possible. Then the graph induced by $P^+ \cup P^-$ forms a fixed cycle of G in α.

Similarly, v lies on a fixed path with end vertices u and w (each on a fixed cycle) if and only if $u \in L_{i'}^+ \cap L_i^+$ for some $i' > i > 0$ and $w \in L_{j'}^- \cap L_j^-$ for some $j' > j > 0$. To see this, first observe that if v lies on a fixed path with end-vertices u and w, we can conclude, without loss of generality, that $u \in L_i^+$ and $w \in L_j^-$ for some $i, j > 0$. Then, because u and w are each part of a fixed cycle, by the argument above we have that $u \in L_{i'}^+$ for $i' = i + p$, and $w \in L_{j'}^-$ for $j' = j + q$, where p and q are the respective lengths of the fixed cycles of u and w. Hence $u \in L_i \cap L_{i'}$, where $i' > i > 0$, and $w \in L_j \cap L_{j'}$, where $j' > j > 0$. For the converse, suppose we have $u \in L_i \cap L_{i'}$, for some $i' > i > 0$, and $w \in L_j \cap L_{j'}$, for some $j' > j > 0$. Ensuring that $i' > i > 0$ and $j' > j > 0$ are all chosen as small as possible, we can then consider a sequence of vertices $v = p_0^+, p_1^+, \dots, p_i^+ = u, p_{i+1}^+, \dots, p_{i'}^+ = u$, where $p_k^+ \in L_k^+(v)$ for $0 \leq k \leq i'$, and such that $p_0^+, \dots, p_i^+$ form a fixed path and $p_i^+, \dots, p_{i'-1}^+$ induce a fixed cycle. We can choose a similar sequence of vertices from the negative levels of v which includes w (twice) to complete the proof.

Proof of Claim 4.7 (and Proposition 4.3). Suppose the procedure described above is run on G, α, and has terminated. Note that a vertex that lies on a fixed cycle or path is certainly in S. We shall show that for each vertex $v \in V(G)$, either

- v lies on a fixed cycle or path (so is both fixed and in S), or
- v is neither fixed nor in S.

This will prove that $S = F_\alpha$, and also Proposition 4.3.

Fix a vertex v of G and consider the levels of v. We have observed that if there is a vertex that is in L_i^+, for some $i > 0$, and also in L_j^-, for some $j > 0$, then v lies on a fixed cycle. Also, if there is a vertex that belongs to L_i^+ and $L_{i'}^+$, for some $i' > i > 0$, and another vertex that belongs to L_j^- and $L_{j'}^-$, for some $j' > j > 0$, then v lies on a fixed path.

If neither of these two properties hold, then either the positive or negative levels (or both) are disjoint and thus only finitely many of them are non-empty. We show that this means we can recolour v and so v is not fixed. Let us assume therefore that $L_t^+ = \varnothing$ or $L_t^- = \varnothing$ for some $t > 0$. Without loss of generality, let us assume $L_t^+ = \varnothing$. Thus each vertex $u \in L_{t-1}^+$ can be recoloured with $\alpha(u) + 1 \pmod 3$. Then each vertex $w \in L_{t-2}^+$ can be recoloured with $\alpha(w) + 1 \pmod 3$, and so on, until v is recoloured. The fact that v can be recoloured implies it is not in S: every vertex in S has a pair of differently coloured neighbours, so no vertex in S can be the first to be recoloured. $\square$

Claim 4.7 allows us to find F_α and F_β. If $F_\alpha \neq F_\beta$, or if there is a vertex $v \in F_\alpha$ such that $\alpha(v) \neq \beta(v)$, then there is no path from α to β. The algorithm outputs F_α, F_β and, if necessary, v.

Henceforth we assume that condition (C1) is satisfied, so $F_\alpha = F_\beta$ and for all $v \in F_\alpha$, $\alpha(v) = \beta(v)$.

If $F_\alpha \neq \varnothing$, we construct, from G, a new graph G^f by identifying, for $i = 1, 2, 3$, all vertices in S_i and denoting the newly created vertex by f_i. In other words:

- $V(G^f) = (V(G) \setminus F_\alpha) \cup \{f_1, f_2, f_3\}$, and
- $$E(G^f) = \{uv \in E(G) \mid u, v \in V(G) \setminus F_\alpha\} \cup \{f_1f_2, f_1f_3, f_2f_3\}$$
$$\cup \bigcup_{i=1,2,3} \{uf_i \mid u \in V(G) \setminus F_\alpha \text{ and } \exists v \in S_i \text{ with } uv \in E(G)\}.$$

If G has no fixed vertices with respect to α, then we set $G^f = G$.

It is convenient to assume that all edges are retained so that G and G^f have the same edge set. Since S_1, S_2, S_3 are independent sets (they are subsets of the colour classes of the colouring α), this means G^f is a graph with possibly multiple edges, but no loops. Let α^f and β^f be the colourings induced on G^f by α and β. It is easy to observe that if $F_\alpha \neq \varnothing$,

- f_1, f_2 and f_3 are the only fixed vertices of G^f in α^f and β^f, and
- f_1, f_2 and f_3 induce a (fixed) 3-cycle in G^f in both colourings.

Note that if α and β belong to the same component of $C_3(G)$, this component is isomorphic to the component of $C_3(G^f)$ that contains α^f and β^f. Hence we have the following.

Claim 4.8
There is a path from α to β in $C_3(G)$ if and only if there is a path from α^f to β^f in $C_3(G^f)$.

To prove Theorem 4.6, we shall prove the following claim.

Claim 4.9
Two 3-colourings α^f and β^f of a graph G^f belong to the same component of $C_3(G^f)$ if and only if

(C2') for every cycle C in G^f, $W(\vec{C}, \alpha^f) = W(\vec{C}, \beta^f)$.

Let us first establish that the claim implies the theorem, recalling that we are assuming condition (C1). Let $\vec{C}$ be an oriented cycle in G. In G^f, the oriented edges of $\vec{C}$ form a set of edge-disjoint oriented cycles. (Here we are using the convention that all edges from G are retained in G^f.) Since these cycles contain the same edges as $\vec{C}$, similarly oriented, it is easy to see that the sum of the weights of these cycles is equal to $W(\vec{C}, \alpha)$. Thus if G^f, α^f and β^f satisfy (C2′), then G, α, β satisfy (C2).

Now, let $\vec{P}$ be an oriented path between fixed vertices in G. If the end-vertices of P have the same colour, then the oriented edges of $\vec{P}$ again form a set of edge-disjoint oriented cycles in G^f, and (C2′) implies that $W(\vec{P}, \alpha) = W(\vec{P}, \beta)$. If the end-vertices of P have a different colour, then we can suppose, without loss of generality, that the end-vertices of P are coloured 1 and 2 and that $\vec{P}$ is oriented from the end-vertex coloured 1 towards the end-vertex coloured 2. This means that the union of the oriented edges of $\vec{P}$ and the edge $\overrightarrow{f_2 f_1}$ forms a set of oriented cycles in G^f. Since we have $w(\overrightarrow{f_2 f_1}, \alpha^f) = w(\overrightarrow{f_2 f_1}, \beta^f)$, (C2′) again implies that $W(\vec{P}, \alpha) = W(\vec{P}, \beta)$. Hence if G^f, α^f, β^f satisfy (C2′), then G, α, β satisfy (C3).

Conversely, if there is a cycle C in G^f such that $W(\vec{C}, \alpha^f) \neq W(\vec{C}, \beta^f)$, then this same cycle can be found in G or, if C intersects $\{f_1, f_2, f_3\}$, then there is a path between fixed vertices in G that has different weights under α and β. This shows that if G^f, α^f, β^f do not satisfy (C2′), then one of (C2) or (C3) fails for G, α, β.

Proof of Claim 4.9. To prove the claim we describe an algorithm that either finds a path from α^f to β^f in $\mathcal{C}_3(G^f)$, or finds a cycle C in G^f such that $W(\vec{C}, \alpha^f) \neq W(\vec{C}, \beta^f)$. The algorithm attempts to find a sequence of recolourings that transforms α^f into β^f. It maintains a set $F \subseteq V(G^f)$ such that the subgraph induced by F is connected and for each $v \in F$, the current colouring of v is $\beta^f(v)$. Initially, if F_α and F_β were not empty, we let $F = \{f_1, f_2, f_3\}$. Otherwise, we set $F = \varnothing$. We then try to increase the size of F one vertex at a time.

We show how to extend F if $F \neq V(G^f)$. If $F \neq \varnothing$, then choose a vertex $v \notin F$ such that v is adjacent to a vertex $u \in F$. If $F = \varnothing$, then we choose an arbitrary vertex v, and u does not exist. Suppose the current colouring is α'. If $\alpha'(v) = \beta^f(v)$, we can extend F to include v immediately. Otherwise, let us assume that $\alpha'(v) = 2$ and $\beta^f(v) = 3$. Note that this means that $\alpha'(u) = 1$ (if u exists), since $\alpha'(u) = \beta^f(u)$ and u is adjacent to v.

Now we attempt to find the positive levels of v in α'. This is easily done algorithmically: $L_1^+(v)$ contains those neighbours of v coloured 3; $L_2^+(v)$ contains neighbours of vertices in $L_1^+(v)$ coloured 1, and so on. We stop if either

(L1) we reach a level L_i^+ that is empty, or

(L2) we find a level that contains a vertex $w \in F$.

Note that one of (L1) or (L2) must occur. This is because any vertex not in F belongs to at most one level (if a vertex belongs to two levels it is fixed, and all fixed vertices are in F). Hence we eventually reach either a level that contains a vertex $w \in F$, or an empty level. If F is empty, then, of course, (L1) must occur.

If (L1) occurs, then we can recolour each vertex z in L_j^+, $j = i-1, i-2, \ldots, 0$, with $\alpha'(z)+1 \pmod 3$, starting with the highest level and working down. Thus, ultimately, v is recoloured 3 and we can now add v to F. If there are still vertices not in F, we repeat the procedure.

Suppose (L2) occurs. Then there is a path P from u to w coloured 1-2-3-1-2-3-$\cdots$-$\alpha'(w)$. Moreover, no internal vertex of P is in F. As u and w are in F, and F induces a connected subgraph, we can extend P to a cycle C using a path $Q = w, \ldots, u$ in F. We claim that $W(\vec{C}, \alpha') \neq W(\vec{C}, \beta^f)$, and hence the cycle C is an obstruction that shows that α' and β^f do not belong to the same component of $\mathcal{C}_3(G)$. Because α^f and α' do belong to the same component of $\mathcal{C}_3(G)$, this cycle is also an obstruction showing that α^f and β^f do not belong to the same component of $\mathcal{C}_3(G)$.

To see that $W(\vec{C}, \alpha') \neq W(\vec{C}, \beta^f)$, choose the orientation $\vec{C}$ so that the edge uv is oriented from u to v. The weight of $\vec{C}$ is the sum of the weights of $\vec{P}$ and $\vec{Q}$ (taking $\vec{P}$ and $\vec{Q}$ to have the same orientation as $\vec{C}$). Let $W(\vec{Q}, \alpha') = k$. As vertices in F are coloured alike in α' and β^f, $W(\vec{Q}, \beta^f) = k$. Let p be the number of edges in P. Then $W(\vec{P}, \alpha') = p$, since each edge has weight $+1$. But $W(\vec{Q}, \beta^f) < p$, since $w(\overrightarrow{uv}, \beta^f) = -1$. Thus we find $W(\vec{C}, \beta^f) < k + p = W(\vec{C}, \alpha')$.

All the above was done under the assumption that $\alpha'(v) = 2$ and $\beta^f(v) = 3$. In the cases $\alpha'(v) = 3$, $\beta^f(v) = 1$ and $\alpha'(v) = 1$, $\beta^f(v) = 2$ we do exactly the same, again using the positive levels $L_i^+(v)$. In the other three cases, we follow the same steps, but now using the negative levels $L_i^-(v)$ of v. This completes the proof of the claim. □

This completes the proof of Theorem 4.6.

Note that if α and β are in the same component of $\mathcal{C}_3(G)$ and G has n vertices, the algorithm in the proof of Claim 4.9 will use at most $\frac{1}{2}n(n+1)$ recolouring steps: each time a vertex is added to F, we have recoloured all vertices not in F at most once. This proves Theorem 4.2.

Note also that the procedure which finds the fixed vertices of a given 3-colouring, the construction of G^f from G, and the algorithm in the proof of Claim 4.9 can clearly be performed in polynomial time. This proves Theorem 4.1.

Using Theorem 4.6, it is now possible to give an alternative proof of Theorem 4.1. We describe a modification of the algorithm that proves Theorem 4.6 which, given a graph G together with two 3-colourings α and β as input, decides whether or not α and β belong to the same component of $\mathcal{C}_3(G)$ by simply checking conditions (C1), (C2) and (C3).

As before, we first check whether condition (C1) is satisfied. We proceed by assuming it (else the algorithm terminates), and then transform the instance G, α, β into the instance G^f, α^f, β^f. We have already observed that these operations can be performed in polynomial time.

Having seen that condition (C2′) is equivalent to conditions (C2) and (C3), we now claim that condition (C2′) can be verified in polynomial time. (Note that, a priori— that is, without having proved Theorem 4.1—, this is not immediately obvious, since the graph G^f may contain an exponential number of cycles.) In order to prove this claim, we need to recall some definitions.

Let H be a connected graph with n vertices and m edges. It is well-known that (the edge sets of) the cycles of H form a vector space over the field $\mathbb{F}_2 = \{0, 1\}$, where addition is symmetric difference. This vector space is known as the *cycle space* of H. Given any spanning tree T of H, adding any of the $m - n + 1$ edges $e \in E(H) \setminus E(T)$ to T yields a unique cycle C_e of H. These $m - n + 1$ cycles are called the *fundamental cycles* of T, and they form a basis of the cycle space of H known as a *cycle basis*. In fact, it is easy to prove that for every cycle C,

$$C = \sum_{e \in E(C) \setminus E(T)} C_e,$$

where addition is as in the vector space $(\mathbb{F}_2)^m$. We refer the reader to [17, Section 1.9] for full details.

That we can check if G^f, α^f, β^f satisfies condition (C2′) in polynomial time follows from the following lemma.

Lemma 4.10

Let H be a connected graph with n vertices and m edges. Let α be a 3-colouring of H, T a spanning tree of H, and $\{C_e \mid e \in E(H) \setminus E(T)\}$ the set of fundamental cycles of T. Then for any cycle C in H, $W(\vec{C}, \alpha)$ is determined by the values of $W(\vec{C_e}, \alpha)$, for all $e \in E(H) \setminus E(T)$.

Proof. Let C be any cycle in H, and write $C = \sum_{e \in E(C) \setminus E(T)} C_e$, with addition as in the vector space $(\mathbb{F}_2)^m$. Choose an orientation $\vec{C}$ for C. For each $e \in E(C) \setminus E(T)$, orient the fundamental cycle C_e so that e has the same orientation in $\overrightarrow{C}$ and in $\overrightarrow{C_e}$. We claim that

$$W(\overrightarrow{C}, \alpha) = \sum_{e \in E(C) \setminus E(T)} W(\overrightarrow{C_e}, \alpha), \tag{4.1}$$

where addition is now the normal addition of integers. We prove (4.1) by counting edge-weight contributions to both sides of the equation.

Let $e = uv$ be an edge of C, with orientation $\overrightarrow{uv}$ on $\overrightarrow{C}$. Clearly $w(\overrightarrow{uv}, \alpha)$ is counted exactly once on the left-hand side (LHS) of (4.1). To count the contributions that e makes to the right-hand side (RHS) of (4.1), we distinguish two cases, according to whether or not e is an edge of T. If $e \notin E(T)$, then the definition of C_e and the choice of the orientation $\overrightarrow{C_e}$ immediately give that e contributes exactly the weight $w(\overrightarrow{uv}, \alpha)$ to the RHS. If $e = uv \in E(T)$, we claim that it appears oriented as $\overrightarrow{uv}$ exactly one more time than it appears oriented as $\overleftarrow{uv}$ in the cycle expansion of $\overrightarrow{C}$. Note that uv is a cut-edge of T and, as such, its removal splits T into two subtrees T_u and T_v, with $u \in V(T_u)$ and $v \in V(T_v)$. We also have $V(T_u) \cup V(T_v) = V(H)$. Let $f \in E(C) \setminus E(T)$ with $uv \in E(C_f)$. Then, in fact, we can take $f = xy$ with $x \in V(T_u)$ and $y \in V(T_v)$. If f has the orientation $\overrightarrow{xy}$ in $\overrightarrow{C}$, then it has the same orientation in $\overrightarrow{C_f}$, and hence the edge uv has the orientation $\overleftarrow{uv}$ in C_f. The reverse is the case if f has the orientation $\overleftarrow{xy}$ in $\overrightarrow{C}$. Going along the oriented edges of the cycle $\overrightarrow{C}$, we have the same number of edges $\overrightarrow{xy}$ with $x \in V(T_u)$ and $y \in V(T_v)$, as we have edges between $V(T_u)$ and $V(T_v)$ going in the other direction. But since uv is one of the edges of the first count, we get exactly one more edge $xy \neq uv$ of $\overrightarrow{C}$ with $x \in V(T_u)$ and $y \in V(T_v)$ oriented as $\overleftarrow{xy}$ than oriented the other way round. This means that in the sum on the RHS of (4.1) we have exactly one more contribution of the form $w(\overrightarrow{uv}, \alpha)$ than of the form $w(\overleftarrow{uv}, \alpha)$.

Now suppose that $e = uv$ is not an edge of C. Clearly this edge makes no contribution to the LHS of the equation. Again, to count the contributions of this edge to the RHS of the expression, we distinguish the cases where e is an edge of T and where it is not. If $e = uv \in E(T)$, we can argue as in the preceding paragraph to see that this time we have, in the RHS of (4.1), exactly the same number of contributions of the form $w(\overrightarrow{uv}, \alpha)$ as of the form $w(\overleftarrow{uv}, \alpha)$. Hence the net contribution to the RHS is zero. Lastly, if $e \notin E(T)$ it makes no contribution either, since the fundamental cycle C_e to which it corresponds does not appear in the cycle expansion of C.

This completes the proof of the lemma. □

4.2 Distances between 3-colourings

We have seen that if α and β are 3-colourings of a graph G that are in the same component of $\mathcal{C}_3(G)$, then they are at distance $O(|V(G)|^2)$. In this section we show that this bound on the distance between 3-colourings is of the right order. Before doing so, we prove that in the case that α and β are connected and $F_\alpha \neq \varnothing$ (so $F_\beta \neq \varnothing$ and for all $v \in F_\alpha$, $\alpha(v) = \beta(v)$), the algorithm described in Section 4.1 finds a shortest path from α to β in $\mathcal{C}_3(G)$. Once again, throughout this section, G will denote a fixed 3-colourable connected graph. We also use the notation and terminology introduced in the previous section.

Finding shortest paths between 3-colourings

Theorem 4.11
Let α and β be two 3-colourings of a connected graph G that are in the same component of $\mathcal{C}_3(G)$, and suppose that $F_\alpha \neq \varnothing$. Then the algorithm described in Section 4.1 finds a shortest path between α and β.

Proof. Our algorithm in fact finds a path from α^f to β^f in G^f, but, as we observed earlier, the relevant components of the two colour graphs are isomorphic. For a 3-colouring γ of G^f, let us denote by $\mathcal{C}_\gamma$ the component of $\mathcal{C}_3(G^f)$ containing γ. Note that since we are assuming that α and β are connected, so are α^f and β^f; that is, $\mathcal{C}_{\alpha^f} = \mathcal{C}_{\beta^f}$.

Recall that G^f has exactly three fixed vertices f_1, f_2, f_3 in the colourings α^f and β^f.

Let γ be a 3-colouring in $\mathcal{C}_{\beta^f}$. For any vertex v of G^f, let $\overrightarrow{P}$ be an oriented path from f_1 to v. Then the *height of v in γ* is defined as

$$h(v,\gamma) \;=\; |W(\overrightarrow{P},\gamma) - W(\overrightarrow{P},\beta^f)|.$$

We need to prove that this definition is independent of the choice of P. If there are two oriented paths $\overrightarrow{P_1}$ and $\overrightarrow{P_2}$ from f_1 to v, then, noting that their union is a set of oriented cycles and applying Lemma 2.10, we have

$$W(\overrightarrow{P_1},\gamma) - W(\overrightarrow{P_2},\gamma) = W(\overrightarrow{P_1},\beta^f) - W(\overrightarrow{P_2},\beta^f).$$

Rearranging, we obtain

$$|W(\overrightarrow{P_1},\gamma) - W(\overrightarrow{P_1},\beta^f)| = |W(\overrightarrow{P_2},\gamma) - W(\overrightarrow{P_2},\beta^f)|.$$

Now let γ and δ be adjacent 3-colourings in $\mathcal{C}_{\beta^f}$ and let w be the unique vertex on which they differ. Note that this means that all neighbours of w are coloured the same as one another, and all these neighbours are coloured the same in both γ and δ. Let $\overrightarrow{P}$ be an oriented path from f_1 to some vertex v and let us consider how the height of v changes as γ is recoloured to δ. If w is not on $\overrightarrow{P}$, then clearly $h(v,\gamma) = h(v,\delta)$. We know $w \neq f_1$, as f_1 is fixed. If w is an internal vertex of $\overrightarrow{P}$, then the sum of the weights of the two edges of $\overrightarrow{P}$ incident with w is zero for both γ and δ, so again $h(v,\gamma) = h(v,\delta)$. If $w = v$, then the sign of the weight of the edge of $\overrightarrow{P}$ incident with v changes as we recolour. So in this last case we have $|h(v,\gamma) - h(v,\delta)| = 2$.

Note that finding a path from α^f to β^f is equivalent to finding a sequence of recolourings that reduces the height of every vertex v from $h(v,\alpha^f)$ to zero. In the previous paragraph we saw that each time we recolour, only the height of the vertex being recoloured changes, and it either increases or decreases by 2. So if we can find a sequence of recolourings that always reduces the height of the vertex being recoloured, we will have found a shortest path. We show that this is indeed what the algorithm of Claim 4.9 does.

Recall that the algorithm starts with a set $F = \{f_1, f_2, f_3\}$ and then repeatedly adds vertices v to F, where v has a neighbour $u \in F$. To add v to F, the vertices in either all its positive levels or all its negative levels are recoloured before v itself is recoloured. Assume that we are in the case that to recolour v all positive levels need to be recoloured; the other case is proved in the same way. Let y be a vertex that is about to be recoloured at some stage in this process (this can be v itself, or any of the vertices in the positive levels of v). We must show that its height will be reduced. Let γ and δ be the colourings before and after y is recoloured. Let $\overrightarrow{Q}$ be an oriented path from u to y that contains one vertex from each non-negative level of v. So if there are k edges in $\overrightarrow{Q}$, then $W(\overrightarrow{Q},\gamma) = k$. Thus $W(\overrightarrow{Q},\delta) = k-2$, since the edge of $\overrightarrow{Q}$ incident with y has its weight changed from 1 to -1 when y is recoloured. Let $\overrightarrow{R}$ be an oriented path from f_1 to u containing only vertices in F, and let $\overrightarrow{P}$ be the union of $\overrightarrow{R}$ and $\overrightarrow{Q}$.

Since the colourings β^f, γ, δ agree on F, we have $W(\overrightarrow{R},\beta^f) = W(\overrightarrow{R},\gamma) = W(\overrightarrow{R},\delta)$. We also know that $w(\overrightarrow{uv},\beta^f) = -1$, and since $\overrightarrow{Q}$ has k edges, this means

$$W(\overrightarrow{Q},\beta^f) \leq k-2 = W(\overrightarrow{Q},\delta) < k = W(\overrightarrow{Q},\gamma).$$

From this we can derive

$$\begin{aligned} h(y,\gamma) &= |W(\overrightarrow{P},\gamma) - W(\overrightarrow{P},\beta^f)| = |W(\overrightarrow{Q},\gamma) - W(\overrightarrow{Q},\beta^f)| \\ &= W(\overrightarrow{Q},\gamma) - W(\overrightarrow{Q},\beta^f) = k - W(\overrightarrow{Q},\beta^f) \end{aligned}$$

and similarly,

$$h(y,\delta) = k-2-W(\overrightarrow{Q},\beta^f).$$

Hence every recolouring indeed reduces the height of the vertex being recoloured. This completes the proof. □

Now let us observe that if there are no fixed vertices, the algorithm may find a much longer path. For example, consider two colourings of a path that differ only on an end-vertex v and its neighbour: α = 1-2-3-1-2-3-1-$\cdots$-1-2-3 and β = 2-1-3-1-2-3-1-$\cdots$-1-2-3. The algorithm starts by setting $F = \varnothing$ and then chooses an arbitrary first vertex to start the recolouring. If that first vertex is v, then the algorithm will start by recolouring every vertex on the path. But clearly it is possible to get from α to β via only three recolourings. The reader can check that this shortest number of recolourings would be obtained if the first vertex chosen by the algorithm were any vertex other than v.

We believe that the algorithm from Section 4.1 will also, with an appropriate choice of initial vertex, find a shortest path between two 3-colourings without fixed vertices.

Conjecture 4.12
Let α and β be two 3-colourings of a graph G that are in the same component of $\mathcal{C}_3(G)$, and suppose that $F_\alpha = F_\beta = \varnothing$. For $v \in V(G)$, let $T(v)$ be the number of recolourings required by the algorithm in Section 4.1 when the algorithm starts by adding v to $F = \varnothing$. Then the length of the shortest path between α and β is equal to $\min_{v\in V(G)} T(v)$.

Pairs of 3-colourings at quadratic distance

We construct a class of instances G, α, β where, for each G, α and β are connected and at distance $\Omega(|V(G)|^2)$ in $\mathcal{C}_3(G)$. For $N \in \mathbb{N}$, let us define the graph G_N as the graph consisting of a 3-cycle with an attached path of length N. More precisely, let

- $V(G_N) = \{f_1, f_2, f_3\} \cup \{v_1, v_2, \ldots, v_N\}$, and
- $E(G_N) = \{f_1f_2, f_1f_3, f_2f_3\} \cup \{f_3v_1, v_1v_2, v_2v_3, \ldots, v_{N-1}v_N\}$.

Let α_N be the 3-colouring of G_N given by $\alpha_N(f_i) = i$, for $i = 1, 2, 3$, and where the vertices $v_1, v_2, \ldots, v_N$ are coloured $1, 2, 3, 1, 2, 3, \ldots$. Similarly, let β_N be the 3-colouring of G_N given by $\beta_N(f_i) = i$, for $i = 1, 2, 3$, and where the vertices $v_1, v_2, \ldots, v_N$ are coloured $2, 1, 3, 2, 1, 3, \ldots$.

Theorem 4.13
Let $N \in \mathbb{N}$ and let G_N, α_N, β_N be as described above. Then the 3-colourings α_N and β_N of G_N are connected and at distance $\frac{1}{2}N(N+1) = \Omega(|V(G_N)|^2)$ in $\mathcal{C}_3(G_N)$.

Proof. It is clear that G_N, α_N and β_N satisfy conditions (C1), (C2) and (C3). Therefore, by Theorem 4.6, α_N and β_N are connected in $\mathcal{C}_3(G_N)$.

As in the proof of Theorem 4.11, we consider heights of vertices. For any vertex v of G_N, let $\overrightarrow{P}$ be an oriented path from f_3 to v, noting that $f_3 \in F_{\alpha_N}$. Define the height of v in α_N as $h(v, \alpha_N) = |W(\overrightarrow{P}, \alpha_N) - W(\overrightarrow{P}, \beta_N)|$.

We have seen, in the proof of Theorem 4.11, that finding a shortest path from α_N to β_N is equivalent to finding a sequence of recolourings that reduces the height of every vertex in α_N to zero, and that, with each recolouring, we reduce the height of the recoloured vertex by 2, while the height of all other vertices remains the same. This enables us to calculate the distance between α_N and β_N: we just need to calculate the heights of all vertices in α_N.

First observe that $h(f_i, \alpha_N) = 0$, for $i = 1, 2, 3$. For $i = 1, \ldots, N$, let $\overrightarrow{P_i}$ be the oriented path from f_3 to v_i, and observe that $W(\overrightarrow{P_i}, \alpha_N) = i$, while $W(\overrightarrow{P_i}, \beta_N) = -i$. This means that $h(v_i, \alpha_N) = |W(\overrightarrow{P_i}, \alpha_N) - W(\overrightarrow{P_i}, \beta_N)| = 2i$. We thus find that the distance between α_N and β_N is equal to

$$\frac{1}{2}\sum_{i=1}^{N} h(v_i, \alpha_N) = \sum_{i=1}^{N} i = \frac{1}{2}N(N+1).$$

Since G_N has $N + 3$ vertices, we obtain that this distance is indeed $\Omega(|V(G_N)|^2)$. □

5

Paths between k-colourings

We saw in Chapter 4 that the decision problem 3-Colour Path is solvable in polynomial time. In this chapter we determine the complexity of the problem k-Colour Path for values of $k \geq 4$, proving the following.

Theorem 5.1
*For every fixed $k \geq 4$, the decision problem k-*Colour Path *is* PSPACE-*complete. Moreover, it remains* PSPACE-*complete for the following restricted instances:*

(i) bipartite graphs and any fixed $k \geq 4$;

(ii) planar graphs and any fixed $4 \leq k \leq 6$; and

(iii) bipartite planar graphs and $k = 4$.

The reader will also recall that we proved in Chapter 4 that if α and β are two 3-colourings of a graph G connected in $\mathcal{C}_3(G)$, then the distance between them is $O(|V(G)|^2)$. Again, we will see that things are remarkably different for the case of general k-colourings: we will prove that if $k \geq 4$, the distance between two k-colourings of a graph can be superpolynomial in the size of the graph. More precisely, we will prove the following.

Theorem 5.2
For every fixed $k \geq 4$, there exists a class of graphs $\{G_{N,k} \mid N \in \mathbb{N}\}$ with the following properties. The graphs $G_{N,k}$ have size $O(N^2)$, and for each of them there exist two k-colourings in the same component of $\mathcal{C}_k(G_{N,k})$ which are at distance $\Omega(2^N)$. Moreover,

(i) the graphs $G_{N,k}$ may be taken to be bipartite;

(ii) *for every* $4 \leq k \leq 6$, *the graphs* $G_{N,k}$ *may be taken to be planar (in such a case the graphs actually have size* $O(N^4)$*); and*

(iii) *for* $k = 4$, *the graphs* $G_{N,k}$ *may be taken to be planar and bipartite (in such a case the graphs actually have size* $O(N^4)$*).*

The proofs of Theorems 5.1 and 5.2 both involve the construction of particular k-COLOUR PATH instances. In both cases it will be convenient, in order to simplify the proofs, to first define some preliminary constructions. We do this in Section 5.1. We then prove Theorem 5.1 in Section 5.2 and Theorem 5.2 in Section 5.3. Theorems 4.1, 4.2, 5.1 and 5.2 together suggest that the computational complexity of k-COLOUR PATH and the possible distance between k-colourings are intimately linked. We investigate the extent of this correspondence in Section 5.4.

5.1 Preliminaries

List-colouring instances

In Sections 5.2 and 5.3 we will construct particular k-COLOUR PATH instances G, α, β: first for the PSPACE-hardness proof, and then for the superpolynomial distance proof. In both cases, it is easier to first define *list-colouring instances*: for such instances we give every vertex v a *colour list* $L(v) \subseteq \{1, 2, 3, 4\}$. A proper list-colouring is a proper vertex colouring with the additional requirement that every vertex colour needs to be chosen from the colour list of the vertex. In the same way as that in which we define the colour graph $\mathcal{C}_k(G)$ of G with nodes corresponding to proper k-colourings, we define the *list-colour graph* $\mathcal{C}(G, L)$ of G with nodes corresponding to proper list-colourings, where L represents the colour lists. The problem LIST-COLOUR PATH is now defined as follows.

LIST-COLOUR PATH

Instance: Graph G, colour lists $L(v) \subseteq \{1, 2, 3, 4\}$ for all $v \in V(G)$, two list-colourings of G, α and β.

Question: Is there a path between α and β in $\mathcal{C}(G, L)$?

Whenever colour lists are given for the vertices of the graph, 'proper list-colouring' should be read when we say 'colouring'. In figures we will write colour lists as 123 instead of $\{1, 2, 3\}$, for example.

A list-colouring instance can then be turned into a normal 4-colouring instance, for example, by adding a complete graph K_4 on vertex set $\{u_1, u_2, u_3, u_4\}$. Since any 4-colouring of K_4 is frozen, we may without loss of generality assume that $\kappa(u_i) = i$ in all colourings κ in the component of the colour graph we consider. Now adding edges vu_i if and only if $i \notin L(v)$ turns the graph into a 4-colouring instance, where in all 4-colourings κ we consider, $\kappa(v) \in L(v)$. The next lemma proves formally that this can be done for various k without increasing the size of the graph too much, even when we require that planarity and bipartiteness should be maintained.

Lemma 5.3
For any $k \geq 4$, a List-Colour Path *instance G, L, α, β with lists $L(v) \subseteq \{1,2,3,4\}$ can be transformed into a k-*Colour Path *instance G', α', β' such that the distance between α and β in $\mathcal{C}(G, L)$ (possibly infinite) is the same as the distance between α' and β' in $\mathcal{C}_k(G')$. Moreover,*

(i) if G is bipartite, this can be done so that G' is also bipartite, for all $k \geq 4$;

(ii) if G is planar, this can be done so that G' is also planar, when $4 \leq k \leq 6$; and

(iii) if G is planar and bipartite, this can be done so that G' is also planar and bipartite, when $k = 4$.

In all cases, the transformation can be accomplished in a way such that $|V(G')| \leq |V(G)|f(k)$ and $|E(G')| \leq |E(G)| + |V(G)|g(k)$, for some functions $f(k)$ and $g(k)$.

Proof. For our transformations we need: for every $k \geq 4$, a bipartite graph with a frozen k-colouring; for every $4 \leq k \leq 6$, a planar graph with a frozen k-colouring; and a planar bipartite graph with a frozen 4-colouring. For the first case we can take the graphs L_k described in Definition 2.2: we observed in Chapter 2 that these graphs satisfy our requirements. For the second we can use K_4 and the planar graphs with frozen colourings shown in Figure 2.2. For the third case we just need to observe that the graph L_4, which is in fact isomorphic the 3-dimensional cube, is planar.

The transformation from a List-Colour Path instance G, L, α, β to a k-Colour Path instance G', α', β' is now as follows. Let F be a graph with a frozen k-colouring κ. For every vertex $v \in V(G)$ and colour $c \in \{1, \ldots, k\} \setminus L(v)$, we add a copy of F to G, labelled $F_{v,c}$. We also add an edge between v and a vertex u of $F_{v,c}$ with $\kappa(u) = c$. This yields G'. The colourings α' and β' are obtained by extending α and β using the colouring κ for every $F_{v,c}$.

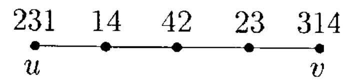

Figure 5.1 A $(1,3)$-forbidding path from u to v.

It is easy to see that every k-colouring obtainable from α' or β' by recolouring induces the same frozen colouring on every copy of F. Also, because of the way the edges between v and vertices of $F_{v,c}$ are added, all these k-colourings of G' correspond to list-colourings of G, and vice-versa. This proves that the distance between α and β in $\mathcal{C}(G, L)$ is exactly the same as the distance between α' and β' in $\mathcal{C}_k(G')$.

When G and F are bipartite, the construction of G' starts with a number of bipartite components, and edges are added only between different components. So in this case G' is also bipartite. It can also be seen that G' is planar when G and F are planar: start with a planar embedding of G and for each copy $F_{v,c}$ of F, consider a planar embedding that has a vertex with colour c on its outer face. These embeddings of $F_{v,c}$ can be inserted into a face of G that is incident with v. Now adding an edge between v and a vertex of $F_{v,c}$ with colour c can be done without violating planarity.

Since for all $k \geq 4$ we can choose F to be bipartite, for $4 \leq k \leq 6$ we can choose F to be planar, and for $k = 4$ we can choose F to be both planar and bipartite, we are done. $\square$

Adding (a,b)-forbidding paths

The next notion that will be used in the following sections is that of an (a,b)-forbidding path. For $a, b \in \{1,2,3,4\}$, an *(a,b)-forbidding path* from u to v is a (u,v)-path with colour lists L, with $L(u)$, $L(v) \neq \{1,2,3,4\}$, such that in any colouring, it is not possible that u has colour a and v simultaneously has colour b. Any other combination of colours for u and v (chosen from their colour lists) is possible. In addition, any recolouring of u and v is possible—perhaps after first recolouring a few internal vertices of the path—as long as it does not yield the forbidden colour combination. (Note that if $a \neq b$, an (a,b)-forbidding path from u to v is not the same as an (a,b)-forbidding path from v to u.) Figure 5.1 shows an example of a $(1,3)$-forbidding path from u to v. We formalise these concepts in the following definition.

Definition 5.4

A colouring κ of a (u, v)-path is a (c, d)-*colouring* if $\kappa(u) = c$ and $\kappa(v) = d$. A (u, v)-path P with colour lists L, where $a \in L(u)$ and $b \in L(v)$ is an (a, b)-*forbidding path* if the following two conditions are satisfied.

- A (c, d)-colouring exists if and only if $c \in L(u)$, $d \in L(v)$ and $(c, d) \neq (a, b)$. Such a pair (c, d) is called *admissible* for P.
- If both (c, d) and (c', d) are admissible, then for any (c, d)-colouring, a sequence of recolourings exists that ends with a (c', d)-colouring, without ever recolouring v, and only recolouring u in the last step. A similar statement holds for admissible pairs (c, d) and (c, d').

In the constructions in the following sections we will often say 'add an (a, b)-forbidding path between u and v'. This means that we add an (a, b)-forbidding (u', v')-path P with $L(u') = L(u)$ and $L(v) = L(v')$ to the graph, and then identify u with u' and v with v'. Then for the colourings and recolourings of u and v in the resulting graph, the above properties will hold. This means that in our proofs we do not have to consider colourings and recolourings of the internal vertices of the path in detail; we can simply assume that any recolouring of u and v is possible, as long as this does not respectively give them colours a and b.

The next lemma shows that we do not even have to describe such an (a, b)-forbidding path in detail; as long as $L(u), L(v) \neq \{1, 2, 3, 4\}$, such a path always exists.

Lemma 5.5

For any $L_u \subset \{1, 2, 3, 4\}$, $L_v \subset \{1, 2, 3, 4\}$, $a \in L_u$ and $b \in L_v$, there exists an (a, b)-forbidding (u, v)-path P with $L(u) = L_u$, $L(v) = L_v$ and all other colour lists $L(w) \subseteq \{1, 2, 3, 4\}$. Moreover, we can insist P has even length at most six.

Proof. Let $c \in \{1, 2, 3, 4\} \setminus L(u)$ and $d \in \{1, 2, 3, 4\} \setminus L(v)$. If $c \neq d$ then we let P be a path of length four with the following colour lists along the path: L_u, $\{a, c\}$, $\{c, d\}$, $\{d, b\}$, L_v. We prove it is an (a, b)-forbidding path: if in a given colouring u has colour a, then the second vertex has colour c, the third colour d, the fourth colour b, so v cannot have colour b. When v has colour b the reasoning is analogous. It can also be seen that for every admissible (x, y), an (x, y)-colouring exists. This colouring is unique if $x = a$ or $y = b$. If not, then it can be verified that all (x, y)-colourings can be obtained from each other by recolouring internal vertices of P only. Adjacent (x, y)- and (x, y')-colourings are found as follows: if $x = a$, then both colourings are unique, and they are adjacent. If $x \neq a$ then we find adjacent colourings by, if necessary, colouring the vertex next to u with a, the middle vertex with c, and the vertex adjacent to v with colour d, in both colourings. Adjacent (x, y)- and (x', y)-colourings are found similarly. We conclude that P with these colour lists is indeed an (a, b)-forbidding path with the required properties.

If $c = d$, then we let P be a path of length six with the following colour lists along the path: L_u, $\{a, c\}$, $\{c, e\}$, $\{e, f\}$, $\{f, c\}$, $\{c, b\}$, L_v, for some $e \in \{1, 2, 3, 4\} \setminus \{a, c\}$ and $f \in \{1, 2, 3, 4\} \setminus \{b, c\}$ with $e \neq f$. As before, it can be verified that this is an (a, b)-forbidding path. □

5.2 PSPACE-completeness of k-Colour Path

In this section we prove Theorem 5.1. We recall that PSPACE is defined as the class of decision problems that are decidable by a deterministic Turing machine that uses at most a polynomial (in the size of the input) amount of work space. Similarly, NPSPACE is the class of decision problems decidable by a *non*-deterministic Turing machine using a polynomially-bounded amount of space. The PSPACE-hardness of k-Colour Path will be shown using a reduction from Sliding Tokens, one of several decision problems defined and proved to be PSPACE-complete by Hearn and Demaine in [30]. We first reduce Sliding Tokens to List-Colour Path and then apply Lemma 5.3 to prove the existence of equivalent k-Colour Path instances. We first establish that k-Colour Path is indeed in PSPACE.

Claim 5.6
*The decision problem k-*Colour Path *is in the complexity class* PSPACE.

Proof. We actually prove that k-Colour Path is in NPSPACE, and then appeal to Savitch's Theorem, which asserts that PSPACE = NPSPACE (see [53, p. 150] or [57] for details). Given an instance G, α, β of k-Colour Path together with a sequence of recolourings transforming α into β (the *certificate*), we can easily check the validity of the certificate using a polynomial amount of space. This means that k-Colour Path is in NPSPACE. □

A PSPACE-complete problem: SLIDING TOKENS

The main result of Hearn and Demaine [30] is the presentation of a new non-deterministic model of computation based on reversing edge directions in weighted directed graphs with minimum in-flow constraints on vertices. This model, called *non-deterministic constraint logic*, or NCL, is shown to have the same computational power as a space-bounded Turing machine, and several decision problems surrounding it are proved to be PSPACE-complete. These decision problems are then used to prove the PSPACE-completeness of certain sliding-block puzzles such as Rush Hour and Sokoban. The last section of [30] gives an equivalent formulation of NCL in terms of sliding tokens along graph edges, and it is this formulation that we use for our reductions. We proceed to describe it, first giving some definitions. (The interested reader will find a more detailed description of NCL and its different formulations in the Appendix.)

A *token configuration* of a graph G is a set of vertices on which tokens are placed, in such a way that no two tokens are adjacent. (Thus a token configuration can be thought of as an independent set of vertices of G.) A *move* between two token configurations is the displacement of a token from one vertex to an adjacent vertex. Note that a move must result in a valid token configuration.

Amongst others, the following decision problem, which we call SLIDING TOKENS, is proved in [30] to be PSPACE-complete.

SLIDING TOKENS
Instance : Graph G, two token configurations of G, T_A and T_B.
Question : Is there a sequence of moves transforming T_A into T_B?

The reduction used to prove PSPACE-completeness of SLIDING TOKENS in [30] actually shows that the problem remains PSPACE-complete for very restricted graphs and token configurations. Our reduction to LIST-COLOUR PATH is actually from a slightly wider class of restricted instances for which SLIDING TOKENS remains PSPACE-complete, but we do not give a reduction from the general problem. We proceed to describe the instances G, T_A, T_B of SLIDING TOKENS that we will use for our reduction.

The graphs G are made up of *token triangles* (copies of K_3) and *token edges* (this involves a slight abuse of terminology: when we say token edge, we actually mean a copy of K_2). Token triangles and token edges are all mutually disjoint, and joined together by edges called *link edges*, in such a way that every vertex of G is part of exactly one token triangle or token edge. Moreover, every vertex in a token triangle ends up with degree 3, and G has a planar embedding where every token triangle bounds a face. The graphs G have

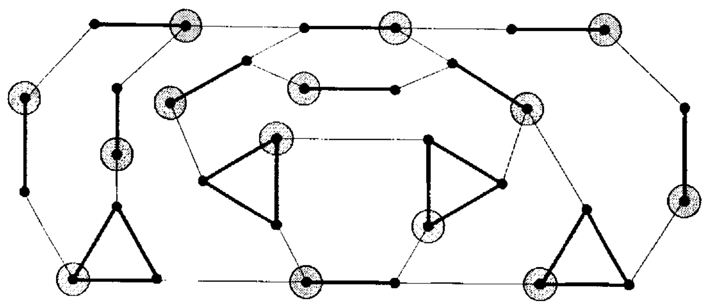

Figure 5.2 An example of a restricted instance graph together with a standard token configuration.

maximum degree 3 and minimum degree 2.

The token configurations T_A and T_B are such that every token triangle and every token edge contain exactly one token on one of their vertices. In any sequence of moves from T_A or T_B, a token will never leave its triangle or its edge: the first time a token were to do so, we would cease to have a valid token configuration. Hence tokens will never slide along a link edge. (We remark that it is this limitation on possible token displacements that allows for a reasonably straightforward reduction.) Token configurations where every token triangle and every token edge contain exactly one token are called *standard token configurations* of G—thus T_A and T_B are standard token configurations. A simple example of a restricted instance graph G with a standard token configuration is shown in Figure 5.2, where token triangles and token edges are shown in bold. We insist: for these restricted instances, SLIDING TOKENS is PSPACE-complete. For further details, we refer the reader to the Appendix and [30].

The construction of equivalent LIST-COLOUR PATH instances

Given a restricted instance G, T_A, T_B of SLIDING TOKENS as described above, we construct an instance G', L, α, β of LIST-COLOUR PATH such that standard token configurations of G correspond to list-colourings of G', and sliding a token in G corresponds to a sequence of vertex recolourings in G'.

We first label the vertices of G: the token triangles are labelled $1, \ldots, n_t$, and the vertices of triangle i are labelled t_{i1}, t_{i2} and t_{i3}. The token edges are labelled $1, \ldots, n_e$, and the vertices of token edge i are labelled e_{i1} and e_{i2}.

The construction of G' is as follows: for every token triangle i we introduce a vertex t_i, with colour list $L(t_i) = \{1, 2, 3\}$. For every token edge i we introduce a vertex e_i in G', with colour list $L(e_i) = \{1, 2\}$. Whenever a link edge of G joins a vertex t_{ia} with a vertex e_{jb}, we add an (a, b)-forbidding path (of even length at most 6) between t_i and e_j in G'. We do the same for pairs t_{ia} and t_{jb}, and pairs e_{ia} and e_{jb}. Note that this is a polynomial time transformation.

Standard token configurations of G now correspond to colourings of G' as follows: a token configuration where the token of token edge i is on e_{ij} ($j = 1, 2$) corresponds to colourings of G' where e_i has colour j. Analogously, if the token of token triangle i is on t_{ij} ($j = 1, 2, 3$), this corresponds to colourings where t_i has colour j. Since tokens are not adjacent, it is possible to choose colours for the internal vertices of the (a, b)-forbidding paths so as to obtain a proper colouring of G'. Two colourings α and β corresponding respectively to T_A and T_B are constructed this way. Note that to a given standard token configuration of G there can correspond multiple colourings of G' because of the freedom in choice of colours for the internal vertices of the (a, b)-forbidding paths.

Claim 5.7

The graph G' as constructed above is planar and bipartite.

Proof. Let us consider a planar embedding of G where all token triangles bound a face. A planar embedding of G' can be obtained from that of G by contracting all token triangles and token edges, and subdividing the remaining (link) edges. All (a, b)-forbidding paths in G' have even length, so G' is bipartite. □

Claim 5.8

Let G, T_A, T_B be a restricted instance of Sliding Tokens, *and let G', L, α, β be a corresponding instance of* List-Colour Path *as constructed above. Then G, T_A, T_B is a YES-instance if and only if G', L, α, β is a YES-instance.*

Proof. Recall that a token configuration in which the token of token edge i (token triangle i) is on e_{ij} (on t_{ij}) corresponds to multiple colourings of G' where e_i (t_i) has colour j. Because of this multiplicity of colourings, we define *colour classes* of colourings: if two colourings κ and λ of G' have $\kappa(t_i) = \lambda(t_i)$ and $\kappa(e_i) = \lambda(e_i)$ for every i, then κ and λ are said to be in the same colour class.

Hence the correspondence between standard token configurations of G and colourings of G' defines a mapping between standard token configurations and colour classes. This mapping is in fact a bijection: (a, b)-forbidding paths restrict their end vertices from having colours a and b respectively, but they pose no other restriction on the possible colours of their end vertices. So t_{ia} and e_{jb} cannot both be occupied by a token in a token configuration if and only if no colouring κ has $\kappa(t_i) = a$ and $\kappa(e_j) = b$. (Similar statements hold for pairs t_i and t_j, and pairs e_i and e_j.)

Now we claim that if there exists a sequence of moves that transforms T_A into T_B, then there exists a sequence of recolourings that transforms α into β. We mentioned earlier that any token configuration obtainable from T_A is a standard token configuration. Hence every token move corresponds to recolouring a vertex t_i or a vertex e_i. Note that before recolouring t_i (or e_i), it may be necessary to first recolour some internal vertices of (a, b)-forbidding paths incident with t_i (or e_i), but by the definition of (a, b)-forbidding paths, we know this is always possible. It can also be seen that when we finally arrive at the colour class that contains β in this way, the internal vertices of all (a, b)-forbidding paths can be recoloured so that exactly the colouring β is obtained.

Similarly, for every sequence of recolourings from α to β we can construct a sequence of token moves from T_A to T_B: whenever a vertex t_i (e_i) is recoloured from colour a to colour b, we move the corresponding token from t_{ia} to t_{ib} (from e_{ia} to e_{ib}). This completes the proof. □

Claim 5.8 shows that the instance G', L, α, β of LIST-COLOUR PATH we constructed above is equivalent to the given instance of SLIDING TOKENS. In addition, G' is planar and bipartite, by Claim 5.7. Now we can use Lemma 5.3 to construct equivalent k-COLOUR PATH instances from G', L, α, β. All of these transformations can be accomplished in polynomial time, and we saw in Claim 5.6 that k-COLOUR PATH is in PSPACE. This completes the proof of Theorem 5.1.

Let us now observe that the values of k in parts (ii) and (iii) of Theorem 5.1 are tight. We saw in Chapter 2 that a planar graph is always k-mixing for $k \geq 7$. We saw this as a consequence of Theorem 2.7 and the fact that the degeneracy of a planar graph is at most 5. Hence any instance G, α, β of k-COLOUR PATH, where G is planar and α, β are k-colourings with $k \geq 7$ is trivially a YES-instance. Similarly, if we note that the degeneracy of a bipartite planar graph is at most 3 and appeal to Theorem 2.7, we have that k-COLOUR PATH is trivial for bipartite planar graphs and $k \geq 5$. The fact that for a bipartite graph G we have $\deg(G) \leq 3$ can be seen as follows. Considering an embedding of G in the plane, let us write $F(G)$ for the set of faces of G and, for ϕ a face of G, let us denote the number of edges bounding ϕ by $d(\phi)$. Then, noting that

$\sum_{v\in V(G)} d(v) = \sum_{\phi\in F(G)} d(\phi) = 2m$, we may write Euler's formula $n - m + f = 2$ as

$$\sum_{v\in V(G)} (d(v) - 4) + \sum_{\phi\in F(G)} (d(\phi) - 4) = -8.$$

Since G is bipartite, $d(\phi) \geq 4$ for all $\phi \in F(G)$. This means that for some vertices of G we must have $d(v) < 4$; that is, $\deg(G) \leq 3$.

Together with Theorems 4.1 and 5.1, the observations above allow us to completely determine the complexity of k-Colour Path for planar and bipartite planar graphs.

Theorem 5.9
*Restricted to planar graphs, the decision problem k-*Colour Path *is* PSPACE-*complete for $4 \leq k \leq 6$, and in* P *for all other values of k.*

Theorem 5.10
*Restricted to bipartite planar graphs, the decision problem k-*Colour Path *is* PSPACE-*complete for $k = 4$, and in* P *for all other values of k.*

5.3 Distances between k-colourings

In this section we construct classes of k-Colour Path instances such that the distance between the two colourings is superpolynomial in the size of the graph. As in the proof of Theorem 5.1, we will do this by first constructing classes of List-Colour Path instances and then applying Lemma 5.3.

For every integer $N \geq 1$, we construct a graph G_N with colour lists L. (To avoid cluttering the notation, we will denote the colour lists of each G_N by L; which graph these lists belong to will be clear from the context.) The graphs G_N will have size $O(N^2)$ and the list-colour graphs $\mathcal{C}(G_N, L)$ will have a component of diameter $\Omega(2^N)$. Later in the section we will show how to obtain bipartite and planar instance classes with the same property. In the case of planar instances, the graphs G_N will have size $O(N^4)$ and the list-colour graphs $\mathcal{C}(G_N, L)$ will have a component of diameter $\Omega(2^N)$.

The number N can be seen as the number of 'bits' that is used in the graph: the graph will have N vertices whose colour can be thought of as a binary variable. For every combination of binary values there will exist a corresponding colouring of G_N. These combinations can be mapped to values $0, \ldots, 2^N - 1$ in such a way that one can only increase or decrease this value by one when recolouring G_N.

For a given N, the graph G_N is constructed as follows. Start with N triangles, each consisting of vertices v_i, v_i' and v_i^* with $L(v_i) = \{1,2\}$, $L(v_i') = \{1,2,3\}$ and $L(v_i^*) = \{3,4\}$, for $i = 1,\dots,N$. In a colouring κ where $\kappa(v_i^*) = 3$, triangle i is said to be *locked*, otherwise it is *unlocked*. Now between every pair v_i^* and v_j^* with $i \neq j$ we add a $(4,4)$-forbidding path. Hence we have the following.

Claim 5.11

At most one triangle can be unlocked in any colouring.

For every i, we add (a,b)-forbidding paths from v_i^* to every v_j with $j < i$: we add a $(4,1)$-forbidding path from v_i^* to v_{i-1}, and $(4,2)$-forbidding paths from v_i^* to v_j with $j \leq i-2$. This ensures the following.

Claim 5.12

Triangle i can only be unlocked in a colouring κ when $\kappa(v_{i-1}) = 2$ and $\kappa(v_j) = 1$ for all $j \leq i-2$.

This yields the graph G_N.

Claim 5.13

The graphs G_N have $O(N^2)$ vertices and $O(N^2)$ edges.

Proof. The graph G_N consists of N triangles, $N(N-1)/2$ $(4,4)$-forbidding paths, and $N(N-1)/2$ paths that are either $(1,4)$-forbidding or $(2,4)$-forbidding.

Because by Lemma 5.5 we can assume that all (a,b)-forbidding paths have length at most 6, we get $|V(G_N)| \leq 3N + 5N(N-1)$ and $|E(G_N)| \leq 3N + 6N(N-1)$. □

To show that there exists a pair of colourings of G_N such that exponentially many steps (exponential in N) are needed to go from one to the other, we need only consider the colours of the vertices v_i. These can be seen as the N bits with value 1 or 2. We call a colouring κ of G_N a $(c_1, c_2, \dots, c_N)$*-colouring* if $\kappa(v_i) = c_i$ for all i. All $(c_1, c_2, \dots, c_N)$-colourings together form the *colour class* $(c_1, c_2, \dots, c_N)$.

Claim 5.14

Let $c_i \in \{1,2\}$ for $1 \leq i \leq N$. Then the colour class $(c_1, \dots, c_N)$ is non-empty.

Proof. Consider a colouring κ where $\kappa(v_i) = c_i$, $\kappa(v_i') = 3 - c_i$ and $\kappa(v_i^*) = 3$ for all i. Since all triangles are locked, this colouring does not violate any of the constraints imposed by the forbidding paths, and so can be extended to a full colouring of G_N. □

Lemma 5.15

Let $(x_1, x_2, \ldots, x_N)$ and $(y_1, y_2, \ldots, y_N)$ be distinct tuples with all $x_i, y_i \in \{1, 2\}$.

- *If the tuples differ only on position i, with $x_{i-1} = 2$ and $x_j = 1$ for all $j < i - 1$, then from any colouring in class $(x_1, x_2, \ldots, x_N)$ we can reach some colouring in class $(y_1, y_2, \ldots, y_N)$ via a sequence of recolourings, without ever leaving colour class $(x_1, x_2, \ldots, x_N)$ in the intermediate colourings.*
- *Otherwise, there is no colouring in class $(x_1, x_2, \ldots, x_N)$ that is adjacent to a colouring in class $(y_1, y_2, \ldots, y_N)$.*

Proof. Let $(x_1, x_2, \ldots, x_N)$ and $(y_1, y_2, \ldots, y_N)$ be tuples as described above, and suppose that the conditions described in the first bullet point hold. We show that any colouring κ in class $(x_1, x_2, \ldots, x_N)$ can be recoloured to a colouring in class $(y_1, y_2, \ldots, y_N)$. Note that by the definition of (a, b)-forbidding paths, we may ignore all recolourings of the internal vertices of these paths, since we know that any necessary recolouring of these vertices is always possible.

We first show how to recolour κ to an $(x_1, x_2, \ldots, x_N)$-colouring in which only triangle i is unlocked. If all triangles are locked in κ, we can immediately recolour v_i^* to 4—this does not violate any of the constraints imposed by the forbidding paths. Otherwise, there is exactly one triangle which is unlocked. Let this triangle be triangle j, where $j \neq i$. We now lock this triangle. If we cannot immediately recolour v_j^* to 3, this must be because $\kappa(v_j') = 3$. We change this colour to $\kappa(v_j') = 3 - \kappa(v_j)$, and then triangle j can be locked. Next, triangle i can be unlocked: no other triangles are unlocked, so the $(4, 4)$-forbidding paths pose no restriction. Since $\kappa(v_{i-1}) = 2$ and $\kappa(v_j) = 1$ for all $j < i - 1$, the $(4, 1)$ and $(4, 2)$-forbidding paths starting at v_i^* pose no restriction either. At this point, we can set $\kappa(v_i') = 3$, and then set $\kappa(v_i) = y_i$ to obtain a colouring in class $(y_1, \ldots, y_N)$. This proves the first statement.

Now let α be an $(x_1, x_2, \ldots, x_N)$-colouring, let β be a $(y_1, y_2, \ldots, y_N)$-colouring, and suppose that that α and β are adjacent. This means they differ only on one vertex, and because the tuples are distinct, α and β must therefore differ precisely on a vertex v_i, for some i. This means triangle i is unlocked in both colourings. Because of the $(4, 1)$- and $(4, 2)$-forbidding paths starting at v_i^*, $\alpha(v_{i-1}) = 2$ and $\alpha(v_j) = 1$ for all $j < i - 1$. This proves the second statement. □

It follows from Lemma 5.15 that every colour class is adjacent to at most two other colour classes (we use the concept of adjacency of colour classes with the obvious meaning).

$$\begin{array}{ccccccccc}
1111 & \to & 2111 & \to & 2211 & \to & 1211 & \to \\
1221 & \to & 2221 & \to & 2121 & \to & 1121 & \to \\
1122 & \to & 2122 & \to & 2222 & \to & 1222 & \to \\
1212 & \to & 2212 & \to & 2112 & \to & 1112 &
\end{array}$$

Figure 5.3 Colour classes visited in a shortest path between a $(1,1,1,1)$-colouring and a $(1,1,1,2)$-colouring of G_4.

Firstly, the colour of v_1 can always be changed. In addition, there is at most one v_i such that v_{i-1} has colour 2 and v_j has colour 1 for all $j < i-1$; this is the only other vertex of $v_1, \dots, v_N$ whose colour can be changed without first changing that of one of the others. Figure 5.3 shows all colour classes of G_4 and the order in which these need to be visited in order to go from a $(1,1,1,1)$-colouring to a $(1,1,1,2)$-colouring of G_4 all 16 different classes need to be visited. We now prove this formally for every N.

Theorem 5.16
Every graph G_N has two colourings α and β in the same component of $C(G_N, L)$ which are at distance at least $2^N - 1$.

Proof. For the colouring α we choose a colouring in class $(1,\dots,1)$. Colouring β will be a colouring in class $(1,\dots,1,2)$. Such colourings exist by Claim 5.14. We first prove by induction that such colourings are connected, using the following induction hypothesis.

Induction hypothesis
There is a path in $C(G_N, L)$ from any colouring α' in class $(1,\dots,1,x_0,x_1,\dots,x_{N-n})$ to some colouring β' in class $(1,\dots,1,3-x_0,x_1,\dots,x_{N-n})$.

The colourings differ on vertex v_n: we have $\alpha'(v_n) = x_0$ and $\beta'(v_n) = 3 - x_0$, while for all $i \neq n$, we have $\alpha'(v_i) = \beta'(v_i)$. If $n = 1$, the statement follows directly from Lemma 5.15. If $n > 1$, then from α' we recolour to a $(1,\dots,1,2,x_0,x_1,\dots,x_{N-n})$-colouring (which differs from the initial class only in the $(n-1)$-th position), using the induction hypothesis. Then we recolour to a $(1,\dots,1,2,3-x_0,x_1,\dots,x_{N-n})$-colouring, using Lemma 5.15. Finally, using the induction hypothesis again, we can recolour to a $(1,\dots,1,1,3-x_0,x_1,\dots,x_{N-n})$-colouring, which proves the statement.

Now we show that to go from a $(1,\dots,1)$-colouring to a $(1,\dots,1,2)$-colouring, at least

$2^N - 2$ other colour classes need to be visited, using the following induction hypothesis.

Induction hypothesis

To go from a $(1, \ldots, 1, 1, x_1, \ldots, x_{N-n})$-colouring to a $(1, \ldots, 1, 2, y_1, \ldots, y_{N-n})$-colouring, at least $2^n - 2$ other colour classes need to be visited.

Let us denote the vertex recoloured from 1 to 2 (appearing before the vertex coloured x_1 and y_1) by v_n. If $n = 1$, the statement is obvious. If $n > 1$, then consider a shortest path between two colourings in these classes, if it exists. At some point in the sequence of recolourings, the colour of v_n is changed for the first time; before this we must have a $(1, \ldots, 1, 2, 1, z_1, \ldots, z_{N-n})$-colouring, by Lemma 5.15 (in this colouring, v_{n-1} has colour 2). By the induction hypothesis, at least $2^{n-1} - 2$ colour classes have been visited before this colour class was reached. Now changing the colour of v_n to 2 yields a $(1, \ldots, 1, 2, 2, z_1, \ldots, z_{N-n})$-colouring. Using the induction hypothesis again, at least $2^{n-1} - 2$ colour classes need to be visited before class $(1, \ldots, 1, 2, y_1, \ldots, y_{N-n})$ is reached. This means that in total, at least $2^n - 4 + 2$ intermediate colour classes have been visited in the recolouring procedure. This completes the proof. □

Claim 5.13 and Theorem 5.16 show that G_N with its colour lists L is a list-colouring instance such that $C(G_N, L)$ has a component of diameter superpolynomial in the size of G_N. Unfortunately, the graphs G_N are neither bipartite nor planar. We now use the graphs G_N to construct bipartite and planar list-colouring instances with the same property.

Making the graphs planar and bipartite

We start with a copy of G_N with lists L and obtain a bipartite graph G_N^B with lists L as follows. For every i, we remove the edge $v_i v_i^*$: this does not influence the possible colourings and recolourings of v_i and v_i^* since the colour-lists of these vertices are disjoint. All forbidding paths can be chosen of even length by Lemma 5.5, and since all vertices v_i and v_i^* are now in the same part of the bipartition, the resulting graph is bipartite. As before, we can find two colourings α and β of G_N^B that are at distance at least $2^N - 1$. The size of these graphs is not significantly different to that of the graphs G_N.

Claim 5.17

The graphs G_N^B have $O(N^2)$ vertices and $O(N^2)$ edges.

Next, we use the graphs G_N^B to construct bipartite planar LIST-COLOUR PATH instances G_N^P. Observe that G_N^B can be drawn in the plane so that only edges of forbidding paths cross; that is, so that edges that were formerly part of the triangles never cross. Using such a drawing of G_N^B (without too many crossings, see Claim 5.20 below), we replace every (a,b)-forbidding path P on which there are r crossings by a long path consisting of $r+2$ new paths $Q_0,\ldots,Q_{r+1}$, drawn along the same curve as the old path. We do this in a way such that the paths Q_i contain exactly one crossing, for $1 \le i \le r$, and Q_0 and Q_{r+1} contain no crossings. For $0 \le i \le r$, the paths Q_i and Q_{i+1} share a vertex with colour list $\{1,2\}$. For $1 \le i \le r$, the path Q_i will be a $(1,2)$-forbidding path; Q_0 will be an $(a,2)$-forbidding path and Q_{r+1} will be a $(1,b)$-forbidding path. Together, these even length paths form an (a,b)-forbidding path of even length, as can be seen from repeated application of the following observation.

Claim 5.18

Let Q be an (a,b)-forbidding path from u to v, and let Q' be a (c,d)-forbidding path from v to w such that $V(Q)\cap V(Q')=\{v\}$, where $L(v)=\{b,c\}$. Together, Q and Q' form an (a,d)-forbidding path from u to w.

After this is done for every (a,b)-forbidding path that contains crossings, we end up with a drawing where the only crossings occur between (1,2)-forbidding paths, where both end vertices of both paths have colour list $\{1,2\}$. All such pairs are now replaced with a crossing component such as that depicted in Figure 5.4: the figure shows an example of the crossing component for an (n,s)-path and a (w,e)-path that are both $(1,2)$-forbidding paths.

After replacing all such crossings we obtain a planar graph. Note that bipartiteness is maintained: previously all end vertices of (a,b)-forbidding paths were in the same part of the bipartition, and this is also true for the end vertices of the crossing component. In addition, all cycles in the crossing component are even. We call the resulting graph G_N^P. The following lemma shows that, with regard to the possible colourings and recolourings of the end vertices n,s,w,e, this crossing component behaves in exactly the same way as the two old forbidding paths.

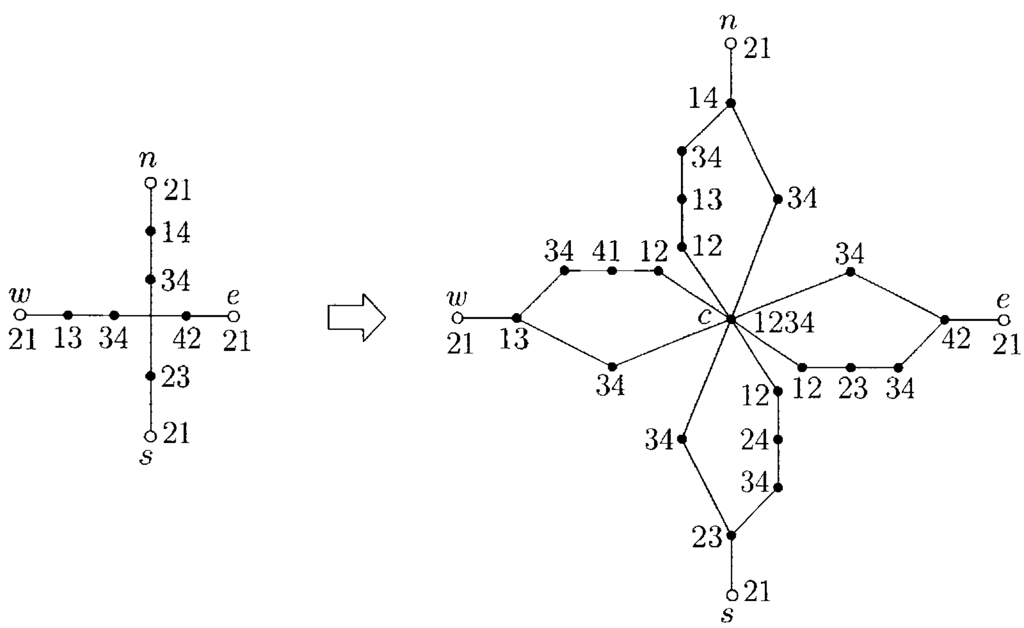

Figure 5.4 A crossing component corresponding to two $(1, 2)$-forbidding paths.

Lemma 5.19

The crossing component of Figure 5.4 has the following properties.

- *For $c_n, c_s, c_w, c_e \in \{1, 2\}$, a colouring κ with $\kappa(n) = c_n$, $\kappa(s) = c_s$, $\kappa(w) = c_w$ and $\kappa(e) = c_e$ exists if and only if*

$$\neg(c_n = 1 \wedge c_s = 2) \wedge \neg(c_w = 1 \wedge c_e = 2).$$

- *For any colouring κ with $\kappa(s) = 1$, there exists a sequence of recolourings that ends by changing $\kappa(n)$, without ever changing $\kappa(s)$, $\kappa(w)$ or $\kappa(e)$. Similar statements hold for recolouring s when $\kappa(n) = 2$, recolouring w when $\kappa(e) = 1$ and recolouring e when $\kappa(w) = 2$.*

Proof. The vertex c is the central vertex of the crossing component. The graph consists of four branches around c, called the north, south, west and east branches. Before we begin the proof of the above statements, let us make the following observation, which spares us a lot of case analysis: swapping colours 1 and 2 in the lists of the crossing component corresponds to mirroring the drawing in the bottom-left to top-right diagonal, and swapping colours 3 and 4 corresponds to mirroring in the top-left to bottom-right diagonal. So whenever we prove a statement for the north branch, the same statement holds for the east (west) branch when we swap the colours 1 and 2 (3 and 4) in the statement. Swapping both 1 with 2 and 3 with 4 yields a correct statement for the south branch.

If c has colour 3, then n must have colour 2 (arguing along the right path of the north branch). If c has colour 2, then n again has colour 2 (consider the left path in the north branch). In general we find, for a colouring κ:

- if $\kappa(c) \in \{2,3\}$, then $\kappa(n) = 2$;
- if $\kappa(c) \in \{1,4\}$, then $\kappa(s) = 1$;
- if $\kappa(c) \in \{2,4\}$, then $\kappa(w) = 2$;
- if $\kappa(c) \in \{1,3\}$, then $\kappa(e) = 1$.

Since either $c \in \{2,3\}$ or $c \in \{1,4\}$, it follows that $\kappa(n) = 1$ and $\kappa(s) = 2$ cannot occur simultaneously; similarly for w and e. It can also be seen that whenever c is not coloured with 2 or 3, there exist colourings of the north branch where n has colour 1, and colourings where n has colour 2. Similar statements hold for the other three branches. All this proves that for every combination of colours c_n, c_s, c_w, c_e for the four vertices, a corresponding colouring κ exists, except when $c_n = 1$ and $c_s = 2$, or when $c_w = 1$ and $c_e = 2$. This proves the first statement about possible colourings. Now we consider possible recolourings of the crossing component.

We prove that we can always recolour n, as long as s has colour 1, without ever recolouring w or e. Whenever c has colour 1 or 4, it is easy to see that we can recolour the north branch and change the colour of n without any recolouring of c or of the other branches.

Now suppose $\kappa(c) = 3$. This means $\kappa(n) = 2$ and $\kappa(e) = 1$. In this case we first change the colours of all vertices adjacent to c to 2 or 4, without changing $\kappa(n)$, $\kappa(s)$, $\kappa(w)$ or $\kappa(e)$.

- It is obvious this can be done in the west branch.
- For the east branch we use the fact that $\kappa(e) = 1$.
- For the south branch we use the fact that $\kappa(s) = 1$.
- For the north branch we use the fact that $\kappa(n) = 2$.

At this point we can recolour c to 1. Now it can be checked that the vertices in the north branch can be recoloured so that n gets colour 1.

Similarly, when $\kappa(c) = 2$ all of c's neighbours can be recoloured to 1 or 3 without recolouring n, s, w or e. Then c can be recoloured to 4, which in turn allows n to receive colour 1, after a few steps.

This shows that we can always recolour n whenever $\kappa(s) = 1$. For the other three branches, similar statements follow from the above mentioned symmetries. □

Claim 5.18 and Lemma 5.19 show that after replacing forbidding paths with multiple forbidding paths, and replacing crossings with crossing components, the new structures act like the old forbidding paths with regard to possible colourings and recolourings of v_i, v_i' and v_i^* (though perhaps 'a few' more recolourings of internal vertices are needed). So the statements from Lemma 5.15 and Theorem 5.16 can be proved for these graphs. Adapting the two colourings of G_N to colourings of G_N^P is straightforward. It remains only to consider the size of the graphs G_N^P.

Claim 5.20

The graphs G_N^P have $O(N^4)$ vertices and $O(N^4)$ edges.

Proof. We start with a drawing of G_N in which only (a, b)-forbidding paths cross. It is easy to see that a drawing can be found such that every pair of forbidding paths crosses at most once. An informal proof runs as follows. First embed the $3N$ vertices in the N triangles along a circular closed curve in the plane, where the three vertices of each triangle are placed consecutively along the circle. The edges of the triangles are then placed along the circle, or outside it. The forbidding paths are now added as straight lines across the interior of the circle. If more than two paths go through the same point, then this can be corrected by small perturbations. This yields the desired drawing.

The graph G_N has $O(N^2)$ forbidding paths, so the drawing we have just described has at most $O(N^4)$ crossings. For every crossing we introduce a number of new vertices that is bounded by some constant (closely related to the number of vertices in a crossing component), so the number of vertices, which was $O(N^2)$, increases to at most $O(N^4)$. So the number of vertices of G_N^P is $O(N^4)$. Since G_N^P is planar, its average degree is less than six, so the number of edges is $O(N^4)$ as well. □

We have constructed bipartite LIST-COLOUR PATH instances with size $O(N^2)$ (Claim 5.17) and bipartite planar LIST-COLOUR PATH instances with size $O(N^4)$ (Claim 5.20). The pairs of colourings for each of these instances are at distance at least $2^N - 1$, just as for the original LIST-COLOUR PATH instances, proved in Theorem 5.16. Lemma 5.3 shows that these can be transformed into k-COLOUR PATH instances without a significant size increase. This completes the proof of Theorem 5.2.

5.4 Tractability of k-COLOUR PATH and distances between k-colourings

In this section we examine the relationship between the tractability of k-COLOUR PATH and the possible distances between k-colourings.

Let us first examine the relationship between Theorems 5.1 and 5.2. In terms of the well-known NP$\neq$PSPACE conjecture, Theorem 5.1 means the following. Loosely speaking, having established that k-COLOUR PATH is PSPACE-complete, asserting that NP$\neq$ PSPACE is *equivalent* to saying that for every possible YES-certificate for k-COLOUR PATH, there exist instances for which the certificate cannot be verified in polynomial time. Theorem 5.2 of course shows this only for a particular certificate—the certificate for a YES-instance consisting of a list of colourings constituting a path from the first colouring to the second colouring—but this is in some sense the the most natural certificate. It is for this reason that we consider the construction of these instances to be of independent interest. In addition, they have a clear bearing on the limitations of sampling colourings via Glauber dynamics.

Theorems 4.1 and 4.2 from Chapter 4 tell us that 3-COLOUR PATH is polynomial time solvable and that for any YES-instance G, α, β of this problem, the distance between α and β in $\mathcal{C}_3(G)$ is at most quadratic in the size of G. On the other hand, Theorems 5.1 and 5.2 establish a connection between instance classes for which k-COLOUR PATH is PSPACE-complete and possible superpolynomial distances in the k-colour graphs of these instances. How strong is this connection between PSPACE-completeness and superpolynomial distances in the colour graph? For completeness let us point out that artificial graph classes can be constructed for which k-COLOUR PATH is easy, but which still contain instances with colourings at superpolynomial distance. This can be done, for example, using the graphs from Section 5.3.

We remark that the reason why we cannot make the values of k in parts (ii) and (iii) of Theorem 5.2 larger by a straightforward extension of our methods rests fundamentally on the fact that for a planar graph G, $\deg(G) \leq 5$, and that for a bipartite planar graph G, $\deg(G) \leq 3$. These considerations, together with Theorems 5.9 and 5.10, beg the following question: is it true that for a planar graph G and $k \geq 7$, or G a bipartite planar graph and $k \geq 5$, $\mathcal{C}_k(G)$ always has polynomial diameter? More generally, given that an instance of k-Colour Path is always a YES-instance for $k \geq \deg(G) + 2$, is it true that for any graph G and $k \geq \deg(G) + 2$, $\mathcal{C}_k(G)$ has polynomial diameter? Noting that the proof of Theorem 2.7 only gives an exponential upper bound, we conjecture that this is indeed the case, and that in fact a quadratic bound is the correct answer. (Let us remark that in [7], the paper containing the results of this chapter, it is in fact a cubic upper bound which is conjectured.)

Conjecture 5.21
For a graph G with n vertices and $k \geq \deg(G) + 2$, the diameter of $\mathcal{C}_k(G)$ is $O(n^2)$.

For values of $k \geq 2\deg(G) + 1$, we are in fact able to prove this bound.

Theorem 5.22
For a graph G with n vertices and $k \geq 2\deg(G) + 1$, the diameter of $\mathcal{C}_k(G)$ is $O(n^2)$.

Proof. We can iteratively delete vertices of degree at most $\deg(G)$ until no vertices are left. Using such an elimination ordering, we label the vertices $v_1, v_2, \ldots, v_n$ so that every vertex has at most $\deg(G)$ neighbors with a lower index. (The label v_n corresponds to the first deleted vertex.) Using this vertex ordering, we first prove the following statement by induction on n.

Induction hypothesis
Let α and β be distinct k-colourings of G, and let i be the lowest index such that $\alpha(v_i) \neq \beta(v_i)$. There exists a recolouring sequence that starts with α and ends with recolouring v_i to $\beta(v_i)$, where every v_j with $j < i$ is never recoloured, and every v_j with $j \geq i$ is recoloured at most once.

The statement is trivial for $n = 1$. If $i = n$, then v_n can be recoloured to $\beta(v_n)$ because β is a proper colouring that coincides with α on all other vertices. Now suppose $i < n$, and let $G' = G - \{v_n\}$. Let α' be the k-colouring of G' induced by α. By induction we can assume there exists a recolouring sequence starting with α' that ends with recolouring v_i to $\beta(v_i)$, in which vertices v_j with $j < i$ are not recoloured, and vertices v_j with $j \geq i$ are recoloured at most once. So for every vertex we can identify an *old* colour and a *new* colour in this recolouring sequence (which may in fact be the same). Because there are at least $2\deg(G) + 1$ available colours, and v_n has at most $\deg(G)$ neighbours, a colour c can be chosen for v_n that is not equal to the old colour or new colour of any of its neighbours. First recolour v_n to c if necessary, and then recolour the rest of the graph according to the recolouring sequence for G'. By the choice of colour c, all intermediate colourings are proper, so this is the desired recolouring sequence for G.

Now we can keep repeating the above procedure, each time for a new vertex v_i with a higher index, since the colours of the vertices with a lower index are not changed. So every vertex v_i is considered only once this way, and for every v_i only $n-i$ recolourings of other vertices are needed before it can be recoloured to $\beta(v_i)$. This will yield β after $O(n^2)$ recolouring steps. □

Let us now observe that if in Conjecture 5.21 we replace the degeneracy with the maximum degree, we again obtain a quadratic bound on the diameter of the k-colour graph. This answers a question of Bill Jackson.

Proposition 5.23
For a graph G with n vertices and $k \geq \Delta(G)+2$, the diameter of $C_k(G)$ is $O(n^2)$.

Proof. Let α and β be distinct k-colourings of G. We claim that it is possible to recolour α to β using at most Δn recolouring steps. Let $v_1, v_2, \ldots, v_n$ be an arbitrary ordering of the vertices of G, and consider the following recolouring procedure that transforms α into β. For $i = 1, 2, \ldots, n$, we attempt to recolour v_i to $\beta(v_i)$. If for some i this is not possible, this must be because v_i has a neighbour w that is currently coloured $\beta(v_i)$. But because w has degree at most $\Delta \leq k-2$, there is a colour $c \neq \beta(v_i)$ that does not appear on any of the neighbours of w. Hence we can first recolour w to c, and repeat the same procedure for any other neighbour of v_i coloured $\beta(v_i)$. This allows us to then recolour v_i to $\beta(v_i)$ and continue. Because any v_i has at most Δ such neighbours, and once vertex v_i has colour $\beta(v_i)$ it will not be necessary to recolour it again, we reach β after at most Δn recolourings. Noting that $\Delta \leq n-1$ yields the result. □

If we now observe that for a regular graph G, $\Delta(G) = \deg(G)$, Proposition 5.23 allows us to deduce that Conjecture 5.21 is true for regular graphs.

6

Miscellaneous results about recolouring

In this chapter we prove some miscellaneous results obtained during the development of this thesis. In Section 6.1 we explore the problem of finding a sequence of recolourings between two k-colourings of a graph when we are allowed to use some extra colours. Section 6.2 covers some results about the complexity of finding alternative colourings of graphs. Specifically, we investigate several versions of the following decision problem: given a graph G together with a k-colouring of G, how easily can we decide whether there exists a k-colouring of G with certain specific properties?

6.1 Recolouring using extra colours

Suppose we are given a graph G and two k-colourings of G, α and β. We have seen that these colourings may or may not be connected in $\mathcal{C}_k(G)$, and that deciding if they are is in general a PSPACE-complete problem. If we are very keen to recolour one to the other (as may be the case in a frequency reassignment context), with perhaps the use of some extra colours, how many extra colours are enough to guarantee that such a recolouring is possible? It is obvious that this can always be done for a sufficiently large number of extra colours, but it should also be obvious that we might want to minimise the number of extra colours used. The problem can be put another way: what (reasonably small) value of q will guarantee that all k-colourings of a graph G are in the same connected component of $\mathcal{C}_q(G)$? The theorem below provides an answer to this question, originally put to us by Steve Noble.

Theorem 6.1
Let α and β be two k-colourings of a graph G, and let G have chromatic number χ. Then for any $q \geq k + (\chi - 1)$, there is a path between α and β in $C_q(G)$.

Proof. We show that we can recolour α to β with the use of $\chi - 1$ new colours. Consider a χ-colouring γ of G: this gives a partition of the vertex set of G into independent sets $\Gamma_1, \Gamma_2, \ldots, \Gamma_\chi$. We recolour α to β using γ. First we recolour, from α, all vertices in Γ_i with colour $k + i$, for $1 \leq i \leq \chi - 1$. It is clear that no recolouring in this sequence violates the constraint that we maintain a proper-colouring of G. Vertices that are not recoloured in this way are precisely those in the set Γ_χ, but because Γ_χ is independent, we can recolour all vertices in this set to their colours in β. It is easy to see that we can now recolour all vertices $v \in V \setminus \Gamma_\chi$ to $\beta(v)$ without introducing any edges with end-vertices coloured alike. This completes the proof. □

Note that this proof requires knowledge of a χ-colouring of G, which in general will not be readily available since determining the chromatic number of a graph is NP-hard. The best bound we have on the number of sufficient extra colours supported by a *constructive* proof—that is, one that will allow us to actually recolour α to β without knowledge of a χ-colouring—is of $k - 1$ new colours. The idea is similar to that of the proof of Theorem 6.1. From α, we recolour all vertices coloured i to $k + i$, for $1 \leq i \leq k - 1$, and then recolour to β, recolouring vertices coloured k in β last.

We now show that Theorem 6.1 is best possible, in the sense that no lower number of extra colours will always be enough to guarantee a path between any two given k-colourings. That is, we show that $\chi - 1$ extra colours are sometimes necessary.

Theorem 6.2
For every $k \geq \chi \geq 2$, there exists a χ-chromatic graph G that has two k-colourings which are not connected in $C_q(G)$, where $q = k + (\chi - 2)$.

Proof. We let G be the categorical (or tensor) product of K_χ and K_k, which we denote by $B_{\chi,k}$. This is the graph with vertex set $\{(i,j) \mid 1 \leq i \leq \chi, 1 \leq j \leq k\}$ and edge set $\{(i,j)(i',j') \mid i \neq i' \text{ and } j \neq j'\}$. (An example of such a graph—the graph $B_{3,4}$—is depicted in Figure 6.1.) We will think of the is as indexing rows and the js as indexing columns of $B_{\chi,k}$. For the two k-colourings of $B_{\chi,k}$ we take α and β given by $\alpha((i,j)) = i$ and $\beta((i,j)) = j$. Note that α is in fact a χ-colouring, but because $k \geq \chi$ we can actually regard it as a k-colouring. We prove the graphs $B_{\chi,k}$ are χ-chromatic by showing that no $(\chi - 1)$-colouring of $B_{\chi,k}$ exists. Let us assume the contrary. Observe that given *any*

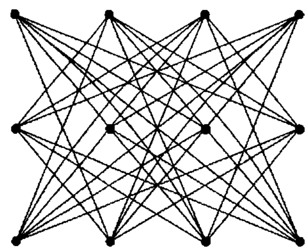

Figure 6.1 The graph $B_{3,4}$.

colouring of $B_{\chi,k}$, if in some row there are two vertices with the same colour, this colour cannot appear in any other row (and similarly for columns). Thus if we are considering a $(\chi - 1)$-colouring of $B_{\chi,k}$, we cannot have all rows each containing two vertices with the same colour. This means there is at least one row with all its k vertices coloured differently, but since $k > \chi - 1$, we have a contradiction.

We now prove that it is not possible to recolour α to β using $\chi - 2$ new colours. Suppose it is possible, and consider a sequence of recolourings that accomplishes the transformation. Note that α is a colouring where all vertices in any given row of $B_{\chi,k}$ have the same colour, and β is a colouring where no two vertices in a given row have the same colour. Hence there must come a point in the sequence of recolourings where for the first time we see a row that has all its vertices coloured differently—row i^*, say. Consider this colouring in the sequence: how many different colours do we see on $B_{\chi,k}$? Because any row other than row i^* has at least two vertices with the same colour, we see at least $\chi - 1$ different colours on rows other than row i^*. Because none of these colours can appear on row i^*, which has its k vertices coloured differently, in total we see at least $k + \chi - 1$ different colours. This contradiction completes the proof. □

We note that the results of Theorems 6.1 and 6.2 have been obtained independently in [35] and [47]. In fact, the graphs of Theorem 6.2 that illustrate the tightness of the result are the same in [35] and (with a minor modification) in [47]. The result in [35] analogous to Theorem 6.1 is in fact a refinement of our result: the author considers k-colourings as possibly using different sets of colours, and proves, for α a k-colouring using colour set A and β a k-colouring using colour set B, that the recolouring can be achieved using $\max\{0, |A \cap B| - 1\}$ extra colours.

6.2 The complexity of finding alternative colourings

The results in this section are motivated by the following question of Peter Winkler: what is the complexity of deciding whether the k-colour graph of a k-colourable graph G contains an isolated node? We answer this question below, in Theorem 6.3, first giving a formal definition of the problem.

FROZEN k-COLOURING
Instance: A connected graph G together with a k-colouring α of G.
Question: Does G have a frozen k-colouring?

It is obvious that the decision problem FROZEN 2-COLOURING is trivial: the 2-colouring given with a connected bipartite graph is frozen. We now prove that for any $k \geq 3$, the problem is NP-complete, initially giving a reduction from 3-COLOURABILITY (defined formally in Section 3.2) to the $k = 3$ case.

Theorem 6.3
For every fixed $k \geq 3$, the decision problem FROZEN k-COLOURING *is* NP-*complete.*

Proof. That FROZEN k-COLOURING is in NP is clear. We first prove that FROZEN 3-COLOURING is NP-complete by giving a polynomial time reduction from 3-COLOURABILITY, and then show that FROZEN k-COLOURING is reducible to FROZEN $(k+1)$-COLOURING.

Given an instance G of 3-COLOURABILITY, we construct an instance G' of FROZEN 3-COLOURING such that G is 3-colourable if and only if G' has a frozen 3-colouring. We obtain G' from G as follows. We replace every edge uv of G with two internally disjoint paths between u and v, one of length 2 and another of length 4, effectively obtaining a 6-cycle between every two vertices that were previously joined by an edge. More formally, for every edge $e = uv$ of G, we delete e and add vertices w_e, x_e, y_e, z_e and edges $uw_e, w_ev, ux_e, x_ey_e, y_ez_e, z_ev$ to obtain G'. Note that G' is bipartite so we can trivially find a 3-colouring of G' to form part of the instance of FROZEN 3-COLOURING. Now suppose G is 3-colourable, and consider a 3-colouring τ of G. An observation: given any 6-cycle C with two vertices at distance two precoloured with two different colours from $\{1, 2, 3\}$, we can extend this precolouring to obtain a frozen 3-colouring of C. It is now clear how to obtain a frozen 3-colouring of G': we just use this observation on every 6-cycle of G' that contains vertices w_e, x_e, y_e, z_e in G', for some specified edge e of G. On the other hand, suppose G' has a frozen 3-colouring τ_f. Restricting τ_f to vertices originally in G yields a proper 3-colouring of G: otherwise, for some vertices u, v of G' originally forming an edge e of G we have $\tau_f(u) = \tau_f(v)$, and this contradicts τ_f being frozen, for we could then recolour w_e.

To show that FROZEN k-COLOURING is reducible to FROZEN $(k+1)$-COLOURING it suffices to take an instance G, α of FROZEN k-COLOURING and form the graph G' by adding a new vertex v adjacent to all vertices of G. We can easily obtain a $(k+1)$-colouring α' of G' by setting $\alpha'(v) = k+1$ and $\alpha'(x) = \alpha(x)$ for all $x \neq v$. Clearly G has a frozen k-colouring if and only if G' has a frozen $(k+1)$-colouring. □

Given that 3-COLOURABILITY remains NP-complete for planar graphs of maximum degree 4, we readily conclude from the above proof that FROZEN 3-COLOURING remains NP-complete for planar graphs of maximum degree 8. We observe that by arguments similar to those of Lemma 5.3 (ii), we can deduce that FROZEN k-COLOURING remains NP-complete for planar graphs and $4 \leq k \leq 6$. For $k \geq 7$, it should be clear that any planar instance of FROZEN k-COLOURING is a NO-instance.

Similarly, the reduction to FROZEN 3-COLOURING yields a bipartite graph, and arguments similar to those of Lemma 5.3 (i) can then be used to see that FROZEN k-COLOURING actually remains NP-complete for bipartite graphs and all values of k.

We observed in Section 2.1 that finding a frozen k-colouring of a particular $(k-1)$-regular graph G is equivalent to verifying that G is a cover of the complete graph K_k. (Remember that we define a graph G as a cover of a graph H if there exists a surjection $\varphi : V(G) \to V(H)$ such that for every vertex v of G, φ maps the neighbours of v in G bijectively to the neighbours of $\varphi(v)$ in H, and thus deciding if G is a cover of K_k is equivalent to deciding if G has a frozen k-colouring.) In [40] it is proved that deciding if a given graph G is a cover of K_k, for any fixed $k \geq 4$, is NP-complete. Hence other than for $k = 3$, Theorem 6.3 is not new.

We can also regard the decision problem FROZEN k-COLOURING as related to the problem of determining whether a given k-colourable graph is uniquely k-colourable. This asks, given a graph G together with a k-colouring α, whether G admits a k-colouring that induces colour classes different to those induced by α, and is known to be NP-complete, [16]. Note that if a graph is uniquely k-colourable, its k-colour graph will consist of $k!$ isolated nodes. We now study two other problems related to deciding unique colourability.

ALTERNATIVE k-COLOURING

Instance: A connected graph G together with a k-colouring α of G and two vertices u, v of G with $\alpha(u) = \alpha(v)$.

Question: Does there exist a k-colouring β of G with $\beta(u) \neq \beta(v)$?

ALTERNATIVE k-COLOURING II

Instance: A connected graph G together with a k-colouring α of G and two vertices u, v of G with $\alpha(u) \neq \alpha(v)$.

Question: Does there exist a k-colouring β of G with $\beta(u) = \beta(v)$?

Again we find the same dichotomy for the computational complexity of these problems: trivial for $k = 2$ and NP-complete for any $k \geq 3$. For both, we give an initial reduction to the $k = 3$ case from the problem 3-PRECOLOURING EXTENSION, proved NP-complete in [41], even when restricted to planar graphs.

3-PRECOLOURING EXTENSION

Instance: A connected bipartite graph G with some vertices properly precoloured with colours from $\{1, 2, 3\}$.

Question: Does the precolouring of G extend to a 3-colouring of G?

Theorem 6.4

For every fixed $k \geq 3$, the decision problems ALTERNATIVE k-COLOURING *and* ALTERNATIVE k-COLOURING II *are* NP-*complete.*

Proof. Both problems are clearly in NP. For each, we first give a reduction from 3-PRECOLOURING EXTENSION to the $k = 3$ case. Before doing so, we show how we may first simplify a general instance of 3-PRECOLOURING EXTENSION so that we can assume that only 3 vertices of the graph are precoloured. (This trick is from [34, Lemma 2.2].) Let G be an instance graph of 3-PRECOLOURING EXTENSION and let X, Y be the bipartition of G. Note that if we identify two vertices $x, x' \in X$ that are precoloured the same we end up with an equivalent bipartite instance, in the sense that the new instance is a YES-instance if and only if the original one is. Hence we can assume that G is precoloured in such a way that each colour occurs at most once in X and at most once in Y. We now add two new disjoint sets of vertices to G, $X' = \{x'_1, x'_2, x'_3\}$ and $Y' = \{y'_1, y'_2, y'_3\}$, so that $(X \cup X'), (Y \cup Y')$ is the bipartition of the new graph G' formed by introducing the following edges, where $x \in X, y \in Y$ and $i, j \in \{1, 2, 3\}$:

- x'_i is adjacent to y'_j if and only if $i \neq j$;

- x'_i is adjacent to $y \in Y$ if and only if y is precoloured with a colour distinct from i;
- y'_j is adjacent to $x \in X$ if and only if y is precoloured with a colour distinct from j.

If we now consider G' as an instance of 3-PRECOLOURING EXTENSION, where only vertices x'_1, x'_2, x'_3 are respectively coloured with colours 1,2,3, we obtain an instance equivalent to the original instance.

We now transform our simplified instance G', x'_1, x'_2, x'_3 of 3-PRECOLOURING EXTENSION into an instance G^*, α, u, v of ALTERNATIVE 3-COLOURING. To obtain the graph G^*, we simply add to G new vertices a, b, c together with edges ab, bc, ca, x'_1c, x'_2c, x'_3a, x'_3b, putting $u = x'_1$ and $v = x'_2$. To obtain a 3-colouring α of G^* with $\alpha(u) = \alpha(v)$, we set $\alpha(x) = 1$ for all $x \in X \cup X' \setminus \{x'_3\}$; $\alpha(x'_3) = 3$; $\alpha(y) = 2$ for all $y \in Y \cup Y'$; and $\alpha(a) = 1$, $\alpha(b) = 2$, $\alpha(c) = 3$. It is straightforward to check that the precolouring of G' extends to the whole of G' if and only if there is a 3-colouring of G^* where u and v receive different colours.

For ALTERNATIVE k-COLOURING II the reduction is even simpler: we transform G', x'_1, x'_2, x'_3 with its precolouring into G^*, α, u, v by adding a single new vertex a and edges x'_1a, $x'_1x'_2$, x'_2a to G', and putting $u = a$ and $v = x'_3$. The colouring α is obtained by setting $\alpha(x'_1) = 1$, $\alpha(x'_2) = 2$, $\alpha(x) = 1$ for all $x \in X \cup X' \setminus \{x'_1, x'_2\}$; $\alpha(y) = 3$ for all $y \in Y \cup Y'$, and $\alpha(a) = 3$. Note $\alpha(u) = 3 \neq 1 = \alpha(x'_3)$ and that the precolouring of G' extends to the whole of G' if and only if there is a 3-colouring of G^* where u and v receive the same colour.

For both ALTERNATIVE k-COLOURING and ALTERNATIVE k-COLOURING II, a k-colouring instance is easily reduced to a $k+1$-colouring instance by adding a new vertex adjacent to all other vertices and extending the colouring accordingly. This completes the proof. □

We note that Theorem 6.4 has been obtained independently by Rackham [54]. In that paper, however, both the approach to the problem and the reduction that proves NP-hardness are different. The reduction is from 3-COLOURABILITY, and the problems ALTERNATIVE k-COLOURING and ALTERNATIVE k-COLOURING II are studied in the context of extending precolourings of a graph. In particular, Section 6 of [54] considers the following problem: given a graph G, a k-colouring of G, and two vertices of G properly precoloured with colours from $\{1, 2, \ldots, k\}$, does the precolouring extend to a proper k-colouring of G? It is shown that this problem can be solved in polynomial time when $k \geq \Delta(G)$, but that it is NP-complete for $k \leq \Delta(G) - o(k)$.

Chlebík and Chlebíková [14] show that the precolouring extension problem with any number of precoloured vertices is solvable in polynomial time when $k \geq \Delta(G)$. This shows that the precolouring extension problem is in P for graphs of maximum degree 3: given a graph G, an integer $k \geq 2$ and a precolouring of G using at most k colours, we can decide whether the precolouring extends to a k-colouring of G as follows. If $k \geq \Delta(G) = 3$, we just use the aforementioned algorithm described in [14]. Otherwise $k = 2$ and the problem is reduced to deciding if the precolouring extends to a 2-colouring, which is easily solvable in polynomial time.

In contrast, it is also proved in [14] that 3-PRECOLOURING EXTENSION remains NP-complete for planar bipartite graphs of maximum degree 4.

7

Conclusion

We close in this chapter with a discussion of the results presented in this thesis. We first attempt to put our work into a wider context, examining results related to our own in Section 7.1. Section 7.2 summarises our results and outlines some open problems and possibilities for further research.

7.1 Related work

As was mentioned in Chapter 1, the study of the colour graph is not new. For instance, the question of its connectedness has been addressed by researchers interested in rapidly mixing Markov chains for sampling colourings of a graph. In addition, during the development of this thesis we have come across a series of lines of research that can be thought of as related to the study of the colour graph. Some of them bear close similarity to our own, or even impinge on them directly—indeed we have seen that some of our results have been independently obtained by other researchers. Other lines can be considered as generalisations of the problems we have addressed. We proceed to give an overview of all of these, beginning with results that, to some extent or other, match our own.

Recolouring graph colourings

In [35]—an unpublished graduate thesis—we find results which closely resemble those of Chapter 5. In particular, the author of [35] proves that the problem of deciding whether there exists a sequence of recolourings between two given colourings of a graph, using no extra colours, is PSPACE-complete. This is, in essence, the result of Theorem 5.1, but is significantly weaker in various respects. Firstly, for the problem as studied in

[35], the number of colours is part of the input. Secondly, the result is not proved for any restricted graph classes such as planar or bipartite graphs—in fact the construction used is highly non-planar. The reduction is also different; from the problem of deciding whether a space-bounded deterministic Turing machine will halt in an accepting state, and via several other decision problems involving word replacement on strings. Because deciding whether a space-bounded deterministic Turing machine will halt in an accepting state is known to possibly take a superpolynomial number of steps, and all the steps in the reductions involve problems with 'states', the proof of PSPACE-completeness in fact also yields a proof of the existence of superpolynomial paths between colourings. The intricacy of the reductions used, however, indicates that actually constructing such instances would be far from straightforward.

We mentioned in Chapter 6 that Theorems 6.1 and 6.2 have also been independently obtained in [35], and that in fact the former is refined. The author of [35] also studies recolouring problems in an online setting, where vertices continually leave or join the graph whose colourings are under consideration.

We saw that Theorems 6.1 and 6.2 were also obtained (independently of [35] and of this thesis) in [47] in the context of the so-called colour switching problem, described in Chapter 1. A rather surprising result on the algorithmic complexity of a variant of colour switching is to be found in [13]. Here it is shown that if, in requiring a transformation from a k-colouring to a k'-colouring of a given graph (with $k' < k$), we only care about the partition induced by the k'-colouring (and not on the actual colours used), then a shortest possible sequence of recolourings can be found in polynomial time. This is achieved by a reduction of the problem to the weighted matching problem on bipartite graphs, a well-known polynomial time solvable problem.

Generalisations of the colour graph

Instead of recolouring a single vertex, we could consider a different transformation between colourings: for example, that provided by a Kempe change. Given a graph G, a k-colouring α of G and colours $c_1, c_2 \in \{1, 2, \ldots, k\}$, let $G(c_1, c_2)$ be the subgraph of G induced by vertices coloured c_1 or c_2. Switching colours c_1 and c_2 on any connected component of $G(c_1, c_2)$ yields a new k-colouring of G. This operation is known as a *Kempe change*, and two colourings are said to be *Kempe-equivalent* if one can be obtained from the other by a sequence of Kempe changes. Analogously to the way in which we define the k-colour graph of a graph G, we could consider the graph with vertex set the k-colourings of G and edges between colourings that are connected by a single Kempe change. Note that $C_k(G)$ is a subgraph of this graph, which we call the *Kempe k-colour graph*, and note that Kempe-equivalent colourings form the connected components of this graph. Questions similar to the ones we are interested in have been addressed in this context: Fisk [22] proved that all 4-colourings of an Eulerian triangulation of the plane are Kempe-equivalent, and Meyniel [49] that all 5-colourings of a planar graph are Kempe-equivalent. Later Las Vergnas and Meyniel [42] showed that the property of Kempe-equivalence holds for all 5-colourings of a graph containing no K_5 minor, and more recently, Mohar [51] has done so for all k-colourings of a planar graph with chromatic number less than k.

A generalisation in a different direction is considered by Brightwell and Winkler in [8, 9]. For a graph G and a *constraint graph* H (which may have loops), they define the graph $\mathrm{Hom}(G, H)$ as the graph with vertex set the homomorphisms from G to H, and two homomorphisms adjacent when they differ on precisely one vertex of G. (Recall that a homomorphism from G to H is a function $\varphi : V(G) \to V(H)$ such that for every $uv \in E(G)$ we have $\varphi(u)\varphi(v) \in E(H)$, and note that a k-colouring of a graph G is nothing more than a homomorphism from G to the complete graph K_k.) They investigate an important dichotomy of constraint graphs, giving several equivalent characterisations of graphs H which they call dismantlable. Letting u, v be two vertices of a finite H with $N(u) \subseteq N(v)$, where $N(x)$ denotes the set of neighbours of a vertex x, they define a *fold* of H as the homomorphism from H to $H - \{u\}$ mapping u to v and every other node to itself. The graph H is said to be *dismantlable* if there exists a sequence of folds reducing H to a graph with one node (looped or not). Amongst other things, they address the question of connectedness of $\mathrm{Hom}(G, H)$. In particular, they prove that a constraint graph H is dismantlable if and only if it is true that for any finite graph G, $\mathrm{Hom}(G, H)$ is connected.

Mixing Boolean satisfiability solutions

Remarkably similar results to those contained in this thesis, but for a wholly different problem, are to be found in [26]. The authors of [26] consider the exact analogues of our decision problems k-MIXING and k-COLOUR PATH in the context of the Boolean satisfiability problem. We proceed to examine their results in some detail, first giving some necessary definitions.

A *logical relation* R is a non-empty subset of $\{0, 1\}^k$, where $k \geq 1$ is the *arity* of R. For S a finite set of logical relations, a $\mathrm{CNF}(S)$-*formula* over a set of variables $V = \{x_1, x_2, \ldots, x_n\}$ is a finite conjunction $C_1 \wedge C_2 \wedge \ldots \wedge C_m$ of clauses built using relations from S, variables from V, and the constants and 0 and 1. Hence each C_i is an expression of the form $R(\xi_1, \xi_2, \dots, \xi_k)$, where R is a relation of arity k, and each ξ_j is a variable in V or one of the constants 0, 1.

The *satisfiability problem* $\text{SAT}(S)$ associated with a finite set of logical relations S asks: given a CNF(S)-formula φ, is it satisfiable? Schaefer [58] proved a celebrated dichotomy theorem for the complexity of $\text{SAT}(S)$: for certain sets S—known as *Schaefer* sets—$\text{SAT}(S)$ is solvable in polynomial time, while for all other sets S, the problem is NP-complete. We refer the reader to [58] for the full details; a definition of Schaefer sets may also be found in [26].

For an instance φ of $\text{SAT}(S)$, the authors of [26] define the graph $G(\varphi)$ as the graph with vertex set the satisfying assignments of φ, and assignments adjacent whenever they differ in exactly one bit. The graph $G(\varphi)$ is a subgraph of the n-dimensional hypercube—this is the graph with vertex set $\{0, 1\}^n$ and edges between vertices that differ in exactly one bit. Hence a path in $G(\varphi)$ corresponds to a sequence of different satisfying assignments of φ, each obtained from the previous one by flipping precisely one bit.

They define the following two decision problems, whose close resemblance to k-MIXING and k-COLOUR PATH should be obvious.

CONN(S)
Instance: A CNF(S)-formula φ.
Question: Is $G(\varphi)$ connected?

ST-CONN(S)
Instance: A CNF(S)-formula φ and two satisfying assignments of φ, $\mathbf{s}$ and $\mathbf{t}$.
Question: Is there a path between $\mathbf{s}$ and $\mathbf{t}$ in $G(\varphi)$?

The authors of [26] prove dichotomy theorems for the complexity of both of these decision problems. They also prove a dichotomy theorem for the possible diameter of the graphs $G(\varphi)$, finding, for both problems, the same correspondence between PSPACE-complete instances and possible superpolynomial-length shortest-paths in the graph of satisfying assignments as we do for k-COLOUR PATH. The key concept on which their results rely is that of a *tight* set of relations S—see [26] for a precise definition of this concept. The class of tight sets of relations properly contains the class of Schaefer relations: if S is Schaefer, then S is tight; the converse, however, is not true.

In some detail, they prove the following results.

Theorem 7.1 (Gopalan, Kolaitis, Maneva and Papadimitriou [26])
Let S be a finite set of logical relations. If S is tight, then $\textsc{Conn}(S)$ *is in* coNP*; if it is tight but not Schaefer, then it is* coNP*-complete; otherwise, it is* PSPACE*-complete.*

Theorem 7.2 (Gopalan, Kolaitis, Maneva and Papadimitriou [26])
Let S be a finite set of logical relations. If S is tight, then $\textsc{st-Conn}(S)$ *is in* P*; otherwise, it is* PSPACE*-complete.*

Theorem 7.3 (Gopalan, Kolaitis, Maneva and Papadimitriou [26])
Let S be a finite set of logical relations. If S is tight, then for every CNF*(S)-formula φ, the diameter of any component of $G(\varphi)$ is linear in the number of variables of φ; otherwise, there are* CNF*(S)-formulas φ such that $G(\varphi)$ has some component with diameter superpolynomial in the number of variables of φ.*

The authors of [26] in fact conjectured a trichotomy for the complexity of $\textsc{Conn}(S)$, claiming that if S is Schaefer, then $\textsc{Conn}(S)$ is actually in P (and showing that this is true for a particular type of Schaefer sets). This conjecture was recently disproved in [46], where a set of Schaefer relations for which the problem $\textsc{Conn}(S)$ remains coNP-complete is exhibited. In a recent updated version of [26], Gopalan, Kolaitis, Maneva and Papadimitriou [27] formulate a (modified) trichotomy conjecture for the complexity of $\textsc{Conn}(S)$, where it only remains to determine the complexity of $\textsc{Conn}(S)$ for a certain subset of Schaefer sets of relations.

We summarise their results, along with those of Schaefer [58], in Table 7.1 below.

S	$\textsc{Sat}(S)$	$\textsc{Conn}(S)$	$\textsc{st-Conn}(S)$	Diameter
Schaefer	P	coNP	P	$O(n)$
Tight, non-Schaefer	NP-complete	coNP-complete	P	$O(n)$
Non-tight	NP-complete	PSPACE-compl.	PSPACE-compl.	$2^{\Omega(\sqrt{n})}$

Table 7.1 The complexity of $\textsc{Sat}(S)$, $\textsc{Conn}(S)$ and $\textsc{st-Conn}(S)$, together with the possible diameter of components of $G(\varphi)$, for various types of relation sets S.

We note that despite the close parallelism between the results presented in this thesis and those of [26], the proofs are, in each case, very different.

7.2 Discussion and open problems

We have studied the basic properties of mixing, exploring the relationship between the mixing properties of a graph and certain graph invariants, notably the chromatic number and the degeneracy.

We have also obtained strong results for the computational complexity of the decision problems k-MIXING and k-COLOUR PATH. In particular, we have settled the complexity 3-MIXING, finding an important distinction between the general problem and its restriction to planar graphs. (Given that most NP-complete decision problems relating to 3-colouring become no easier for planar graphs, it is a curious fact that 3-MIXING, a coNP-complete problem, becomes polynomial time solvable when restricted to planar graphs.) We have also characterised those graphs which are 3-mixing.

The complexity of k-COLOUR PATH has also been settled, and an important and what appears to be fundamental relationship between the tractability of the problem and its underlying structure has been established. In terms of the number of colours k and the degeneracy $\deg(G)$ of the instance graph, we have proved a full dichotomy for the complexity of k-COLOUR PATH. If $k \leq 3$ or $k \geq \deg(G) + 2$, the problem is in P. In all other cases, the problem is PSPACE-complete (note that the reductions that prove Theorem 5.1 yield instances with $\deg(G) = k - 1$). Moreover, we have seen how this completely determines the complexity of k-COLOUR PATH for planar and bipartite planar graphs.

We have also shown that for $k \leq 3$ or $k \geq 2\deg(G) + 1$, the components of $\mathcal{C}_k(G)$ always have quadratic diameter. On the other hand, for $4 \leq k \leq \deg(G) + 1$, there exist graphs whose k-colour graph has components of superpolynomial diameter (the reader can easily verify that the graphs of Theorem 5.2 also have degeneracy $k - 1$). Thus it remains to determine whether for every graph G, the diameter of $\mathcal{C}_k(G)$ is polynomial (perhaps even quadratic) in the size of G when $k \geq 4$ and $\deg(G) + 1 < k < 2\deg(G) + 1$. If true, this would provide a complete correspondence between the PSPACE-completeness of k-COLOUR PATH and possible superpolynomial diameter components in $\mathcal{C}_k(G)$, according to our classification of instances by number of colours and degeneracy.

Our most obvious open problem is determining the complexity of k-MIXING for $k \geq 4$. An intimately related problem is of course finding a characterisation theorem for k-mixing graphs. Using the fact that k-COLOUR PATH is in PSPACE, Claim 5.6, we can at least determine that k-MIXING is in PSPACE.

Claim 7.4
The decision problem k-MIXING is in the complexity class PSPACE.

Proof. Given a graph G with n vertices, we can determine whether its k-colour graph is connected using a polynomial (in n) amount of space by the following procedure. Let us assume that the vertex set of G is $\{v_1, v_2, \ldots, v_n\}$ and observe that given a string $s = s_1 s_2 \ldots s_n$ from $\{1, 2, \ldots, k\}^n$ we can check in polynomial space whether or not this corresponds to a proper k-colouring α of G where $\alpha(v_i) = s_i$. Now, given two strings s and s' from $\{1, 2, \ldots, k\}^n$ corresponding to k-colourings of G, Claim 5.6 tells us that checking whether these colourings are connected in $\mathcal{C}_k(G)$ also takes a polynomial amount of space. Given these observations, all we need to do is sequentially run through all k^{2n} pairs of strings (using the obvious ordering), checking whether or not they correspond to colourings of G, and if they do, then checking if they are connected in $\mathcal{C}_k(G)$. Because we are running through the strings in order, at each stage we can re-use our working space, which is always polynomially bounded. $\square$

A first step towards determining the complexity of k-MIXING (for $k = 4$, at least) might be provided by an answer to the following question. Let G be a 3-chromatic graph and let α and β be two 3-colourings of G not connected in $\mathcal{C}_3(G)$. Note that by Theorem 6.1 these colourings are connected in $\mathcal{C}_5(G)$. What is the complexity of deciding if they are connected in $\mathcal{C}_4(G)$?

Our main results, together with the complexity of k-COLOURABILITY, are summarised in Table 7.2 below.

k	k-COLOURABILITY	k-MIXING	k-COLOUR PATH	Diameter
2	P	P	P	0
3	NP-complete	coNP-complete	P	$O(n^2)$
≥ 4	NP-complete	PSPACE	PSPACE-complete	$2^{\Omega(\sqrt{n})}$

Table 7.2 The complexity of k-COLOURABILITY, k-MIXING and k-COLOUR PATH, together with the possible diameter of components of $\mathcal{C}_k(G)$, for different values of k.

It is very interesting to compare the results from Tables 7.1 and 7.2: the similarity between them is striking. The comparison suggests that k-MIXING might well be PSPACE-complete for $k \geq 4$; this is also hinted at by the complexity of k-COLOUR PATH. If true, this would provide an example of a decision problem exhibiting a trichotomy of complexity, much the same as CONN(S) would, if the conjecture in [27] (that for any set S of relations, CONN(S) is PSPACE-complete, coNP-complete, or in P) is true.

Given the similarity between Tables 7.1 and 7.2, it would be interesting to try to find a relationship between the problems, perhaps expressing the problems k-MIXING and k-COLOUR PATH within the framework of [26]. This seems unlikely to be straightforward. A standard first approach would be to encode a k-colouring of a graph as a satisfiability problem by introducing a variable for every vertex, colour pair (v, c) which is set to true when v is coloured c. This, however, would not yield a correspondence between flipping bits in the graph of satisfying assignments and recolouring vertices of the graph being coloured.

Let us briefly turn our attention to list-colouring versions of our problems. We saw in Chapter 5 that the problem LIST-COLOUR PATH is PSPACE-complete, and that instances of this problem all have colour lists contained in $\{1, 2, 3, 4\}$. The reader will have no trouble verifying, however, that the reduction that proves the PSPACE-hardness of this problem (from SLIDING TOKENS) actually yields instances where each colour list has size at most 3. Hence the problem equivalent to 3-COLOUR PATH for list-colourings is PSPACE-complete. This fact has also been independently observed by Jan van den Heuvel and Zsolt Tuza, who also proved that for colour lists of size at most 2, the problem is solvable in polynomial time, [32]. In this case the list colouring problem is reducible to a 2-SAT problem where the (v, c)-encoding mentioned above yields a correspondence between flipping bits in the graph of satisfying assignments and recolouring vertices of the graph. Then a result of [26] shows it is possible to verify the connectedness of two satisfying assignments (list-colourings) in polynomial time.

It would also be interesting to further explore the properties of colour graphs themselves. For example, what sort of structures might we find in colour graphs? On the other hand, what graphs can occur as colour graphs? Let us mention at this point two results related to the structure of colour graphs, which we phrase in the terminology of this thesis. In the context of finding Gray codes for k-colourings of a graph G, MacGillivray and Choo [15] prove that if $k \geq \deg(G) + 3$, then $\mathcal{C}_k(G)$ is Hamiltonian. Mačaj [45] proves, further to a study of the metric structure of the category of finite sets and mappings between them, that the k-colour graph of the complete graph K_n is vertex-transitive for any $k > n$, and that for these same values the automorphism group of $\mathcal{C}_k(K_n)$ is in fact isomorphic to $S_n \times S_k$, where S_m denotes the symmetric group.

We recall two other questions that this thesis leaves unanswered. One, what is the mixing number of the Klein bottle? And two, is it true that the algorithm for 3-Colour Path from Chapter 4 can be implemented so as to always find a shortest path between two given 3-colourings?

Closing, we mention a question related to the rapid mixing of Markov chains for sampling colourings, the field from whence the inspiration for this thesis arose. If $\mathcal{C}_k(G)$ is not connected, what might be sensible edges to add between certain k-colourings to ensure that it *is* connected? That is, what additional moves might ensure that the state space of the chain is irreducible? This is a question that is often addressed when trying to obtain efficient algorithms for sampling k-colourings of particular graphs, but can anything be said in general?

Appendix

Non-deterministic constraint logic

We describe the non-deterministic constraint logic (NCL) model of computation of Hearn and Demaine [30], together with some associated decision problems. We also describe how the restricted instances of SLIDING TOKENS used to prove the PSPACE-completeness of k-COLOUR PATH in Theorem 5.1 arise.

An NCL *machine* is specified by an undirected graph together with an assignment of non-negative integer weights to its vertices and edges; the vertex weights are *minimum in-flow constraints.* A *configuration* of the machine is specified by an orientation of its edges such that the sum of incoming edge-weights at each vertex is at least the minimum in-flow constraint of the vertex. A *move* from one configuration to another is simply the reversal of a particular edge direction such that all minimum in-flow constraints remain satisfied.

The authors of [30] present the following three decision problems associated with NCL machines.

1. Given an NCL machine together with a particular configuration, can a specified edge be eventually reversed by some sequence of moves?
2. Given an NCL machine together with two particular configurations A and B, is there a sequence of moves from A to B?
3. Given two edges e_A and e_B of an NCL machine, and orientations for each, are there configurations A and B such that e_A has its desired orientation in A, e_B has its desired orientation in B, and there is a sequence of moves from A to B?

We remark that it is the second problem that we use for our definition of SLIDING TOKENS in Chapter 5, after some suitable transformations which we now describe.

It turns out that certain vertex configurations in NCL machines are of particular interest. A vertex with minimum in-flow constraint 2 and three incident edges with weights $1, 1, 2$ behaves as a logical AND: the edge with weight 2 can be directed outwards only when the other two edges are directed inwards. Such a vertex is called an AND vertex. Likewise, a vertex with minimum in-flow constraint 2 and three incident edges with weights $2, 2, 2$ behaves as a logical OR: a given edge may be directed outward if and only if at least one of the other two is directed inwards. Such a vertex is called an OR vertex.

The authors of [30] claim without proof that every NCL graph is reducible in logarithmic space to an equivalent (in terms of the given decision problems) AND/OR constraint graph—this is a graph composed exclusively of AND and OR vertices. They then prove all three of the above decision problems to be PSPACE-complete for such graphs. Note that this unproved claim is not used in our reductions, and is therefore unnecessary for our results: our reductions are always from AND/OR constraint graphs, or rather, their sliding-token versions (see below).

The three problems are then shown to remain PSPACE-complete for 3-connected planar AND/OR graphs; this is achieved by the construction of a suitable crossover gadget and a suitable connectivity-augmentation gadget. After describing some applications of NCL and the above decision problems by proving strong PSPACE-completeness results for a variety of sliding-block puzzles, the final section of [30] contains an alternative formulation of AND/OR constraint graphs in terms of sliding tokens along graph edges.

In this context, the 'machine' is again an undirected graph G. A *token configuration* of a graph G is a set of vertices on which tokens are placed, in such a way that no two tokens are adjacent. (Thus a token configuration can be thought of as an independent set of vertices of G.) A *move* between two token configurations is the displacement of a token from one vertex to an adjacent vertex. Note that a move must result in a valid token configuration.

The simulation of NCL AND and OR vertices via sliding-token gadgets is depicted in Figure A.1. The gadgets are in fact the vertex configurations within the dotted lines and the edges that cross the dotted lines are termed *port-edges*—these connect the gadgets. For the AND sliding-token gadget, the two lower port-edges correspond to the edges of an NCL AND vertex with weight 1. A token on an outer port-edge vertex represents an NCL edge directed inwards, and a token on an inner port-edge vertex represents an edge directed outwards.

Hence given an AND/OR constraint graph and configuration, a corresponding sliding-token graph can be constructed by joining AND and OR vertex gadgets and placing

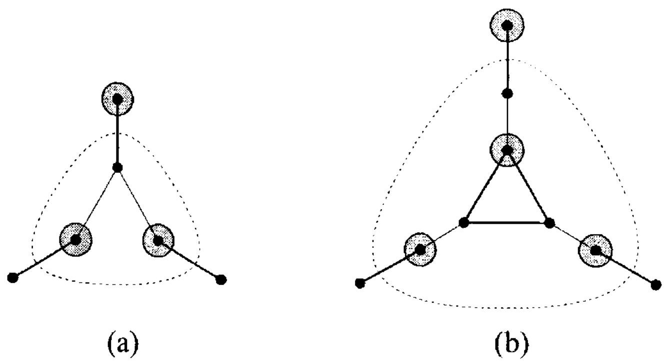

Figure A.1 (a) an AND sliding-token gadget, and (b) an OR sliding-token gadget.

the port tokens appropriately. Moreover, it is not hard to see that such a sliding-token instance is equivalent to the original NCL instance. The AND gadget satisfies the same constraints as an NCL AND vertex: the upper token can slide in precisely when both lower tokens are slid out. Similarly, the OR gadget satisfies the same constraints as an NCL OR vertex: the upper token can slide in when either lower token is slid out—the internal token can then be displaced to allow the upper token to slide in.

We finish by making some remarks about the way we describe these sliding-token instances in Chapter 5. The *token triangles* of our SLIDING TOKENS instances—copies of K_3—are precisely the triangles in OR configurations; *token edges*—copies of K_2—are the port edges on the boundaries of both AND and OR configurations. Because the original instances of NCL can be taken to be planar, we can see that every sliding-token instance has a planar embedding where every token triangle bounds a face. Moreover, because the NCL instances can be taken to be 3-connected, every sliding-token gadget is connected to three other gadgets and so we can take our instances of SLIDING TOKENS to have minimum degree 2.

Bibliography

[1] M.O. Albertson and J.P. Hutchinson, The three excluded cases of Dirac's map-color theorem. *Ann. New York Acad. Sci.* **319** (1979), 7–17.

[2] K. Appel and W. Haken, Every planar map is four colorable. Part I. Discharging. *Illinois J. Math.* **21** (1977), 429–490.

[3] K. Appel, W. Haken and J. Koch, Every planar map is four colorable. Part II. Reducibility. *Illinois J. Math.* **21** (1977), 491–567.

[4] V. Barbéra and B. Jaumard, Design of an efficient block retuning. *Mobile Netw. Appl.* **6** (2001), 501–510.

[5] R.J. Baxter, *Exactly Solved Models in Statistical Mechanics.* Academic Press, New York, 1982.

[6] J. Billingham, R.A. Leese and H. Rajaniemi, Frequency reassignment in cellular phone networks. Smith Institute Study Group Report (2005). Available from `http://www.smithinst.ac.uk/Projects/ESGI53/ESGI53-Motorola/Report`.

[7] P. Bonsma and L. Cereceda, Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances. Submitted. [A conference proceedings version of this article can be found in *Proceedings of Mathematical Foundations of Computer Science, 32nd International Symposium, MFCS 2007. Lect. Notes Comput. Sci.* **4708** (2007), 738–749.]

[8] G.R. Brightwell and P. Winkler, Graph homomorphisms and phase transitions. *J. Combin. Theory Ser. B* **77** (1999), 221–262.

[9] G.R. Brightwell and P. Winkler, Gibbs measures and dismantlable graphs. *J. Combin. Theory Ser. B* **78** (2000), 141–166.

[10] L. Cereceda, J. van den Heuvel and M. Johnson, Connectedness of the graph of vertex-colourings. *Discrete Math.* **308**(5–6) (2008), 913–919.

[11] L. Cereceda, J. van den Heuvel and M. Johnson, Mixing 3-colourings in bipartite graphs. Submitted. [A conference proceedings version of this article can be found in *Proceedings of the 33rd International Workshop on Graph-Theoretic Concepts in Computer Science, WG 2007. Lect. Notes Comput. Sci.* **4769** (2007), 166–177.]

[12] L. Cereceda, J. van den Heuvel and M. Johnson, Finding paths between 3-colourings. Submitted.

[13] Y.M. Chee and A. Lim, The algorithmic complexity of colour switching. *Inform. Process. Lett.* **43** (1992), 63–68.

[14] M. Chlebík and J. Chlebíková, Hard coloring problems in low degree planar bipartite graphs. *Discrete Appl. Math.* **154**(14) (2006), 1960–1965.

[15] K. Choo and G. MacGillivray, Gray code numbers for graphs. Preprint (2006).

[16] D.P. Dailey, Uniqueness of colorability and colorability of planar 4-regular graphs are NP-complete. *Discrete Math.* **30** (1980), 289–293.

[17] R. Diestel, *Graph Theory*, 3rd edition. Springer-Verlag, Heidelberg, 2005.

[18] G.A. Dirac, Map colour theorems related to the Heawood colour formula. *J. London Math. Soc.* **31** (1956), 460–471.

[19] G.A. Dirac, Short proof of a map-colour theorem. *Canad. J. Math.* **9** (1957), 225–226.

[20] M. Dyer, A. Flaxman, A. Frieze and E. Vigoda, Randomly colouring sparse random graphs with fewer colours than the maximum degree. *Random Structures Algorithms* **29**(4) (2006), 450–465.

[21] L. Euler, Solutio problematis ad geometriam situs pertinentis. *Comment. Academiae Sci. I. Petropolitanae* **8** (1741) 128–140. Available from `http://math.dartmouth.edu/~euler/pages/E053.html`.

[22] S. Fisk, Geometric coloring theory. *Adv. Math.* **24** (1977), 298–340.

[23] P. Franklin, A six color problem. *J. Math. Phys.* **13** (1934), 363–379.

[24] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness.* Freeman, New York, 1979.

[25] L.A. Goldberg, R. Martin and M. Paterson, Random sampling of 3-colorings in $\mathbb{Z}^2$. *Random Structures Algorithms* **24**(3) (2004), 279–302.

[26] P. Gopalan, P.G. Kolaitis, E.N. Maneva and C.H. Papadimitriou, The connectivity of Boolean satisfiability: computational and structural dichotomies. In *Proceedings of Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Part I. Lect. Notes Comput. Sci.* **4051** (2006), 346–357.

[27] P. Gopalan, P.G. Kolaitis, E.N. Maneva and C.H. Papadimitriou, The connectivity of Boolean satisfiability: computational and structural dichotomies. (Revised version.) Preprint (2007). Available from `http://arxiv.org/abs/cs.CC/0609072v2`.

[28] W.K. Hale, Frequency assignment: theory and applications. *Proc. IEEE* **68**(12) (1980), 1497–1514.

[29] J. Han, Frequency reassignment problem in mobile communication networks. *Comput. Oper. Res.* **34** (2007), 2939–2948.

[30] R.A. Hearn and E.D. Demaine, PSPACE-completeness of sliding-block puzzles and other problems through the nondeterministic constraint logic model of computation. *Theoret. Comput. Sci.* **343** (2005), 72–96.

[31] P.J. Heawood, Map colour theorem. *Quart. J. Math. Oxford Ser.* **24** (1890), 332–338.

[32] J. van den Heuvel, personal communication.

[33] P. Hliněný, A note on possible extensions of Negami's conjecture. *J. Graph Theory* **32** (1999), 234–240.

[34] M. Hujter and Zs. Tuza, Precoloring extension II. Graph classes related to bipartite graphs. *Acta Math. Univ. Comenian.* **LXII**(1) (1993), 1–11.

[35] R. Jacob, *Standortplanung mit Blick auf Online-Strategien.* Graduate Thesis, University of Würzburg, 1997. Available from `http://www14.in.tum.de/personen/jacob/Publications/index.html`.

[36] J. Janssen, *Channel Assignment and Graph Labeling.* John Wiley & Sons, New York, 2002.

[37] T.R. Jensen and B. Toft, *Graph Coloring Problems.* Wiley-Interscience, New York, 1995.

[38] M. Jerrum, A very simple algorithm for estimating the number of k-colourings of a low degree graph. *Random Structures Algorithms* **7** (1995), 157–165.

[39] M. Jerrum, *Counting, Sampling and Integrating: Algorithms and Complexity.* Birkhäuser Verlag, Basel, 2003.

[40] J. Kratochvíl, Perfect codes in general graphs. *Colloq. Math. Soc. János Bolyai* **52** (1987), 357–364.

[41] J. Kratochvíl, Precoloring extension with fixed color bound. *Acta Math. Univ. Comenian.* **LXII**(2) (1993), 139–153.

[42] M. Las Vergnas and H. Meyniel, Kempe classes and the Hadwiger conjecture. *J. Combin. Theory Ser. B* **31** (1981), 95–104.

[43] R.A. Leese and S. Hurley (Eds.), *Methods and Algorithms for Radio Channel Assignment.* Oxford University Press, Oxford, 2003.

[44] T. Luczak and E. Vigoda, Torpid mixing of the Wang-Swendsen-Kotecký algorithm for sampling colorings. *J. Discrete Alg.* **3** (2005), 92–100.

[45] M. Mačaj, Vertex-transitive graphs and 2-transitive groups I. Injective mappings. Preprint (2001).

[46] K. Makino, S. Tamaki and M. Yamamoto, On the Boolean connectivity problem for Horn relations. In *Proceedings of the 10th International Conference on Theory and Applications of Satisfiability Testing (SAT) 2007, Lect. Notes Comput. Sci.* **4501** (2007), 187–200.

[47] O. Marcotte and P. Hansen, The height and length of colour switching. In P. Hansen and O. Marcotte (Eds.), *Graph Colouring and Applications, Proceedings of the Centre de Réchérches Mathématiques, Vol. 23.* Oxford University Press, Oxford, 1999.

[48] C.J.H. McDiarmid, Discrete Mathematics and Radio Channel Assignment. In B.A. Reed and C. Linhares-Salas (Eds.), *Recent Advances in Algorithms and Combinatorics.* Springer-Verlag, New York, 2003.

[49] H. Meyniel, Les 5-colorations d'un graphe planaire forment une classe de commutation unique. *J. Combin. Theory Ser. B* **24** (1978), 251–257.

[50] B. Mohar, A linear time algorithm for embedding graphs in an arbitrary surface. *SIAM J. Discrete Math.* **12** (1999), 6–26.

[51] B. Mohar, Kempe equivalence of colorings. In J.A. Bondy, J. Fonlupt, J.L. Fouquet, J.-C. Fournier, and J. Ramírez Alfonsín (Eds.), *Proceedings of Graph Theory in Paris, a Conference in Memory of Claude Berge.* Birkhäuser, 2006.

[52] B. Mohar and C. Thomassen, *Graphs on Surfaces.* The Johns Hopkins University Press, Baltimore, 2001.

[53] C.H. Papadimitriou, *Computational Complexity.* Addison-Wesley, Boston, 1994.

[54] T.J. Rackham, A precolouring extension of Brooks' theorem. Preprint (2006). Available from `http://www.maths.ox.ac.uk/~rackham/pebt2006.pdf`.

[55] G. Ringel and J.W.T. Youngs, Solution of the Heawood map-coloring problem. *Proc. Nat. Acad. Sci. U.S.A.* **60** (1968), 438–445.

[56] N. Robertson, D.P. Sanders, P.D. Seymour and R. Thomas, The four colour theorem. *J. Combin. Theory Ser. B* **70** (1997), 2–44.

[57] W.J. Savitch, Relationships between nondeterministic and deterministic tape complexities. *J. Comput. System Sci.* 4(2) (1970), 177–192.

[58] T.J. Schaefer, The complexity of satisfiability problems. *Proc. 10th Annual ACM Symp. on Theory of Computing* (1978), 216–226.

[59] E. Vigoda, Improved bounds for sampling colorings. *J. Math. Phys.* **41** (2000), 1555–1569.

[60] N. Vikas, Computational complexity of compaction to irreflexive cycles. *J. Comput. System. Sci.* **68** (2004), 473–496.

[61] D.B. West, *Introduction to Graph Theory*, 2nd edition. Prentice-Hall, New Jersey, 2001.