Finding Paths between Graph Colourings: PSPACE-completeness and Superpolynomial Distances

Paul Bonsma*
Institut für Mathematik, Sekr. MA 6-1, Technische Universität Berlin,
Straße des 17. Juni 136, 10623 Berlin, Germany
bonsma @ math.tu-berlin.de

and

Luis Cereceda
Centre for Discrete and Applicable Mathematics
Department of Mathematics, London School of Economics
Houghton Street, London WC2A 2AE, U.K.
luis@maths.lse.ac.uk

18th May 2007

Abstract

Suppose we are given a graph G together with two proper vertex k-colourings of G, α and β. How easily can we decide whether it is possible to transform α into β by recolouring vertices of G one at a time, making sure we always have a proper k-colouring of G? This decision problem is trivial for $k = 2$, and decidable in polynomial time for $k = 3$. Here we prove it is PSPACE-complete for all $k \geq 4$. In particular, we prove that the problem remains PSPACE-complete for bipartite graphs, as well as for: (i) planar graphs and $4 \leq k \leq 6$, and (ii) bipartite planar graphs and $k = 4$. Moreover, the values of k in (i) and (ii) are tight, in the sense that for larger values of k, it is always possible to recolour α to β.

We also exhibit, for every $k \geq 4$, a class of graphs $\{G_{N,k} : N \in \mathbb{N}^*\}$, together with two k-colourings for each $G_{N,k}$, such that the minimum number of recolouring steps required to transform the first colouring into the second is superpolynomial in the size of the graph: the minimum number of steps is $\Omega(2^N)$, whereas the size of G_N is $O(N^2)$. This is in stark contrast to the $k = 3$ case, where it is known that the minimum number of recolouring steps is at most quadratic in the number of vertices. We also show that a class of bipartite graphs can be constructed with this property, and that: (i) for $4 \leq k \leq 6$ planar graphs and (ii) for $k = 4$ bipartite planar graphs can be constructed with this property. This provides a remarkable correspondence between the tractability of the problem and its underlying structure.

Keywords: vertex-recolouring, colour graph, PSPACE-complete, superpolynomial distance.

*Supported by the Graduate School "Methods for Discrete Structures" in Berlin, DFG grant GRK 1408

1 Introduction

Throughout this paper, graphs will be finite, simple and loopless. Most of our terminology and notation is standard and can be found in any textbook on graph theory such as, for example, [4]. Standard references for complexity theory are [6] and [9]. We always regard a k-colouring of a graph $G = (V, E)$ as proper; that is, as a function $\alpha : V \to \{1, 2, \ldots, k\}$ such that $\alpha(u) \neq \alpha(v)$ for all $uv \in E$. For a positive integer k and a graph G, we define the *k-colour graph of G*, denoted $\mathcal{C}_k(G)$, as the graph that has the k-colourings of G as its node set, with two k-colourings joined by an edge in $\mathcal{C}_k(G)$ if they differ in colour on just one vertex of G. We assume throughout that $k \geq \chi(G) \geq 2$, where $\chi(G)$ is the chromatic number of G. Having defined the colourings as nodes of $\mathcal{C}_k(G)$, the meaning of a path between two colourings should be clear. In addition, other graph-theoretical notions such as distance and adjacency can now be used for colourings. A path between two given colourings in $\mathcal{C}_k(G)$ can also shortly be characterised by a *sequence of recolourings*, which is an ordered list consisting of pairs composed of a vertex and a new colour for the vertex. If $\mathcal{C}_k(G)$ is connected, we say that G is *k-mixing*. We use the term *frozen* for a k-colouring of a graph G that forms an isolated node in $\mathcal{C}_k(G)$. Note that the existence of a frozen k-colouring of a graph immediately implies that the graph is not k-mixing.

In [1, 2], some preliminary investigations into the connectedness of the k-colour graph are made. In particular, [2] settles the computational complexity of the following decision problem: given a 3-colourable graph G, is G 3-mixing? This problem is proved to be coNP-complete for bipartite graphs but polynomial-time solvable for bipartite planar graphs. For G a 3-chromatic graph, the answer is always in the negative.

A related problem is that of recognising when two given k-colourings of a graph G are in the same connected component of $\mathcal{C}_k(G)$. Formally, we have the following decision problem:

k-Colour Path
Instance : Graph G, two k-colourings of G, α and β.
Question : Is there a path between α and β in $\mathcal{C}_k(G)$?

It is easy to see that 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. For this reason we will always take our "argument graph" G to be connected.

The problem 2-Colour Path is trivial: the 2-colour graph of a connected bipartite graph always consists of two isolated nodes.

For 3-colourings, we have:

Theorem 1.1 ([3]) *The decision problem* 3-Colour Path *is in* P.

The proof of correctness of the polynomial-time algorithm for 3-Colour Path given in [3] can be employed to exhibit a path between the given 3-colourings, if such a path exists. Moreover, such a path has length $O(|V(G)|^2)$, proving:

Theorem 1.2 ([3]) *Let G be a 3-colourable graph with n vertices. Then the diameter of any component of $\mathcal{C}_3(G)$ is $O(n^2)$.*

Our first main result settles the computational complexity of k-Colour Path:

Theorem 1.3 *For every $k \geq 4$, 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 $k = 4$.

In terms of the well-known NP $\neq$ PSPACE conjecture, Theorem 1.3 means the following. Loosely speaking, having established that k-Colour Path is PSPACE-complete, asserting that NP $\neq$ PSPACE is *equivalent* to saying that there is no possible polynomial size certificate for k-Colour Path. Thus proving that every possible certificate can have superpolynomial size is a daunting task. In our second main result, however, we show that this is indeed the case for the most natural certificate for k-Colour Path: the certificate for a YES-instance consisting of a list of colourings constituting a path from the first colouring to the second colouring. More precisely, we prove:

Theorem 1.4 *For every $k \geq 4$, there exists a class of graphs $\{G_{N,k} : N \in \mathbb{N}^*\}$ with the following properties. The graphs $G_{N,k}$ have size $O(N^2)$, and for each of them there exist two k-colourings α and β 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 $k \in \{4, 5, 6\}$, the graphs $G_{N,k}$ may be taken to be planar (in such a case the graphs 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 have size $O(N^4)$).

The rest of the paper is organised as follows. In Section 2 we introduce the notions that will be used in the proofs. In Section 3 we prove Theorem 1.3 and also show that the values of k in parts (ii) and (iii) of the theorem are tight: for larger values of k, the instance is always a YES instance. This follows from a result that guarantees that for sufficiently large k, a graph will always be k-mixing. Section 4 is devoted to the proof of Theorem 1.4.

Theorems 1.1 to 1.4 together suggest that the computational complexity of k-Colour Path and the possible distance between k-colourings are intimately linked. How strong is this connection between PSPACE-completeness and superpolynomial distances in the colour graph? In particular, bearing in mind the tightness of k in (ii) and (iii) of Theorem 1.3: is it true that for a planar graph G and $k \geq 7$, or G a bipartite planar graph and $k \geq 5$, the components of $\mathcal{C}_k(G)$ always have polynomial diameter? (In these cases $\mathcal{C}_k(G)$ is actually connected—see Section 3.4.) We formulate this question more generally as a conjecture in Section 3.4, and give a partial answer. For completeness, we remark that artificial graph classes can be constructed for which k-Colour Path is easy, but which still contain instances with colourings at superpolynomial distance: the graph classes $\{G_{N,k} : N \in \mathbb{N}^*\}$ from Theorem 1.4 are examples of such classes.

It is very interesting to compare the work presented in this paper and [1, 2] with [7], which contains remarkably similar results. For a given instance φ of the Boolean satisfiability problem, the authors of [7] 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. They consider the analogous question to the one we address here: given φ together with two satisfying assignments, are the assignments in the same connected component of $G(\varphi)$? In consonance with our results, they find the same correspondence between PSPACE-complete instances of this decision problem and possible superpolynomial paths in the graph of satisfying assignments. (In a similar fashion to [1, 2], and again finding similar results, they also study the decision problem: given φ, is $G(\varphi)$ connected?) We note that despite the parallelism between the results, the proofs are, in each case, very different.

2 Preliminaries

2.1 List-colouring instances

In Sections 3 and 4 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 constraint 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 $v \in V(G)$, list-colourings 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 K_4 on vertex set $\{u_1, \ldots, u_4\}$. 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 shows more formally that this can be done for various k, also when we require planarity and bipartiteness to be maintained, without increasing the size of the graph too much.

Lemma 2.1 *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

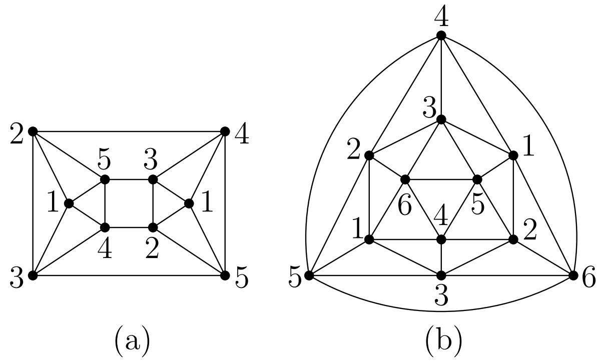

Figure 1: Planar graphs and with respective frozen 5- and 6-colourings.

(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, this can be done so 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 first 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. We proceed to describe such graphs and colourings.

Let L_k be the bipartite graph obtained from the balanced complete bipartite graph $K_{k,k}$ by removing the edges of a perfect matching in $K_{k,k}$. Consider the following k-colouring κ of L_k: colour the vertices in each part of the bipartition of L_k with the colours $1, 2, \ldots, k$, where vertices in opposite parts that were originally connected by an edge from the removed perfect matching are given the same colour. This colouring κ is a frozen colouring of L_k. Note that L_4 is just the 3-dimensional cube, which is a planar graph. So now we only need planar graphs with frozen k-colourings for $k = 5$ and $k = 6$. Such graphs and colourings are shown in Figures 1(a) and (b). (The graph second graph is actually the icosahedron.)

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 $c \in \{1, \ldots, k\} \backslash L(v)$, 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 $F_{v,c}$.

It is easy to see that every k-colouring obtainable from α' and β' 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 destroying planarity.

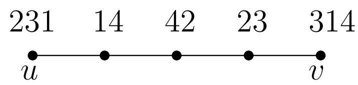

Figure 2: A $(1, 3)$-forbidding path from u to v.

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. $\square$

2.2 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 $a, b \in \{1, \dots, 4\}$, 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 the 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 2 shows an example of a $(1, 3)$-forbidding path from u to v. More formally, we have:

Definition 2.2 *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 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 every time; as long as $L(u), L(v) \neq \{1, 2, 3, 4\}$, such a path always exists.

Lemma 2.3 *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 $c \in \{1, 2, 3, 4\} \backslash L(u)$ and $d \in \{1, 2, 3, 4\} \backslash L(v)$. 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. 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 $e \in \{1, 2, 3, 4\}\backslash\{a, c\}$ and $f \in \{1, 2, 3, 4\}\backslash\{b, c\}$ with $e \neq f$. As before, it can be verified that this is an (a, b)-forbidding path with the desired properties. □

3 PSPACE-completeness of k-Colour Path for $k \geq 4$

3.1 Overview

In this section, we prove that k-Colour Path is PSPACE-complete for several graph classes and values of $k \geq 4$. 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 in [8]. We first reduce Sliding Tokens to List-Colour Path and then apply Lemma 2.1 to prove the existence of equivalent k-Colour Path instances. We first establish that k-Colour Path is indeed in PSPACE.

Claim 3.1 *The decision problem k-Colour Path is in 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 [9] p.150 or [10] 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. □

3.2 A PSPACE-complete problem: Sliding Tokens

The main result of [8] is the presentation of a new nondeterministic model of computation based on reversing edge directions in weighted directed graphs with minimum in-flow constraints on vertices. This model, called *nondeterministic 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 [8] gives an equivalent formulation of NCL in terms of sliding tokens along graph edges—it is this latter formulation that we will use for our reductions and which we now proceed to describe. Let us first give some definitions. 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 following two questions are proved to be PSPACE-complete in [8].

1. Given a graph G and a token configuration of G, can a specified token eventually be moved by some sequence of moves?
2. Given a graph G and two token configurations of G, is there a sequence of moves from one token configuration to the other?

Because we will be using the second of these questions in our reductions, we formally define the problem Sliding Tokens as follows.

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 [8] 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—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 some 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 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 may never leave its triangle or its edge: the first time any token would slide to another triangle or edge, it would become adjacent to the token belonging to this triangle or edge. So tokens may 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 3. (Token triangles and token edges are shown in bold; a circled vertex depicts a vertex on which a token is placed.) We insist: for these restricted instances, Sliding Tokens is PSPACE-complete. For further details, we refer the reader to [8].

3.3 The construction of equivalent List-Colour Path instances

Given a restricted instance G, T_A, T_B of Sliding Tokens as described in Section 3.2, 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'.

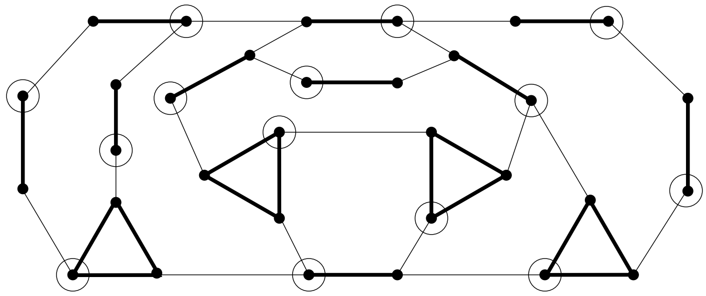

Figure 3: An example of a restricted instance graph G together with a standard token configuration.

We first label the vertices of G: the token triangles are labeled $1, \dots, n_t$, and the vertices of triangle i are labeled t_{i1}, t_{i2} and t_{i3}. The token edges are labeled $1, \dots, n_e$, and the vertices of token edge i are labeled 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 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 to T_A and T_B respectively 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 3.2 *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 3.3 *Let G, T_A, T_B be a resticted instance of* Sliding Tokens *as described in Section 3.2, and let G', L, α, β be the 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 and colourings 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 in 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 3.3 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 (Claim 3.2). Now by Lemma 2.1 we can construct equivalent k-Colour Path instances from G', L, α, β. All of these transformations are polynomial-time, and k-Colour Path is in PSPACE (Claim 3.1). This proves Theorem 1.3.

3.4 Tightness of the hardness results

Recall that the *colouring number* $\operatorname{col}(G)$ of a graph G (also known as the *degeneracy* or the *maximin degree*) is defined as the largest minimum degree of any subgraph of G. That is, $\operatorname{col}(G) = \max_{H \subseteq G} \delta(H)$. The following result appears in [1] but was essentially proved in [5]—we reproduce the proof given in [1] for completeness.

Theorem 3.4 *For any graph G and integer $k \geq \operatorname{col}(G) + 2$, 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 $d_G(v) \leq \operatorname{col}(G)$, and set $G' = G - v$. Note that $\operatorname{col}(G') \leq \operatorname{col}(G)$, hence we also have $k \geq \operatorname{col}(G') + 2$. By induction we can assume that G' is k-mixing.

Take two k-colourings α and β of G, and let α', β' be the k-colourings of G' induced by α, β. Since G' is k-mixing, there exists a sequence $\alpha' = \gamma'_0, \gamma'_1, \ldots, \gamma'_r = \beta'$ of k-colourings of G' so that for $i = 1, \ldots, r$, 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 the colour c_i. But because v has degree at most $\operatorname{col}(G) \leq k-2$, 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, and then continue with recolouring v_i to c_i and move on.

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. □

Recalling that the colouring number of a planar graph is at most 5, and that the colouring number of a bipartite planar graph is at most 3, Theorems 1.1, 1.3 and 3.4 together yield:

Theorem 3.5 *Restricted to planar graphs, the decision problem k-COLOUR PATH is PSPACE-complete for $4 \leq k \leq 6$, and polynomial-time solvable for all other values of k.*

Theorem 3.6 *Restricted to bipartite planar graphs, the decision problem k-COLOUR PATH is PSPACE-complete for $k = 4$, and polynomial-time solvable for all other values of k.*

We saw in Section 1 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 1.3 and 1.4 establish a connection between instance classes for which k-COLOUR PATH is PSPACE-complete and possible superpolynomial distances in the k-colour graph of these instances. We remark that the reason why we cannot make the values of k in parts (ii) and (iii) of Theorem 1.4 larger by a straightforward extension of our methods rests fundamentally on the fact that for a planar graph G, $\mathrm{col}(G) \leq 5$, and that for a bipartite planar graph G, $\mathrm{col}(G) \leq 3$. These considerations, together with Theorems 3.5 and 3.6, 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 $k \geq \mathrm{col}(G) + 2$, is it true that for any graph G and $k \geq \mathrm{col}(G) + 2$, $\mathcal{C}_k(G)$ always has polynomial diameter? We conjecture that this is indeed the case:

Conjecture 3.7 *For a graph G with n vertices and $k \geq \mathrm{col}(G) + 2$, the diameter of $\mathcal{C}_k(G)$ is $O(n^3)$.*

For values of $k \geq 2\,\mathrm{col}(G) + 1$, we are able to prove this statement, and even a stronger bound:

Claim 3.8 *For a graph G with n vertices and $k \geq 2\,\mathrm{col}(G) + 1$, the diameter of $\mathcal{C}_k(G)$ is $O(n^2)$.*

Proof: We can iteratively delete vertices of degree at most $\mathrm{col}(G)$ until no vertices are left. Using such an elimination ordering, we label the vertices $v_1, \ldots, v_n$ so that every vertex has at most $\mathrm{col}(G)$ 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 over n.

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 consider $G' = G - v_n$. 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 (they may be the same). Because there are at least $2\,\mathrm{col}(G) + 1$ available colours, and v_n has at most $\mathrm{col}(G)$ neighbors, a colour c can be chosen for v_n that is not equal to the old colour or new colour of any of its neighbors. 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, every time for a new vertex v_i which will have 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 are needed before it can be recoloured to $\beta(v_i)$. This will yield β after at most $O(n^2)$ recolouring steps. □

4 Graphs with colourings at superpolynomial distance

4.1 The construction of the graphs

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. 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 $\mathcal{C}(G_N, L)$ will have 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 $i = 1, \ldots, N$. 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. So:

Observation 4.1 *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 that:

Observation 4.2 *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.

4.2 Bounds on size and distance

Claim 4.3 *The sizes of $V(G_N)$ and $E(G_N)$ are both bounded by a function in $O(N^2)$.*

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 paths.

Because we may assume that all (a,b)-forbidding paths have length at most 6 (Lemma 2.3), we get $|V(G_N)| \leq 3N + 5N(N-1) \in O(N^2)$, and $|E(G_N)| \leq 3N + 6N(N-1) \in O(N^2)$. □

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 $(c_1, c_2, \ldots, c_N)$*-colouring* if $\kappa(v_i) = c_i$ for all i. All $(c_1, c_2, \ldots, c_N)$-colourings together form the *colour class* $(c_1, c_2, \ldots, c_N)$.

Observation 4.4 *Every colour class $(c_1, \ldots, c_N)$ with $c_i \in \{1,2\}$ 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 4.5 *Let $(x_1, \ldots, x_N)$ and $(y_1, \ldots, y_N)$ be distinct tuples with all $x_i, y_i \in \{1,2\}$.*

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

Proof : Suppose the above conditions on the tuples hold. We show that any colouring κ in the class $(x_1, \ldots, x_N)$ can be recoloured to a colouring in class $(y_1, \ldots, y_N)$. 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 $(x_1, \ldots, x_N)$-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 $(x_1, \ldots, x_N)$-colouring, let β be a $(y_1, \ldots, y_N)$-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

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

Figure 4: Colour classes visited in a shortest path between a $(1,1,1,1)$-colouring and a $(1,1,1,2)$-colouring of G_4.

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 4.5 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). 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 4 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. This is proved formally for every N in Theorem 4.6.

Theorem 4.6 *Every graph G_N has two colourings α and β in the same component of $\mathcal{C}(G_N, L)$ which are at distance at least $2^N - 1$.*

Proof : For the colouring α we choose a colouring in class $(1, \dots, 1)$; such a colouring exists by Observation 4.4. Colouring β will be a colouring in class $(1, \dots, 1, 2)$. We first prove by induction that such colourings exist and that they can be obtained from each other by recolourings, using the following induction hypothesis.

Induction hypothesis

There is a path in $\mathcal{C}(G_N, L)$ 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 4.5. 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 4.5. 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 $(1, \dots, 1, 1, x_1, \dots, x_{N-n})$-colouring to a $(1, \dots, 1, 2, y_1, \dots, y_{N-n})$-colouring, at least $2^n - 2$ other colour classes need to be visited.

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 4.5 (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 4.3 and Theorem 4.6 show that G_N with its colour lists L is a list-colouring instance such that $\mathcal{C}(G_N, L)$ has a component of diameter superpolynomial in the size of G_N. In the next sections, we use the graphs G_N to construct bipartite and planar k-colouring instances for various k with the same property.

4.3 Making the graphs planar and bipartite

In this section we show that the List-Colour Path instances constructed in Sections 4.1 and 4.2 can be used to construct k-Colour Path instances with the same properties, for various graph classes. For this, we will again apply Lemma 2.1. Unfortunately, the graphs G_N constructed in Section 4.1 are neither bipartite or planar. We now show how these instances G_N, L, α, β can be turned into bipartite and planar instances.

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 triangle i, we replace the edge $v_i v_i^*$ by a $(3, 3)$-forbidding path of even length. This yields an even cycle, and does not influence the possible colourings and recolourings of v_i and v_i^*. All other forbidding paths can also be chosen of even length (Lemma 2.3). Since all forbidding paths in the graph have vertices v_i or v_i^* for some i as their end vertices, the resulting graph is bipartite and has all vertices v_i and v_i^* in the same part of the bipartition. 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 4.7 *The graphs G_N^B have $O(N^2)$ vertices and 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 4.10 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 they form an (a, b)-forbidding path:

Observation 4.8 *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.*

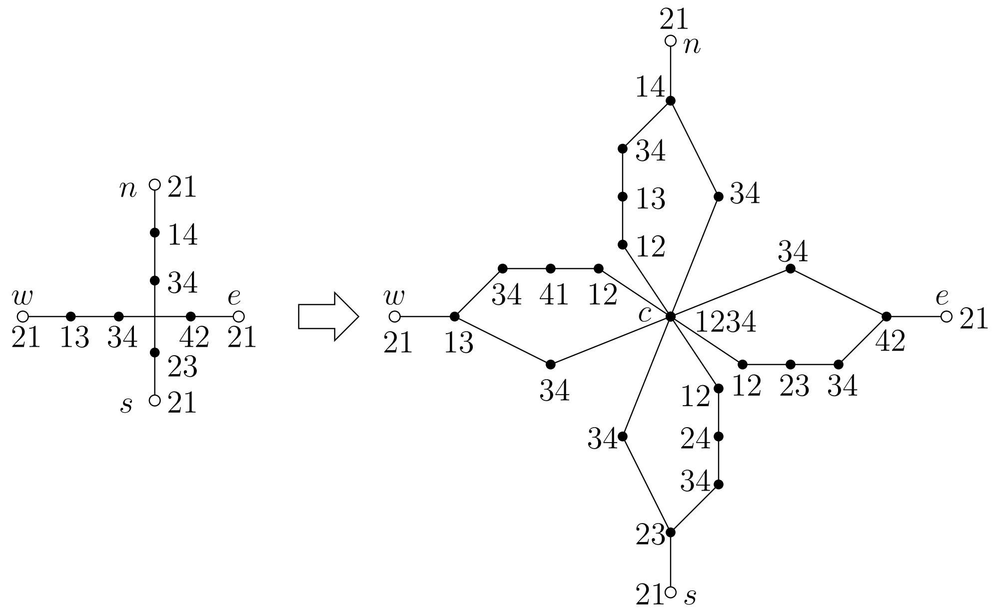

Figure 5: A crossing component corresponding to two $(1,2)$-forbidding paths.

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 the crossing component of Figure 5: this shows how an (n,s)-path and a (w,e)-path that are both $(1,2)$-forbidding paths are replaced. 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 a 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 exactly the same way as the two old forbidding paths.

Lemma 4.9 *The crossing component of Figure 5 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 neighbors 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. □

Observation 4.8 and Lemma 4.9 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 4.5 and Theorem 4.6 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 4.10 *The graphs G_N^P have $O(N^4)$ vertices and edges.*

Proof : We started 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. The graph G_N has $O(N^2)$ forbidding paths, so this drawing 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 in $O(N^4)$. Since G_N^P is planar, its average degree is less than six, so the number of edges is in $O(N^4)$ as well. □

We have constructed bipartite List-Colour Path instances with size $O(N^2)$ (Claim 4.7), and bipartite planar List-Colour Path instances with size in $O(N^4)$ (Claim 4.10). 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 (Theorem 4.6). Lemma 2.1 shows that these can be transformed into k-Colour Path instances without a significant size increase. This completes the proof of Theorem 1.4.

Acknowledgements

We are indebted to Moshe Vardi for initially suggesting that the decision problem k-Colour Path might be PSPACE-complete for $k \geq 4$.

References

[1] L. Cereceda, J. van den Heuvel and M. Johnson, *Connectedness of the graph of vertex-colourings*. CDAM Research Report LSE-CDAM-2005-11 (2005). Available from http://www.cdam.lse.ac.uk/Reports/reports2005.html; accepted for publication in Discrete Math.

[2] L. Cereceda, J. van den Heuvel and M. Johnson, *Mixing 3-colourings in bipartite graphs*. CDAM Research Report LSE-CDAM-2007-06 (2007). Available from http://www.cdam.lse.ac.uk/Reports/reports2007.html.

[3] L. Cereceda, J. van den Heuvel and M. Johnson, *Finding paths between 3-colourings*; in preparation.

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

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

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

[7] 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, LNCS **4051**, 346–357. Available from `http://arxiv.org/abs/cs.CC/0609072`.

[8] 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.

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

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