Finding Shortest Paths between Graph Colourings *

Matthew Johnson[1], Dieter Kratsch[2], Stefan Kratsch[3], Viresh Patel[4], and Daniël Paulusma[1]

[1] School of Engineering and Computing Sciences, Durham University, United Kingdom, {matthew.johnson2,daniel.paulusma}@durham.ac.uk
[2] Université de Lorraine, France, dieter.kratsch@univ-lorraine.fr
[3] Technische Universität Berlin, Germany, stefan.kratsch@tu-berlin.de
[4] Queen Mary, University of London, United Kingdom, viresh.patel@qmul.ac.uk

Abstract. The k-colouring reconfiguration problem asks whether, for a given graph G, two proper k-colourings α and β of G, and a positive integer ℓ, there exists a sequence of at most ℓ proper k-colourings of G which starts with α and ends with β and where successive colourings in the sequence differ on exactly one vertex of G. We give a complete picture of the parameterized complexity of the k-colouring reconfiguration problem for each fixed k when parameterized by ℓ. First we show that the k-colouring reconfiguration problem is polynomial-time solvable for $k = 3$, settling an open problem of Cereceda, van den Heuvel and Johnson. Then, for all $k \geq 4$, we show that the k-colouring reconfiguration problem, when parameterized by ℓ, is fixed-parameter tractable (addressing a question of Mouawad, Nishimura, Raman, Simjour and Suzuki) but that it has no polynomial kernel unless the polynomial hierarchy collapses.

1 Introduction

Graph colouring has its origin in a nineteenth century map colouring problem and has now been an active area of research for more than 150 years, finding many applications within and beyond Computer Science and Mathematics. Given a graph $G = (V, E)$ and a positive integer k, a *k-colouring* of G is a map $c\colon V \to \{1, \ldots, k\}$; it is *proper* if $c(u) \neq c(v)$ for all u, v with $uv \in E$. The problem of deciding whether a graph has a proper k-colouring for fixed $k \geq 3$ was an early example of an NP-complete problem. If, however, one knows that a graph has a proper k-colouring, or several of them, one may wish to know more about them such as how many there are or what structural properties they have.

One way to study these questions is to consider the k-colouring reconfiguration graph: given a graph G, the k-colouring reconfiguration graph $R_k(G)$ of G is a graph whose vertices are the proper k-colourings of G and where an edge is present between two k-colourings if and only if the two k-colourings differ on only a single vertex of G.

* Supported by EPSRC (EP/G043434/1), by a Scheme 7 grant from the London Mathematical Society, and by the German Research Foundation (KR 4286/1).

There are several algorithmic questions one can ask about the graph $R_k(G)$ such as whether $R_k(G)$ is connected, whether there exists a path between two given vertices of $R_k(G)$, or how long is the shortest path between two given vertices of $R_k(G)$. (Note that in general $R_k(G)$ has size exponential in the size of G, making these questions highly non-trivial.) It is the latter question, stated formally below, that we address in this paper.

k-Colouring Reconfiguration

Instance :	An n-vertex graph $G = (V, E)$, two proper k-colourings α and β and a positive integer ℓ.
Question :	Is there a path in the reconfiguration graph of G between α and β of length at most ℓ?

General Motivation. Reconfiguration graphs can be defined for any search problem: the vertices correspond to all solutions to the problem and the edges are defined by a symmetric adjacency relation normally chosen to represent a smallest possible change between solutions. They arise naturally when one wishes to understand the solution space for a search problem.

There has been much research over the last ten years on the structure and algorithmic aspects of reconfiguration graphs, not only for k-Colouring [1, 2, 5, 8–10] but also for many other problems, such as Satisfiability [11], Independent Set [7, 17], List Edge Colouring [13, 15], $L(2, 1)$-Labeling [14], Shortest Path [3, 4, 18], and Subset Sum [16]. From these studies, the following subtle phenomenon has been observed, which one would like to better understand: it is often (but not always) the case that NP-complete search problems give rise to PSPACE-complete reconfiguration problems, whereas polynomial-time solvable search problems often give rise to polynomial-time solvable reconfiguration problems. For further background we refer the reader to the recent survey of van den Heuvel [12].

Reconfiguration graphs are also important for constructing and analyzing algorithms that sample or count solutions to a search problem. Indeed, understanding connectivity properties of the k-colouring reconfiguration graph is fundamental in analyzing certain randomized algorithms for sampling and counting k-colourings of a graph and in analyzing certain cases of the Glauber dynamics in statistical physics (see Section 5 of [12]).

Our Results. Our first result, which we prove in Section 2, shows that k-Colouring Reconfiguration can be solved in polynomial time when $k = 3$, which settles a problem raised by Cereceda, van den Heuvel and Johnson [10]. Note that the cases $k = 1, 2$ are easily seen to be polynomial-time solvable.

In [10], Cereceda et al. were mainly concerned with determining whether, given a graph G and two proper 3-colourings α and β, there exists *any* path between α and β in $R_k(G)$. They found a polynomial-time algorithm to solve this problem and further showed that, for certain instances, their algorithm in fact finds a shortest path between α and β (a precise statement is given in Section 2). Here we complete their result by giving an algorithm for all instances.

Theorem 1. 3-Colouring Reconfiguration *can be solved in time $O(n^2)$.*

For $k \geq 4$, we cannot expect a polynomial-time algorithm for k-Colouring Reconfiguration: Bonsma and Cereceda [5] showed that, for each $k \geq 4$, the problem of determining if there is *any* path between two given proper k-colourings of a given graph is PSPACE-complete. On the other hand, our second result (proven in Section 3) is that for each $k \geq 4$, k-Colouring Reconfiguration is fixed-parameter tractable when parameterized by the path length ℓ.

Recall that, informally, a parameterized problem is a decision problem (in our case k-Colouring Reconfiguration) in which every problem instance I has an associated integer parameter p (in our case the path length ℓ). A parameterized problem is *fixed-parameter tractable* (FPT) if every instance I can be solved in time $f(p)|I|^c$ where f is a computable function that only depends on p and c is a constant independent of p.

Theorem 2. *For each fixed $k \geq 4$, k-Colouring Reconfiguration can be solved in time $O((k{\cdot}\ell)^{\ell^2+\ell}{\cdot}\ell n^2)$. In particular, for each fixed $k \geq 4$, k-Colouring Reconfiguration is* FPT *when parameterized by ℓ.*

Once a problem is shown to be FPT (and it is unlikely that the problem is polynomial-time solvable), one can go further and ask whether it has a *polynomial kernel*. It is well known that a problem is FPT with respect to a parameter p if and only if it can be *kernelized*, i.e., if and only if, for any instance (I,p) of the given parameterized problem, it is possible to compute in polynomial time an *equivalent instance* (I',p') such that $|I'|, p' \leq g(p)$ for some computable function g (two problem instances are equivalent if and only if they are both yes-instances or both no-instances). If $g(p)$ is a polynomial, then the given parameterized problem is said to have a polynomial kernel. We prove the following theorem in Section 4.

Theorem 3. *For each fixed $k \geq 4$, k-Colouring Reconfiguration parameterized by ℓ does not admit a polynomial kernel unless* NP $\subseteq$ coNP/poly.

In fact Theorem 3 holds even when we restrict attention to inputs where the two proper k-colourings of the input graph differ in only two vertices (note that the problem becomes trivial if the two given k-colourings differ in only one vertex).

Our three results give a *complete* picture of the parameterized complexity of k-Colouring Reconfiguration for each fixed k when parameterized by ℓ.

Related work. Fixed-parameter tractability of k-Colouring Reconfiguration was proved independently in recent work of Bonsma and Mouawad [6]. They also prove various hardness results for other parameterizations of k-Colouring Reconfiguration. In particular, they proved that if k is part of the input then k-Colouring Reconfiguration is W[1]-hard when parameterized only by ℓ (note that the problem, when parameterized only by k, is para-PSPACE-complete due to the aforementioned result of Bonsma and Cereceda [5]).

Mouawad, Nishimura, Raman, Simjour and Suzuki [20] were the first to consider reconfiguration problems in the context of parameterized complexity. For various NP-complete search problems, they showed that determining whether there exists a path of length at most ℓ in the reconfiguration graph between two given vertices is W[1]-hard (when ℓ is the parameter); they asked if there exists an NP-complete problem for which the corresponding reconfiguration problem, parameterized by ℓ, is FPT. Theorem 2 and [6] give the second positive answer to this question, the first being an FPT algorithm for a reconfiguration problem related to Vertex Cover [19]. However, perhaps surprisingly, Theorem 1 shows that there even exists an NP-complete problem for which the corresponding shortest path problem in the reconfiguration graph is polynomial-time solvable, and thus trivially FPT when parameterized by ℓ.

As mentioned earlier, deciding whether there exists *any* path in $R_k(G)$ between two k-colourings α and β of an input graph G is polynomial-time solvable for $k \leq 3$ [10] and PSPACE-complete for $k \geq 4$ [5]. The problem remains PSPACE-complete for bipartite graphs when $k \geq 4$, for planar graphs when $4 \leq k \leq 6$ and for planar bipartite graphs for $k = 4$ [5].

The algorithmic question of whether $R_k(G)$ is connected for a given G is addressed in [8, 9], where it is shown that the problem is coNP-complete for $k = 3$ and bipartite G, but polynomial-time solvable for planar bipartite G.

Finally, the study of the diameter of $R_k(G)$ raises interesting questions. In [10] it is shown that every component of $R_3(G)$ has diameter polynomial (in fact quadratic) in the size of G. On the other hand, for $k \geq 4$, explicit constructions [5] are given of graphs G for which $R_k(G)$ has at least one component with diameter exponential in the size of G. It is known that if G is a $(k-2)$-degenerate graph then $R_k(G)$ is connected and it is conjectured that in this case $R_k(G)$ has diameter polynomial in the size of G [8]; for graphs of treewidth $k-2$ the conjecture has been proved in the affirmative [1].

2 A Polynomial-Time Algorithm for $k = 3$

In this section we consider 3-Colouring Reconfiguration and prove Theorem 1. Some proofs are omitted for reasons of space.

First some definitions needed throughout the paper. Let $G = (V, E)$ be a graph on n vertices, and let α and β be two proper k-colourings of G. For *any* two colourings c and d, we say that c and d *agree* on a vertex u if $c(u) = d(u)$ and that otherwise they *disagree* on u. An $(\alpha \to \beta)$*-recolouring* R of *length* $\ell = |R|$ is a sequence of proper colourings $c_0, \ldots, c_\ell$ where $c_0 = \alpha$ and $c_\ell = \beta$, and, for $1 \leq q \leq \ell$, c_q and c_{q-1} disagree on at most one vertex. So possibly $c_q = c_{q-1}$ though in this case c_q could be deleted and the sequence that remained would also be an $(\alpha \to \beta)$-recolouring. The set $\{c_{q-1}c_q : c_{q-1} \neq c_q\}$ is a set of edges in the reconfiguration graph corresponding to a walk from α to β.

In this section, α and β are 3-colourings. The three colours are 1, 2 and 3, and we think of them cyclically: so when, for example, we refer to a colour one greater than a we mean $a + 1 \bmod 3$. A cycle in G is *fixed* with respect to a 3-colouring if the two neighbours of each vertex on the cycle are not coloured alike (one can see that this implies that the cycle is coloured in this way in every other colouring in the same component of $R_3(G)$ since one cannot change the colour of just one vertex and obtain another proper 3-colouring).

Cereceda et al. [10] provided a partial solution to 3-Colouring Reconfiguration. They were interested in recognizing whether or not α and β belong to the same component of the reconfiguration graph. They introduced a polynomial-time algorithm that we will call FindPath(G, α, β) that

- correctly determines when α and β belong to different components of $R_3(G)$;
- finds an $(\alpha \to \beta)$-recolouring of G, of length $O(n^2)$, when α and β belong to the same component of $R_3(G)$;
- moreover, if G contains a fixed cycle with respect to α, the $(\alpha \to \beta)$-recolouring found is the shortest possible.

We also note that it is possible to recognize in time $O(n^2)$ whether or not there is a fixed cycle (this is described in [10], but is an easy exercise). We need to show how to find a shortest possible $(\alpha \to \beta)$-recolouring of G in the case where α and β are known to belong to the same component of G, and G contains no fixed cycle with respect to α. We assume now that these conditions hold.

We require a further notion related to colourings called a *height* function (that extends a concept introduced in [10]). Let $S = c_0, c_1, \dots$ be a sequence of colourings where c_i and c_{i-1} disagree on exactly one vertex and $c_0 = \alpha$. The height function is denoted h^S and has domain $S \times V$ and its range is the set of integers. For each $v \in V$, $h^S(c_0, v) = 0$. For $i > 0$, for each $v \in V$:

$$h^S(c_i, v) = \begin{cases} h^S(c_{i-1}, v), & \text{if } c_i(v) = c_{i-1}(v); \\ h^S(c_{i-1}, v) + 2, & \text{if } c_i(v) \equiv c_{i-1}(v) + 1 \bmod 3; \\ h^S(c_{i-1}, v) - 2, & \text{if } c_i(v) \equiv c_{i-1}(v) - 1 \bmod 3. \end{cases}$$

So each vertex has height 0 initially and is raised or lowered by 2 when its colour is increased or decreased as we move along the sequence of colourings. For any $(\alpha \to \beta)$-recolouring R, let the *total height* of R be $H(R) = \sum_{v \in V} |h^R(\beta, v)|$.

Lemma 1. *Let R be a $(\alpha \to \beta)$-recolouring of length ℓ. Then $\ell \geq \frac{1}{2} H(R)$.*

Proof. For each colouring in R, the height of only one vertex differs from the previous colouring in R and the height difference is 2. Thus, for each vertex v, at least $|h^R(\beta, v)|/2$ distinct colourings in R are needed and the lemma follows. □

Lemma 2. *For any colouring c, for any sequence of colourings S from α to c, for each vertex v in V,*

$$2(c(v) - \alpha(v)) \equiv h^S(c, v) \bmod 6 \tag{1}$$

Proof. We use induction on the length of S. If S contains only one colouring, then this is α, and both sides of (1) are zero with $c = \alpha$.

Suppose that S is longer and that c' is its penultimate colouring. We must show that if (1) is true for c', then it is also true for c. If c and c' agree on v, then we are done. If c and c' disagree on v, then we need only to notice that

$$2(c(v) - c'(v)) \equiv h^S(c, v) - h^S(c', v) \bmod 6$$

and each side of (1) changes by the same amount if we replace c' by c. □

Some more terminology. If an edge is oriented, then we can define its *weight* with respect to a colouring c. The *weight* of an edge oriented from u to v is a value $w(c, \overrightarrow{uv}) \in \{-1, 1\}$ such that $w(c, \overrightarrow{uv}) \equiv c(v) - c(u) \bmod 3$. To orient a path is to orient each edge so that a directed path is obtained. The weight of an oriented path $w(c, \overrightarrow{P})$ is the sum of the weight of its edges.

Lemma 3. *For any colouring c, for any sequence of colourings S from α to c, for each pair of vertices u, v in V, for each oriented path $\overrightarrow{P}$ from u to v,*

$$h^S(c, u) = h^S(c, v) + w(c, \overrightarrow{P}) - w(\alpha, \overrightarrow{P}). \tag{2}$$

Proof. We use induction on the length of S. If S contains only one colouring, then this is α, and both sides of (2) are zero with $c = \alpha$.

Suppose that S is longer and that c' is the penultimate colouring in the sequence. We must show that if (2) is true for c', then it is also true for c. Let x be the vertex on which c' and c disagree.

Suppose that $x \notin \{u, v\}$. If $\overrightarrow{P}$ does not contain x, then clearly the weight of the path is the same for c' and c. If $\overrightarrow{P}$ does contain x, then let $\overrightarrow{yx}$ and $\overrightarrow{xz}$ be the edges of $\overrightarrow{P}$ that x belongs to. As c and c' are proper and $c(x) \neq c'(x)$, we must have $c(y) = c'(y) = c'(z) = c(z)$. Thus

$$\begin{aligned} w(c, \overrightarrow{yx}) + w(c, \overrightarrow{xz}) &= c(x) - c(y) + c(z) - c(x) = 0, \\ w(c', \overrightarrow{yx}) + w(c', \overrightarrow{xz}) &= c'(x) - c'(y) + c'(z) - c'(x) = 0. \end{aligned}$$

So $w(c, \overrightarrow{P}) = w(c', \overrightarrow{P})$ and both sides of (2) are unchanged when c' replaces c.

Suppose that $x = u$. Let y be the vertex adjacent to x on $\overrightarrow{P}$. Suppose that $h^S(c, x) = h^S(c', x) + 2$; that is, the colour of x is increased (as c replaces c'). Then $c(x) \equiv c'(x) + 1 \bmod 3$ and so $c(y) \equiv c'(x) - 1 \bmod 3$. Thus $w(c', \overrightarrow{xy}) = -1$, and, as $c(y) \equiv c(x) + 1 \bmod 3$, $w(c, \overrightarrow{xy}) = 1$, which gives $w(c, \overrightarrow{P}) = w(c', \overrightarrow{P}) + 2$ and (2) remains satisfied. If the height of x is instead lowered, a similar argument can be used. The case $x = v$ can also be proved in this way. $\square$

If β is obtained from α by an $(\alpha \to \beta)$-recolouring, then the vertices can be ordered by their heights. Lemma 3 tells us that this ordering is the same for all $(\alpha \to \beta)$-recolourings and can be found by considering only α, β and paths in G. Let y be the vertex that is a median vertex in this ordering (if $|V|$ is even, arbitrarily choose one of the two vertices in the middle of the ordering). Let g be a function defined on V such that for all $v \in V$

$$g(v) = w(\beta, \overrightarrow{P_{vy}}) - w(\alpha, \overrightarrow{P_{vy}}).$$

Considering Lemma 3, we see that $g(v)$ is the height of v relative to y with respect to β, and that ordering the vertices by g is equivalent to ordering them by height so y is also a median of this ordering.

For any integer k congruent to $2(\beta(y) - \alpha(y)) \bmod 6$, let

$$J(k) = \sum_{v \in V} |k + g(v)|.$$

We observe that if k is the height of y, then $J(k)$ is the sum of the vertices' heights. Let (k_1, k_2) be the unique pair in the set $\{(0,0),(2,-4),(4,-2)\}$ such that $k_1 \equiv k_2 \equiv 2(\beta(y) - \alpha(y)) \bmod 6$. (Notice that, by Lemma 2, k_1 and k_2 are two possible values for the height of y when β is obtained by a recolouring sequence.)

Lemma 4. *Let $k \equiv 2(\beta(y) - \alpha(y)) \bmod 6$ be an integer. Then $J(k)$ is at least $\min\{J(k_1), J(k_2)\}$, and for any $(\alpha \to \beta)$-recolouring R, $|R| \geq \frac{1}{2}\min\{J(k_1), J(k_2)\}$.*

Lemma 5. *Let $k \equiv 2(\beta(y) - \alpha(y)) \bmod 6$ be an integer. If S is a recolouring sequence from α to c such that, for all $v \in V$, $h^S(c, v) = k + g(v)$, then $c = \beta$.*

Lemma 6. *Let $k \equiv 2(\beta(y) - \alpha(y)) \bmod 6$ be an integer. Then there exists an $(\alpha \to \beta)$-recolouring R of length ℓ such that $\ell = \frac{1}{2}J(k)$.*

Proof. We will define R by describing how to recolour from α to a colouring c such that $h^R(c, v) = k + g(v)$. By Lemma 5, $c = \beta$. Let $h(v)$ denote $k + g(v)$.

As we go from one colouring to the next we change the height of one vertex v by 2. If this change is always such that the difference between the current height of v and $k + g(v)$ is reduced by 2, then we will have $\ell = \frac{1}{2}J(k)$.

More definitions: for a vertex u in G and colouring c, a *maximal rising path* from u is a path on vertices $u = v_0, v_1, \ldots v_t$ such that, for $1 \leq i \leq t$, $c(v_i) \equiv c(v_{i-1}) + 1 \bmod 3$, and v_t has no neighbours coloured $c(v_t) + 1 \bmod 3$. A maximal rising path can easily be found: we just repeatedly look for the next vertex along and if none with the required colour can be found we are done; we never return to a vertex that we have already met as this would mean we had found a fixed cycle. A *maximal falling path* from u is the same except that the colours decrease rather than increase moving along the path from u, and one can be found in an analogous way. (That is, the colours along a rising path are, for example, $231231231231\cdots$, and along a falling path are, for example, $321321321321\cdots$)

We need to describe how, at each step, to choose a vertex v to recolour and say what its "new" colour should be. Let c denote the current colouring and S the sequence of colourings found so far (so $h^S(c, x)$ is the current height of a vertex x).

1. Find a vertex x for which $|h(x) - h^S(c, x)|$ is maximum.
2. If $h(x) - h^S(c, x) > 0$, find a maximal rising path from x. Else find a maximal falling path from x. In either case, let v be the end-vertex of the path.
3. Change the colour of v so that $|h(v) - h^S(c, v)|$ is reduced by 2.

We must show that $h(v) \neq h^S(c, v)$ and that the new colouring is proper. We will treat the case that $h(x) - h^S(c, x) > 0$ (the other case is identical in form).

Let p be the number of edges in the maximal rising path P from x to v. Let $\overrightarrow{P}$ be the orientation from x to v. Applying Lemma 3 twice to x and v and then subtracting, we find that

$$\begin{aligned} h^S(c, x) &= h^S(c, v) + w(c, \overrightarrow{P}) - w(\alpha, \overrightarrow{P}), \\ h(x) &= h(v) + w(\beta, \overrightarrow{P}) - w(\alpha, \overrightarrow{P}) \\ h(x) - h^S(c, x) &= h(v) - h^S(c, v) + w(\beta, \overrightarrow{P}) - w(c, \overrightarrow{P}). \end{aligned}$$

Note that $w(c, \overrightarrow{P}) = p$ and that $w(\beta, \overrightarrow{P}) \leq p$ since the weight of a path cannot be more than the number of edges. Thus $0 < h(x) - h^S(c, x) \leq h(v) - h^S(c, v)$ and so $h(v) > h^S(c, v)$. As reducing $|h(v) - h^S(c, v)|$ requires increasing the colour at v by 1, and it is at the end of a maximal rising path, the new colouring is proper. $\square$

Proof of Theorem 1. The algorithm FINDPATH(G, α, β) can be used to determine whether there is a path from α to β of length at most ℓ except when α and β are in the same component of $R_3(G)$ and G contains no fixed cycles with respect to α. In this case, a path of length ℓ can be found if and only if $\ell \leq \frac{1}{2} \min\{J(k_1), J(k_2)\}$. This follows from Lemmas 4 and 6.

Though the running time of FINDPATH is not analyzed in detail in [10], it is easy to prove that it is $O(n^2)$. We omit the details, but it is also straightforward to show that $J(k_1)$ and $J(k_2)$ can be found in time $O(n^2)$. Moreover, if one wishes to find the path from α to β this can be done by using the algorithm in the proof of Lemma 6 which can also be adapted to run in time $O(n^2)$. $\square$

3 An FPT Algorithm for *k*-Colouring Reconfiguration

In this section we will present our FPT algorithm for k-COLOURING RECONFIGURATION when parameterized by ℓ. Let $G = (V, E)$ be a graph on n vertices, and let α, β be two proper k-colourings of G. First we prove three lemmas concerning the vertices that might be recoloured if a path between α and β of length at most ℓ does exist. That is, we assume that (G, α, β, ℓ) is a yes-instance of k-COLOURING RECONFIGURATION. This means that there exists an $(\alpha \rightarrow \beta)$-recolouring $R = c_0, \dots, c_\ell$. We assume that R has *minimum length.*

We say that R *recolours* a vertex u if $c_q(u) \neq \alpha(u)$ for some q. Notice that if for each recoloured vertex u we find the least q such that $c_q(u) \neq \alpha(u)$, these values must be distinct (else c_q and c_{q-1} disagree on more than one vertex). Thus the number of distinct vertices recoloured by R is at most ℓ. We will prove something stronger. For $0 \leq q \leq \ell$, let W_q be the set of vertices on which c_0 and c_q disagree, that is, $W_q = \{u \in V : c_0(u) \neq c_q(u)\}$.

Lemma 7. *For all q with $1 \leq q \leq \ell$, the set W_q has size $|W_q| \leq q$.*

Proof. Suppose this is false and let r be the smallest value such that $|W_r| > r$. So $|W_{r-1}| \leq r-1$ (clearly $r-1 \geq 0$ as W_0 is the empty set). Then there are (at least) two vertices v_1, v_2 in $W_r \setminus W_{r-1}$, and so, for $i \in \{1, 2\}$, $c_{r-1}(v_i) = c_0(v_i) \neq c_r(v_i)$, and c_r and c_{r-1} disagree on more than one vertex; a contradiction. $\square$

For any $u \in V$, let $N(u)$ be the set of neighbours of u. For any $v \in N(u)$, let $N(u, v) = \{w \in N(u) : \alpha(w) = \alpha(v)\}$; that is, the set of neighbours of u with the same colour as v in α. Let $A_0 = \{v \in V : \alpha(v) \neq \beta(v)\}$ be the set of vertices on which α and β disagree. For $i \geq 1$, let $A_i = \bigcup_{u \in A_{i-1}} \{v \in N(u) : |N(u, v)| \leq \ell\}$. That is, to find A_i consider each vertex u in A_{i-1} and partition $N(u)$ into colour classes (according to the colouring α). Vertices in $N(u)$ that belong to colour classes of size at most ℓ belong to A_i. Note that two sets A_h and A_i need not be disjoint. Our first goal is to show that each vertex recoloured by R must be in $A^* = \bigcup_{h=0}^{\ell-1} A_h$. We will then show that the size of A^* is bounded by a function of $k + \ell$. This will then enable us to use brute-force to find R or some other $(\alpha \to \beta)$-recolouring of G (if it exists).

Lemma 8. *Each vertex recoloured by R belongs to A^*.*

Proof. For $i \geq 0$, let $L_i = A_i \setminus (\bigcup_{h<i} A_j)$ be the set of vertices that are in A_i but not in any A_h with $h < i$. Let z be the greatest value such that R recolours a vertex in L_z; denote this vertex by v_z. By definition, every vertex in A_0 is recoloured by R. Let $v_0 \in A_0$. We claim that also for $1 \leq i \leq z-1$, there is a vertex $v_i \in L_i$ that is recoloured by R. Then, as $v_0, \ldots, v_z$ are distinct vertices and R has length ℓ, we have $z \leq \ell - 1$ proving the lemma. For contradiction, assume there is a set L_i $(1 \leq i \leq z-1)$ that contains no vertex recoloured by R.

From R we construct a new recolouring sequence R' by ignoring every recolouring step done to a vertex in $V \setminus \bigcup_{h<i} L_h$. For $0 \leq q \leq \ell$, let d_q be a colouring of G such that

- if $u \in \bigcup_{h<i} L_h$, $d_q(u) = c_q(u)$;
- if $u \notin \bigcup_{h<i} L_h$, $d_q(u) = \alpha(u)$.

Let R' be the sequence $d_0, \ldots, d_\ell$. Note that $d_0 = \alpha$, as $d_0(u)$ is either $c_0(u)$ or $\alpha(u)$, and $c_0 = \alpha$. Moreover, if $u \in \bigcup_{h<i} L_h = \bigcup_{h<i} A_i$ then $d_\ell(u) = c_\ell(u) = \beta(u)$, and if $u \notin \bigcup_{h<i} L_h$ then $d_\ell(u) = \alpha(u) = \beta(u)$ (since α and β only disagree on vertices in A_0); thus $d_\ell = \beta$. This means that if we can show that $d_1, \ldots, d_{\ell-1}$ are proper colourings, then R' is an $(\alpha \to \beta)$-recolouring. We will prove this first.

Assume to the contrary that R' contains a colouring d_q that is not proper. Then there is an edge uv with $d_q(u) = d_q(v)$. If u and v both belong to $\bigcup_{h<i} L_h$ then $c_q(u) = c_q(v)$, and if neither belong to $\bigcup_{h<i} L_h$ then $\alpha(u) = \alpha(v)$. Both cases are not possible, as c_q and α are proper colourings. Hence we may assume, without loss of generality, that $u \in \bigcup_{h<i} L_h$ and $v \notin \bigcup_{h<i} L_h$. Then $c_q(u) = d_q(u) = d_q(v) = \alpha(v)$ by the definition of d_q.

As $v \in N(u)$, the set $N(u, v)$ exists. First suppose $|N(u, v)| \leq \ell$. Then $v \in A_i$ by the definition of A_i. Hence $v \in L_h$ for some $h \leq i$. As $v \notin \bigcup_{h<i} L_h$, we obtain $v \in L_i$. By assumption, no vertex of L_i is recoloured by R. Hence $c_q(v) = \alpha(v)$ and thus $c_q(u) = c_q(v)$ contradicting the fact that c_q is a proper k-colouring.

Now suppose $|N(u, v)| > \ell$. Because $c_q(u) = \alpha(v)$ and c_q is proper, we find that $c_q(w) \neq c_q(u) = \alpha(v) = \alpha(w)$ for all $w \in N(u, v)$. Thus $W_q \supseteq N(u, v)$ and so $|W_q| \geq |N(u, v)| > \ell \geq q$ contradicting the fact that $|W(q)| \leq q$ by Lemma 7. So, d_q must be proper. We conclude that R' is an $(\alpha \to \beta)$-recolouring of length ℓ.

We now proceed as follows. Recall that $v_z \in L_z$. Then there is a pair of colourings c_q and c_{q+1} that differ only on v_z. Because $v_z \in L_z$, $v_z \notin \bigcup_{h<i} L_h$. Hence, d_q and d_{q+1} are identical colourings. We remove d_q from R' to obtain another $(\alpha \to \beta)$-recolouring, which has length shorter than ℓ, contradicting that R has minimum length. This completes the proof. □

Lemma 9 gives a bound on $|A^*|$ depending only on k and ℓ (proof omitted).

Lemma 9. *The set A^* has size $|A^*| \leq \ell \cdot (k\ell)^\ell$.*

We are now ready to present our FPT algorithm and prove Theorem 2.

Proof of Theorem 2. Let $k \geq 1$, and let (G, α, β, ℓ) be an instance of k-Colouring Reconfiguration, where G is a graph on n vertices, and α, β are two proper k-colourings of G. Our algorithm does as follows. First compute the set A^* in $O(n^2)$ time. By Lemma 9, we find that $|A^*| \leq \ell \cdot (k\ell)^\ell$. By Lemma 8, we only have to search for a path of length at most ℓ in $R_k(G)$ among the vertices of A^*. By allowing consecutive recolourings to be equal we may restrict our search to $(\alpha \to \beta)$-recolourings of length exactly ℓ. Use brute force to enumerate all possible sequences of pairs (v_i, c_i), such that for all $0 \leq i \leq \ell - 1$, v_i is a vertex in A^* and c_i is a colour in $\{1, \ldots, k\}$. For each such sequence do as follows. Starting from α, recolour v_i with colour c_i for $i = 0, \ldots, \ell - 1$. As soon as this results in a k-colouring that is not proper, stop considering the sequence. If not, check whether the resulting colouring is equal to β. If this happens, then there is a path of length ℓ in $R_k(G)$. Hence, return `yes`. Otherwise, that is, if no sequence has this property, return `no`. Processing one sequence takes time $O(\ell n^2)$. By using Lemma 9, the number of sequences is at most $(|A^*| \cdot k)^\ell \leq ((\ell \cdot (k \cdot \ell)^\ell) \cdot k)^\ell \leq (k \cdot \ell)^{\ell^2 + \ell}$, leading to a total running time of $O((k \cdot \ell)^{\ell^2 + \ell} \cdot \ell n^2)$. This completes the proof. □

4 A Lower Bound for Kernelization for $k \geq 4$

In this section we sketch the proof of Theorem 3, which states that k-Colouring Reconfiguration parameterized by the maximum path length ℓ does not admit a polynomial kernelization for $k \geq 4$ unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$. Theorem 3 is proved by a polynomial parameter transformation from the Hitting Set problem parameterized by the number m of sets in the input. It is known that this rules out polynomial kernels for the target problem, unless $\mathsf{NP} \subseteq \mathsf{coNP/poly}$.

The main idea for the reduction is to create a 4-coloured tree that serves as a selection gadget for each set, which requires a recolouring at its root. This in turn requires a chain of earlier recolourings starting in one of the leaves; the selection of possible leaves encodes the elements of the set. Finally, recolouring any leaf requires a recolouring in a set of vertices corresponding to the ground set; this encodes the selection of a hitting set. Crucially, the height of the tree construction, which factors into the number ℓ of needed recolourings, can be bounded polynomially in the input parameter m.

The selection trees are composed of claws on four vertices a, b, c, d each, where c is the center vertex. For each of these vertices, α and β colour are the same, but we may (through adjacent gadgets) require a recolouring of d. The latter will be only possible by first recolouring a or b. To ensure this, several colours will be forbidden for a, b, c, d by adjacency to a global k-clique:

1. For a we have $\alpha(a) = \beta(a) = 2$, and, using adjacency to the k-clique, only colours 2 and 4 allow proper k-colourings.
2. For b we have $\alpha(b) = \beta(b) = 3$, and only colours 3 and 4 are possible.

3. For c we have $\alpha(c) = \beta(c) = 1$, and only colours 1, 2, and 3 are possible.
4. For d we have $\alpha(d) = \beta(d) = 4$, and only colours 1 and 4 are possible.

If we need to recolour d then it can only change to colour 1. This requires to first recolour c to either 2 or 3. This in turn, depending on choice of colour 2 or 3, necessitates a recolouring of a to 4 or b to 4. Thus, locally, we make a choice out of two options using constant number of recolourings. By building a tree structure from such claws, always making d-vertices of new claws adjacent to the a- or b-vertex of the current claw, we can make a one out of n choice at cost of $O(\log n)$ recolourings.

By standard arguments when reducing from a Hitting Set instance with m sets (recall that m is the parameter) we have a ground set size of $n \leq 2^m$. Thus, the choice of element to hit in each set costs only $O(\log n) = O(m)$ recolourings per set. To relate the different choices we make a set of n vertices that are adjacent to the corresponding leaves in each selection gadget. If we end up with a recolouring in a leaf of a selection gadget then this requires a recolouring of the corresponding one among these n vertices. By correct choice of number of recolourings and detailed analysis, we can enforce that at most p out of n vertices can be recoloured. Note that this involves also recolouring almost all vertices back to their initial colour since α and β will agree on almost all vertices (which is necessary to make the graph exponentially large in the parameter value). The whole recolouring from α to β is then possible within the chosen number of steps if and only if the given set family has a hitting set of size at most p.

5 Conclusions

We showed that k-Colouring Reconfiguration is fixed-parameter tractable for any fixed $k \geq 1$, when parameterized by the number of recolourings ℓ. It is a natural question to ask whether a single-exponential FPT algorithm can be achieved for this problem. We also proved that the k-Colouring Reconfiguration problem is polynomial-time solvable for $k = 3$, which solves the open problem of Cereceda et al. [10], and that it has no polynomial kernel for all $k \geq 4$, when parameterized by ℓ (up to the standard assumption that $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$).

Acknowledgements. We are grateful to several reviewers for insightful comments that greatly improved our presentation.

References

1. M. Bonamy, N. Bousquet, Recoloring bounded treewidth graphs, Electronic Notes in Discrete Mathematics 44, 257–262 (2013).
2. M. Bonamy, M. Johnson, I.M. Lignos, V. Patel, D. Paulusma, Reconfiguration graphs for vertex colourings of chordal and chordal bipartite graphs, Journal of Combinatorial Optimization 27, 132–143 (2014).
3. P. Bonsma, The complexity of rerouting shortest paths, In B. Rovan, V. Sassone, P. Widmayer (eds.) Mathematical Foundations of Computer Science (MFCS 2012). LNCS, vol. 7464, pp. 222–233. Springer Berlin Heidelberg (2012).

4. P. Bonsma, Rerouting shortest paths in planar graphs, In D. D'Souza, T. Kavitha, J. Radhakrishnan (eds.) IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012). LIPIcs, vol. 18, pp. 337–349. Schloss Dagstuhl–Leibniz-Zentrum für Informatik (2012).
5. P. Bonsma, L. Cereceda, Finding paths between graph colourings: PSPACE-completeness and superpolynomial distances, Theoretical Computer Science, 410, 5215–5226 (2009).
6. P. Bonsma, A. E. Mouawad, The complexity of bounded length graph recolouring, Manuscript (2014) arXiv:1404.0337.
7. P. Bonsma, M. Kamiński, M. Wrochna, Reconfiguring independent sets in claw-free graphs, In 14th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2014), Lecture Notes in Computer Science, vol. 8503, pp. 86–97. Springer Berlin Heidelberg (2014).
8. L. Cereceda, J. van den Heuvel, M. Johnson, Connectedness of the graph of vertex-colourings, Discrete Mathematics 308, 913–919 (2008).
9. L. Cereceda, J. van den Heuvel, M. Johnson, Mixing 3-colourings in bipartite graphs, European Journal of Combinatorics 30, 1593–1606 (2009).
10. L. Cereceda, J. van den Heuvel, M. Johnson, Finding paths between 3-colourings, Journal of Graph Theory 67, 69–82 (2010).
11. P. Gopalan, P. G. Kolaitis, E. N. Maneva, C. H. Papadimitriou, The connectivity of boolean satisfiability: computational and structural dichotomies, SIAM Journal on Computing 38, 2330–2355 (2009).
12. J. van den Heuvel, The complexity of change, In S. R. Blackburn, S. Gerke, M. Wildon (eds.) Surveys in Combinatorics 2013. London Mathematical Society Lecture Note Series, vol. 409, pp. 127–160. Cambridge University Press (2013).
13. T. Ito, M. Kamiński, E. D. Demaine, Reconfiguration of list edge-colorings in a graph, In F. Dehne, M. Gavrilova, J-R. Sack, C. D. Tth, (eds.) Algorithms and Data Structures. Lecture Notes in Computer Science, vol. 5664, pp. 375–386. Springer Berlin Heidelberg (2009).
14. T. Ito, K. Kawamura, H. Ono, X. Zhou, Reconfiguration of list $L(2,1)$-labelings in a graph, In K-M. Chao, T-S. Hsu, D-T. Lee (eds.) Algorithms and Computation (ISAAC 2012). Lecture Notes in Computer Science, vol. 7676, pp. 34–43. Springer Berlin Heidelberg (2012).
15. T. Ito, K Kawamura, X. Zhou, An improved sufficient condition for reconfiguration of list edge-colorings in a tree, In M. Ogihara, J. Tarui (eds.) Theory and Applications of Models of Computation (TAMC 2011). Lecture Notes in Computer Science, vol. 6648, pp. 94–105. Springer Berlin Heidelberg (2011).
16. T. Ito, E. D. Demaine, Approximability of the subset sum reconfiguration problem, In M. Ogihara, J. Tarui (eds.) Theory and Applications of Models of Computation (TAMC 2011). Lecture Notes in Computer Science, vol. 6648, pp. 58–69. Springer Berlin Heidelberg (2011).
17. M. Kamiński, P. Medvedev, M. Milanič, Complexity of independent set reconfigurability problems, Theoretical Computer Science 439, 9–15 (2012).
18. M. Kamiński, P. Medvedev, M. Milanič, Shortest paths between shortest paths, Theoretical Computer Science 412, 5205–5210 (2011).
19. A. E. Mouawad, N. Nishimura and V. Raman, Vertex cover reconfiguration and beyond, Manuscript (2014) arXiv:1402.4926.
20. A. E. Mouawad, N. Nishimura, V. Raman, N. Simjour, A. Suzuki, On the parameterized complexity of reconfiguration problems, In G. Gutin, S. Szeider (eds.) Parameterized and Exact Computation (IPEC 2013). Lecture Notes in Computer Science, vol. 8246, pp. 281–294. Springer Berlin Heidelberg (2013).