Finding a Small Number of Colourful Components

Laurent Bulteau
Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, UPEM, ESIEE Paris, France
laurent.bulteau@u-pem.fr

Konrad K. Dabrowski
Department of Computer Science, Durham University, Durham, UK
konrad.dabrowski@durham.ac.uk

Guillaume Fertin
Université de Nantes, LS2N (UMR 6004), CNRS, Nantes, France
guillaume.fertin@univ-nantes.fr

Matthew Johnson
Department of Computer Science, Durham University, Durham, UK
matthew.johnson2@durham.ac.uk

Daniël Paulusma
Department of Computer Science, Durham University, Durham, UK
daniel.paulusma@durham.ac.uk

Stéphane Vialette
Université Paris-Est, LIGM (UMR 8049), CNRS, ENPC, UPEM, ESIEE Paris, France
stephane.vialette@u-pem.fr

Abstract

A partition $(V_1, \ldots, V_k)$ of the vertex set of a graph G with a (not necessarily proper) colouring c is colourful if no two vertices in any V_i have the same colour and every set V_i induces a connected graph. The Colourful Partition problem, introduced by Adamaszek and Popa, is to decide whether a coloured graph (G, c) has a colourful partition of size at most k. This problem is related to the Colourful Components problem, introduced by He, Liu and Zhao, which is to decide whether a graph can be modified into a graph whose connected components form a colourful partition by deleting at most p edges.

Despite the similarities in their definitions, we show that Colourful Partition and Colourful Components may have different complexities for restricted instances. We tighten known NP-hardness results for both problems by closing a number of complexity gaps. In addition, we prove new hardness and tractability results for Colourful Partition. In particular, we prove that deciding whether a coloured graph (G, c) has a colourful partition of size 2 is NP-complete for coloured planar bipartite graphs of maximum degree 3 and path-width 3, but polynomial-time solvable for coloured graphs of treewidth 2.

Rather than performing an ad hoc study, we use our classical complexity results to guide us in undertaking a thorough parameterized study of Colourful Partition. We show that this leads to suitable parameters for obtaining FPT results and moreover prove that Colourful Components and Colourful Partition may have different parameterized complexities, depending on the chosen parameter.

2012 ACM Subject Classification Mathematics of computing → Graph theory

Keywords and phrases Colourful component, colourful partition, tree, treewidth, vertex cover

Digital Object Identifier 10.4230/LIPIcs.CPM.2019.20

Related Version A full version of this paper is available from `https://arxiv.org/abs/1808.03561` [3].

Funding Supported by EPSRC (EP/K025090/1) and the Leverhulme Trust (RPG-2016-258).

1 Introduction

Research in comparative genomics, which studies the structure and evolution of genomes from different species, has motivated a number of interesting graph colouring problems. In this paper we focus on the multiple genome alignment problem, where one takes a set of sequenced genomes, lets the genes be the vertex set of a graph G, and joins by an edge any pair of genes whose *similarity* (determined by their nucleotide sequences) exceeds a given threshold. The vertices are also coloured to indicate the species to which each gene belongs. This leads to a *coloured graph* (G, c), where $c : V(G) \to \{1, 2, \ldots\}$ denotes the colouring of G. We emphasize that c is not necessarily proper (that is, adjacent vertices may have the same colour).

One seeks to better understand the evolutionary processes affecting these genomes by attempting to partition similar genes into *orthologous* sets (that is, collections of genes that originated from the same ancestral species but diverged following a speciation event). This translates into partitioning $V(G)$ such that each part

(i) contains no more than one vertex of each colour, and
(ii) induces a connected component.

These two conditions ensure that each part contains vertices representing an orthologous set of similar genes. In addition, one seeks to find a partition that is in some sense optimal.

When Zheng et al. [14] considered this model, one approach they followed was to try to delete as few edges as possible such that the connected components of the resulting graph G' are *colourful*, that is, contain no more than one vertex of any colour; in this case the connected components give the partition of $V(G)$. This led them to the Orthogonal Partition problem, introduced in [8], also known as the Colourful Components problem [1, 2, 7]. (Note that a graph is *colourful* if each of its components is colourful.)

Colourful Components	
Instance:	A coloured graph (G, c) and an integer $p \geq 1$
Question:	Is it possible to modify G into a colourful graph G' by deleting at most p edges?

The focus of this paper is on the companion problem Colourful Partition, which was introduced by Adamaszek and Popa [1]. A partition is *colourful* if every partition class induces a connected colourful graph. The *size* of a partition is its number of partition classes.

Colourful Partition	
Instance:	A coloured graph (G, c) and an integer $k \geq 1$
Question:	Does (G, c) have a colourful partition of size at most k?

Let us immediately demonstrate with an example that Colourful Components and Colourful Partition are distinct: a colourful partition with the fewest parts might require the deletion of more edges than the minimum needed to obtain a colourful graph. This has not previously been clearly established. Later, we will find a family of instances on which the problems have different complexities (see Corollary 11).

▶ **Example 1.** Let G have vertices $u_1, \ldots, u_k$, $v_1, \ldots, v_k$, w, w' and edges ww' and $u_i w$, $u_i w'$, $v_i w$, $v_i w'$ for $i \in \{1, \ldots, k\}$. Let c assign colour i to each u_i and v_i, colour $k + 1$ to w and colour $k+2$ to w'. Then $(\{u_1, \ldots, u_k, w\}, \{v_1, \ldots, v_k, w'\})$ is a colourful partition for (G, c) of size 2, which we obtain by deleting $2k+1$ edges. However, deleting the $2k$ edges $v_i w$ and $v_i w'$ for $i \in \{1, \ldots, k\}$ also yields a colourful graph (with a colourful partition of size $k + 1$). See Figure 1 for an illustration.

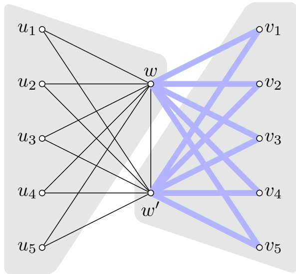

Figure 1 The graph G of Example 1 with $k = 5$.

Known Results. Adamaszek and Popa [1] proved, among other results, that Colourful Partition does not admit a polynomial-time approximation within a factor of $n^{\frac{1}{14}-\epsilon}$, for any $\epsilon > 0$ (assuming P $\neq$ NP). A coloured graph (G, c) is *ℓ-coloured* if $1 \leq c(u) \leq \ell$ for all $u \in V(G)$. Bruckner et al. [2] proved the following two results for Colourful Components. The first result follows from observing that for $\ell = 2$, the problem becomes a maximum matching problem in a bipartite graph after removing all edges between vertices coloured alike. This observation can also be used for Colourful Partition.

▶ **Theorem 2** ([2]). *Colourful Partition and Colourful Components are polynomial-time solvable for 2-coloured graphs.*

▶ **Theorem 3** ([2]). *Colourful Components is NP-complete for 3-coloured graphs of maximum degree 6.*

The situation for trees is different than for general graphs (see Example 1). A tree T has a colourful partition of size k if and only if it can be modified into a colourful graph by at most $k - 1$ edge deletions. Hence, the problems Colourful Partition and Colourful Components are equivalent for trees. The following hardness and FPT results are due to Bruckner et al. [2] and Dondi and Sikora [7]. Note that trees of diameter at most 3 are stars and double stars (the graph obtained from two stars by adding an edge between their central vertices), for which both problems are readily seen to be polynomial-time solvable. A *subdivided star* is the graph obtained by subdividing the edges of a star.

▶ **Theorem 4** ([2]). *Colourful Partition and Colourful Components are polynomial-time solvable for coloured trees of diameter at most 3, but NP-complete for coloured trees of diameter 4.*

▶ **Theorem 5** ([7]). *Colourful Partition and Colourful Components are polynomial-time solvable for coloured paths (which have path-width at most 1), but NP-complete for coloured subdivided stars (which are trees of path-width at most 2).*

▶ **Theorem 6** ([2]). *Colourful Partition and Colourful Components are FPT for coloured trees, when parameterized by the number of colours.*

▶ **Theorem 7** ([7]). *Colourful Partition and Colourful Components are FPT for coloured trees, when parameterized by the number of colourful components (or equivalently, by the size of a colourful partition).*

In addition to Theorem 6, Bruckner et al. [2] showed that Colourful Components is FPT for general coloured graphs when parameterized by the number of colours ℓ and the number of edge deletions p. Misra [10] considered the size of a minimum vertex cover as the parameter.

▶ **Theorem 8** ([10]). Colourful Components *is* FPT *when parameterized by vertex cover number.*

Our Contribution.[1] Our main focus is the Colourful Partition problem. We consider particular graph classes and, for FPT results, parameters. Our choices have a clear motivation as, guided by past work, we seek the key to understanding the problem's (in)tractability. That is, our aims are: **(i)** to improve known NP-hardness results by obtaining tight results and to generalize tractability results for Colourful Partition (Section 3); **(ii)** to prove new results analogous to those known for Colourful Components (Section 3) and to show that the two problems are different (Sections 3 and 4); and **(iii)** to use the new results to determine suitable parameters for obtaining FPT results (Section 4). Along the way we also prove some new results for Colourful Components.

First, we show an analogue of Theorem 3 by proving that Colourful Partition is NP-complete even for 3-coloured 2-connected planar graphs of maximum degree 3. The bounds on this result are best possible, as Colourful Partition is polynomial-time solvable for 2-coloured graphs (Theorem 2) and for graphs of maximum degree 2 (trivial). We show that it also gives us a family of instances on which Colourful Components and Colourful Partition have different complexities.

Second, we focus on coloured trees. Due to Theorem 6, Colourful Partition is polynomial-time solvable for ℓ-coloured trees for every fixed ℓ. Hence the coloured trees in the hardness proofs of Theorems 4 and 5 use an arbitrarily large number of colours. They also have arbitrarily large degree. We define the *colour-multiplicity* of a coloured graph (G, c) as the maximum number of vertices in G with the same colour. We prove that Colourful Partition is NP-complete even for coloured trees of maximum degree 6 and colour-multiplicity 2. As both problems are equivalent on trees, we obtain the same result for Colourful Components (note that the graphs in the proof of Theorem 3 are not trees).

Third, we fix the number k of colourful components, which gives us the k-Colourful Partition problem. For every $k \geq 1$, this problem is polynomial-time solvable for coloured trees (due to Theorem 7) and for ℓ-coloured graphs for every fixed ℓ (such graphs have at most $k\ell$ vertices). We prove that 2-Colourful Partition is NP-complete for split graphs and for coloured planar bipartite graphs of maximum degree 3 and path-width 3, but polynomial-time solvable for coloured graphs of treewidth at most 2. The latter two results form our main results. They complement Theorem 5, which implies NP-completeness of Colourful Partition for path-width 2.

In Section 4 we show that Colourful Partition and Colourful Components are FPT when parameterized by the treewidth and the number of colours. This generalizes Theorem 6. Our choice for this combination of parameters was guided by our results from Section 3. Our results imply that for the six natural parameters: number of colourful components; maximum degree; number of colours; colour-multiplicity; path-width; treewidth; and *all* their possible combinations, we have obtained either para-NP-completeness or an FPT algorithm for Colourful Partition. This motivates the search for other parameters for this problem. As the vertex cover number of the subdivided stars in the proof of Theorem 5 can be arbitrarily large, it is natural to consider this parameter. As an analogue to Theorem 8, we prove that Colourful Partition is FPT when parameterized by vertex cover number. A vertex u of a coloured graph (G, c) is *uniquely coloured* if its colour $c(u)$ is not used

[1] Since this paper was submitted, Chlebikova and Dallard have independently extended some of our results [5].

on any other vertex of G. It is easy to show NP-hardness for Colourful Partition and Colourful Components for instances with no uniquely coloured vertices (see also Theorem 12). By uncovering a surprising connection with the Robertson-Seymour graph minors project, we also prove that, when parameterized by the number of non-uniquely coloured vertices, Colourful Components is para-NP-hard and Colourful Partition is FPT. Thus there are families of instances on which the two problems have different parameterized complexities.

2 Preliminaries

All our graphs are simple, with no loops or multiple edges. Let $G = (V, E)$ be a graph. A subset $U \subseteq V$ is *connected* if it induces a connected subgraph of G. For a vertex $u \in V$, $N(u) = \{v \mid uv \in E\}$ is the *neighbourhood* of u, and $\deg(u) = |N(u)|$ is the *degree* of u. A graph is *cubic* if every vertex has degree exactly 3. A connected graph on at least three vertices is 2-*connected* if it has no vertex whose removal disconnects the graph. A graph $G = (V, E)$ is *split* if V can be partitioned into two (possibly empty) sets K and I, where K is a clique and I is an independent set. A mapping $c : E \to \{1, 2, 3\}$ is a *proper* 3-*edge colouring* of G if $c(e) \neq c(f)$ for any two distinct edges e and f with a common end-vertex. A set $S \subseteq V$ is a *vertex cover* of G if $G - S$ is an independent set. The Vertex Cover problem asks if a graph has a vertex cover of size at most s for a given integer s. The *vertex cover number* $\mathrm{vc}(G)$ of a graph G is the minimum size of a vertex cover in G. We use the following lemma.

▶ **Lemma 9** ([4]). Vertex Cover *is* NP*-complete for* 2*-connected cubic planar graphs with a proper* 3*-edge colouring given as input.*

A *tree decomposition* of a graph G is a pair $(T, \mathcal{X})$ where T is a tree and $\mathcal{X} = \{X_i \mid i \in V(T)\}$ is a collection of subsets of $V(G)$, called *bags*, such that

1. $\bigcup_{i \in V(T)} X_i = V(G)$;
2. for every edge $xy \in E(G)$, there is an $i \in V(T)$ such that $x, y \in X_i$; and
3. for every $x \in V(G)$, the set $\{i \in V(T) \mid x \in X_i\}$ induces a connected subtree of T.

The *width* of $(T, \mathcal{X})$ is $\max\{|X_i| - 1 \mid i \in V(T)\}$, and the *treewidth* of G is the minimum width over all tree decompositions of G. If T is a path, then $(T, \mathcal{X})$ is a *path decomposition* of G. The *path-width* of G is the minimum width over all path decompositions of G.

For some of our proofs we use variants of the Satisfiability problem. Note here only that in instances of the NP-complete problem Not-All-Equal Positive 3-Satisfiability [12], each clause contains three positive literals and the problem is to find a truth assignment τ where each clause contains at least one true literal and at least one false literal. In this context we call such a τ *satisfying*.

In the remainder of this paper, proofs of results and claims marked (♠) are omitted.

3 Classical Complexity

We will prove four hardness results and one polynomial-time result on Colourful Partition. Note that Colourful Partition belongs to NP. We start by proving the following result.

▶ **Theorem 10.** Colourful Partition *is* NP*-complete for* 3*-coloured* 2*-connected planar graphs of maximum degree* 3.

Proof Sketch. We use a reduction from Vertex Cover. By Lemma 9 we may assume that we are given a 2-connected cubic planar graph G with a proper 3-edge colouring c. From G and c we construct a coloured graph (G', c') as follows.

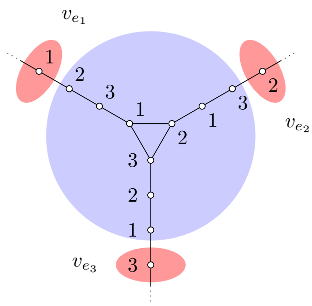

Figure 2 The blue 9-vertex gadget for a vertex v incident to edges e_1, e_2, e_3 in G, connected to the three red vertices $v_{e_1}, v_{e_2}, v_{e_3}$ in G' in the proof of Theorem 10.

- For each $e \in E(G)$ with $c(e) = i$, create a vertex v_e with colour $c'(v_e) = i$ in G' (a *red* vertex).
- For every vertex $v \in V(G)$ with incident edges e_1, e_2, e_3 (with colours 1, 2, 3, respectively), create a copy of the 3-coloured 9-vertex gadget shown in Figure 2 (a *blue* set), and connect it to the red vertices v_{e_1}, v_{e_2} and v_{e_3}, as shown in the same figure.

Note that G' is 3-coloured, 2-connected, planar and has maximum degree 3. We claim that G has a vertex cover of size at most s if and only if G' has a colourful partition of size at most $3n + s$ (♠). ◀

The coloured graph (G', c') constructed in the proof of Theorem 10 can be modified into a colourful graph by omitting exactly one edge adjacent to each red vertex and exactly three edges inside each blue component. It can be readily checked that this is the minimum number of edges required. Hence Colourful Components is polynomial-time solvable on these coloured graphs (G', c'), whereas Colourful Partition is NP-complete. Thus we have:

▶ **Corollary 11.** *There exists a family of instances on which* Colourful Components *and* Colourful Partition *have different complexities (assuming* $\mathsf{P} \neq \mathsf{NP}$*).*

We now present our second, third and fourth NP-hardness results. The last condition in Theorem 12 shows that the number of uniquely coloured vertices is not a useful parameter.

▶ **Theorem 12 (♠).** Colourful Partition *and* Colourful Components *are* NP-*complete for coloured trees with maximum degree at most* 6*, colour-multiplicity* 2*, and no uniquely coloured vertices.*

▶ **Theorem 13 (♠).** 2-Colourful Partition *is* NP-*complete for coloured split graphs.*

▶ **Theorem 14 (♠).** 2-Colourful Partition *is* NP-*complete for coloured planar bipartite graphs of maximum degree* 3 *and path-width* 3*.*

We complement Theorem 14 with a positive result (Theorem 15) whose bounds are best possible due to Theorem 14. The idea behind it is that the sets V_1 and V_2 of a colourful partition of size 2 form connected subtrees in a tree decomposition. Branching over all options, we "guess" two vertices a and b of one bag assigned to different sets V_i. By exploiting the treewidth-2 assumption we can translate the instance into an equivalent instance of 2-Satisfiability.[2]

[2] A graph has treewidth at most 2 if and only if every 2-connected component is a series-parallel graph. This fact can be used for an alternative proof of Theorem 15.

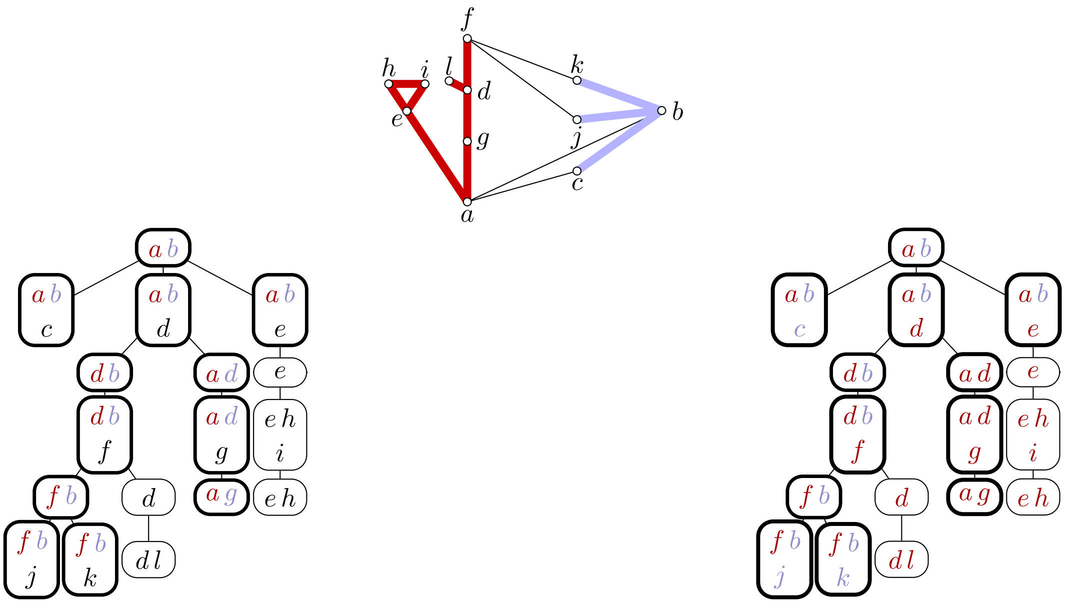

Figure 3 Illustration of the polynomial-time algorithm for treewidth 2. Top: an input graph (the vertex colouring is not represented), with a partition cutting (a, b), i.e. a is in one part, b is in the other. Bottom-left: a tree-decomposition of the graph satisfying Properties 1, 2 and 3. The bags in the head subtree are in bold. Precuts computed as in Claim 19 are shown using colours on the top line of bags that have precuts. For example, the bag $\{d, b, f\}$ is precut on (d, b), as can be verified directly in the graph (if d, b and f are not all in the same part of an (a, b)-partition, then d must be in the same part as a). Here f is attached to b, and e is attached to a but not to b. Bottom-right: the tree decomposition showing a possible output of the 2-Satisfiability formula, where the colours represent the values of x_u.

▶ **Theorem 15.** 2-Colourful Partition *is polynomial-time solvable for coloured graphs of treewidth at most* 2.

Proof. Let (G, c) be a coloured graph on n vertices such that G has treewidth at most 2. Without loss of generality we may assume that G is connected. We may assume that G is not colourful, otherwise we are trivially done. If G has treewidth 1, then it is a tree and we apply Theorem 7. Hence we may assume that G has treewidth 2. Let $(T, \mathcal{X})$ be a tree decomposition of G of width 2 (so all bags of $\mathcal{X}$ have size at most 3). We can obtain this tree decomposition in linear time [13]. Let us state and then explain two properties that we may assume hold for $(T, \mathcal{X})$. These properties, along with the main definitions necessary for our algorithm, are illustrated in Figure 3.

1. For any two adjacent nodes i, j in T, one of X_i and X_j strictly contains the other.
2. All bags are pairwise distinct.

In fact, what we will show is that if we find that $(T, \mathcal{X})$ does not have these two properties, then we can make simple changes to obtain a tree decomposition that does. First, if there are two adjacent nodes i, j in T such that neither X_i nor X_j contains the other, we remove the edge ij from T, create a new node k that is adjacent to both i and j and let $X_k = X_i \cap X_j$. Note that $X_k \neq \emptyset$, as G is connected. We now have a tree decomposition that has Property 1 unless $X_i = X_j$ for a pair of adjacent nodes i and j. But now for *every* pair of identical bags X_i and X_j, we delete j and make each of its neighbours adjacent to i and so obtain a tree decomposition with Properties 1 and 2.

Let a and b be a fixed pair of adjacent vertices in G. Almost all of the remainder of this proof is concerned with describing an algorithm that decides whether or not G has a colourful partition of size 2 in which a and b belong to different parts. Clearly such an algorithm suffices: we can apply it to each of the $O(n^2)$ pairs of adjacent vertices in G to determine whether G has *any* colourful partition of size 2 (since such a partition must separate at least one pair of adjacent vertices).

As a and b are adjacent, there is at least one bag that contains both of them. We may assume that this bag is $X_0 = \{a, b\}$; otherwise we add a new node 0 to T with $X_0 = \{a, b\}$ and make 0 adjacent to i such that X_i is a (larger) bag that contains both a and b.

We orient all edges of T away from 0 and think of T as being rooted at 0. We write $i \to j$ to orient an edge from i to j. If $i \to j$ is present in T, then i is the *parent* of j and j is a *child* of i. The *head subtree* of T is the subtree obtained by removing all 1-vertex bags along with all their descendants. For any oriented edge $i \to j$ in T, we have either $X_i \subsetneq X_j$ or $X_j \subsetneq X_i$ by Property 1. The tree $T[i]$ is the subtree of T rooted at i; in particular $T = T[0]$. The set $V[i]$ denotes $\bigcup_{j \in V(T[i])} X_j$. As we shall explain, we may assume the following property.

3. For any node i of T, the subgraph $G[V[i]]$ induced by $V[i]$ is connected.

It is possible that we must again modify the tree decomposition to obtain this property. Suppose that it does not hold for some node i. That is, the vertices of $V[i]$ can be divided into two sets U and W such that there is no edge from U to W in G. We create two trees $T_{i,U}$ and $T_{i,W}$ that are isomorphic to $T[i]$: for each vertex j in $T[i]$, we let j_U and j_W be the corresponding nodes in $T_{i,U}$ and $T_{i,W}$ respectively and let $X_{j_U} = X_j \cap U$, $X_{j_W} = X_j \cap W$. Then the tree decomposition is modified by replacing $T[i]$ by $T_{i,U}$ and $T_{i,W}$ and making each of i_U and i_W adjacent to the parent of i. (The vertex i certainly has a parent since $i = 0$ would imply that G is not connected.) If at any point we create a node whose associated bag is empty or identical to that of its parent, we delete it and make its children (if it has any) adjacent to its parent. Note that although the number of bags may increase through this operation, the sum of the sizes of the bags of T can only decrease, and at least one bag gets its size reduced, so overall this operation only needs to be applied a polynomial number of times. In this way we obtain a decomposition that now satisfies each of Properties 1, 2 and 3.

We say that a colourful partition $P = (V_1, V_2)$ of G is an (a, b)*-partition* if $a \in V_1$ and $b \in V_2$ and we say that P *cuts* a pair (u_1, u_2) if $u_1 \in V_1$ and $u_2 \in V_2$. We emphasize that the order is important. A colourful partition P *respects* a bag X if $X \subseteq V_1$ or $X \subseteq V_2$. Note that every colourful partition respects all 1-vertex bags. Let X be a bag that contains vertices u and v. Then X is *precut* on (u, v) if every (a, b)-partition either cuts (u, v) or respects X. Note that X_0 is precut on (a, b) by definition. We now prove three structural claims.

▷ Claim 16. If an (a, b)-partition P respects a bag X_i, then it respects every X_j with $i \to j$.

Proof. Consider the set $\mathcal{C}$ of bags that are not respected by P. Then $\mathcal{C}$ is the intersection of the set of bags containing at least one vertex from V_1 and the set of bags containing at least one vertex from V_2. Since both V_1 and V_2 are connected in G, the set of nodes whose bags contain at least one vertex from V_1 and the set of nodes whose bags contain at least one vertex from V_2 induce subtrees of T. Hence their intersection, $\mathcal{C}$, is a set of nodes that also induce a subtree of T. Since (a, b) is not respected by P, we know $0 \in \mathcal{C}$. This means that if i is not in $\mathcal{C}$, then for every other vertex $j \in T[i]$, j cannot be in $\mathcal{C}$. ◁

▷ Claim 17. Let $i \to j$ be an oriented edge of T with $X_i = \{u, v\}$ and $|X_j| = 3$. If X_i is precut on (u, v), then X_j is precut on (u, v).

Proof. By Property 1, we know that $X_i \subsetneq X_j$, so X_j contains u and v. Suppose (u,v) is a precut on X_i and let P be an (a,b)-partition. If P cuts (u,v) then we are done. Otherwise, P respects $X_i = \{u,v\}$, so by Claim 16, P also respects X_j. Thus (u,v) is a precut on X_j. $\triangleleft$

$\triangleright$ Claim 18. Let $i \to j$ be an oriented edge of T with $X_i = \{u,v,w\}$ and $|X_j| = 2$ such that X_i is precut on (u,v). If $X_j = \{v,w\}$, then X_j is precut on (w,v). If $X_j = \{u,w\}$, then X_j is precut on (u,w).

Proof. Let $P = (V_1, V_2)$ be an (a,b)-partition of G. Suppose P does not respect X_j. Then if X_j contains w, we must have that w is not in the same part of the partition as the other vertex – either u or v – of X_j. By Claim 16, P does not respect X_i either, so we know $u \in V_1$ and $v \in V_2$. Thus if $X_j = \{v,w\}$, then $w \in V_1$, and if $X_j = \{u,w\}$, then $w \in V_2$. $\triangleleft$

By Claim 16 and the fact that 1-vertex bags are respected, for every node i not in the head subtree of T, X_i is respected by every (a,b)-partition of G.

$\triangleright$ Claim 19. For every node i in the head subtree, X_i is precut on some pair of its vertices, and these precuts can be computed in linear time.

Proof. We show that Claim 19 holds by proving a slightly stronger statement: for all d, for each node j in the head subtree at distance d from 0, X_j is precut on a pair of its vertices, and if X_j contains three vertices, then it is precut on some pair (u,v) such that $X_i = \{u,v\}$ where i is the parent of j. We prove this by induction on d. The base case holds as X_0 is precut on (a,b). For the inductive case, suppose that j is some node at distance $d > 0$ with parent i. As i and j are in the head subtree, each of X_i and X_j has either two or three vertices. Suppose first that $X_j = \{u,v,w\}$, and then we may assume that X_i is, say, $\{u,v\}$. (Recall that $X_i \subsetneq X_j$ or $X_i \subsetneq X_j$ by Property 1.) By the induction hypothesis, we know that X_i is precut on (u,v) or on (v,u) and so, by Claim 17, X_j is precut in the same way. Now suppose that X_j has two vertices and $X_i = \{u,v,w\}$. Since X_0 has two vertices, it follows that $X_i \neq X_0$, so i has a parent h, and we can suppose that $X_h = \{u,v\}$. By the induction hypothesis, without loss of generality we may assume X_h is precut on (u,v), so X_i is precut on (u,v) by the same argument as above. Then, as $\mathcal{X}$ has no identical bags by Property 2, we find that $X_j = \{u,w\}$ or $X_j = \{v,w\}$. By Claim 18, X_j is precut. $\triangleleft$

We say that two vertices u and v of G are *attached* if they are adjacent or $\mathcal{X}$ contains the bag $\{u,v\}$. If u and v are not attached, then they are *detached*.

$\triangleright$ Claim 20. Let $X_i = \{u,v,w\}$ be a 3-vertex bag precut on (u,v). If w and u (or w and v) are detached, then in every (a,b)-partition of G the vertices w and v (respectively w and u) are in the same colourful set.

Proof. Let $P = (V_1, V_2)$ be an (a,b)-partition. If P respects X_i, then w is in the same colourful set as both u and v. Thus we may assume that P does not respect X_i and so $u \in V_1$ and $v \in V_2$, since X_i is precut on (u,v). We may assume without loss of generality that w is detached from u and so we must show that $w \in V_2$.

Let us assume instead that $w \in V_1$ and derive a contradiction. By the connectivity of $G[V_1]$, there exists an induced path Q on ℓ edges in $G[V_1]$ that connects u to w. As u and w are detached, they are not adjacent, so ℓ is at least 2. No internal vertex of Q is in X_i since u and w are its end-vertices and v is not in V_1. Let $X_{i_1}, \ldots, X_{i_\ell}$ be bags of $\mathcal{X}$ such that X_{i_j} is a bag that contains the pair of vertices joined by the jth edge of Q. We take a walk in T from i_1 to i_ℓ by stitching together paths from i_j to i_{j+1}, $1 \leq j \leq \ell - 1$. As X_{i_j} and $X_{i_{j+1}}$ correspond to incident edges of Q, and both X_{i_j} and $X_{i_{j+1}}$ contain the internal vertex of Q incident with both these edges, every bag of the nodes along the path between them also contains this vertex. Thus every bag of the nodes along our walk from i_1 to i_ℓ contains an internal vertex of Q and so none of these bags is X_i. Therefore our walk must be contained within a connected component of $T \setminus i$. Let j be the node of this component adjacent to i in T. We note that X_i contains u and w, X_{i_1} contains u, X_{i_ℓ} contains w, and the paths from i to each of i_1 and i_ℓ go through j. Thus X_j must contain u and w, and so, by Property 1, it follows that $X_j = \{u, w\}$. As u and w are detached, we have a contradiction. $\lhd$

We now build ϕ, an instance of 2-SATISFIABILITY, that will help us to find an (a, b)-partition of G (if one exists). For each vertex u of G, we create a variable x_u (understood as "$u \in V_1$"). We add clauses (or pairs of clauses) on two variables equivalent to the following statements:

$$x_a = \texttt{true} \quad (1)$$

$$x_b = \texttt{false} \quad (2)$$

$$x_u \neq x_v \text{ if } u \text{ and } v \text{ have the same colour} \quad (3)$$

For any bag X_i precut on some pair (u, v),

$$x_u \vee \neg x_v \quad (4)$$

$$\neg x_u \Rightarrow \neg x_w \quad \forall w \in V[i] \setminus \{u\} \quad (5)$$

$$x_v \Rightarrow x_w \quad \forall w \in V[i] \setminus \{v\} \quad (6)$$

For each bag $X_i = \{u\}$ of size 1,

$$x_w = x_u \quad \forall w \in V[i] \setminus \{u\} \quad (7)$$

For each bag $\{u, v, w\}$ of size 3 precut on (u, v)

$$x_w = x_u \text{ if } w \text{ is detached from } v \quad (8)$$

$$x_w = x_v \text{ if } w \text{ is detached from } u \quad (9)$$

We now claim that G has an (a, b)-partition if and only if ϕ is satisfiable.

First suppose that G has an (a, b)-partition P. For each vertex u in G, set $x_u = \texttt{true}$ if $u \in V_1$, and $x_u = \texttt{false}$ otherwise. As P cuts (a, b), Statements (1) and (2) are satisfied. As P is colourful, Statement (3) is satisfied.

Consider a bag X_i precut on (u, v). Two cases are possible: either P cuts (u, v), or X is respected by P. In the first case we have $x_u = \texttt{true}$ and $x_v = \texttt{false}$, which is enough to satisfy Statements (4), (5) and (6). In the second case, $x_u = x_v$ (which satisfies Statement (4)), and by Claim 16, for each j in $T[i]$, X_j is respected by P. Thus all vertices in $V[i]$ are in the same subset V_1 or V_2, hence they have the same value of x_w, which satisfies Statements (5) and (6).

For Statement (7), note that each bag X_i of size 1, and, by Claim 16, every bag X_j such that j is a descendent of i is always respected, so all values of x_w for $w \in V[i]$ are identical. Statements (8) and (9) follow from Claim 20.

Now suppose ϕ is satisfiable. We construct two disjoint sets V_1 and V_2 by putting a vertex u in V_1 if $x_u = \texttt{true}$ in V_1 and in V_2 otherwise. We show (V_1, V_2) is an (a, b)-partition of G. By Statements (1) and (2), $a \in V_1$ and $b \in V_2$. By Statement (3), two vertices of the same colour cannot belong to the same set. It remains to prove that both sets are connected.

In fact, we shall prove by induction on d, that for the set of vertices belonging to bags of nodes at distance at most d from 0 in T, the two subsets found by dividing the set according to membership of V_1 or V_2 each induce a connected subgraph of G. For the base case, we consider $X_0 = \{a, b\}$ and the two subsets contain a single vertex so we are done. For the inductive case, we consider a node j at distance $d > 0$ from 0. Let i be the parent of j in T. It is enough to show that any vertex in $X_j \setminus X_i$ is in the same component of $G[V_1]$ or $G[V_2]$ as a vertex of X_i assigned to the same set (V_1 or V_2).

Let us first assume that X_j is in the head subtree. If $|X_j| = 2$, then $X_j \subsetneq X_i$ so we may assume that $X_j = \{u, v, w\}$ and also that $X_i = \{u, v\}$, and X_i and X_j are both precut on (u, v) (using Claims 17 and 19).

We distinguish three cases.

▷ Case 1. $x_u = x_v = \texttt{true}$.

Considering Statement (6) for bag X_i, we have $x_w = \texttt{true}$, as well as $x_{w'} = \texttt{true}$ for every vertex $w' \in V[i] \setminus X_i$. Hence $V[i] \subseteq V_1$. By Property 3, $G[V[i]]$ is connected, so w is in the same component of $G[V_1]$ as v.

▷ Case 2. $x_u = x_v = \texttt{false}$.

Symmetrically to Case 1, we can show that w is in the same component of $G[V_2]$ as u.

▷ Case 3. $x_u \neq x_v$.

By Statement (4), in this case we have $x_u = \texttt{true}$ and $x_v = \texttt{false}$. Suppose that $x_w = \texttt{true}$ (the case for $x_w = \texttt{false}$ again follows symmetrically). Since $x_w \neq x_v$, by Statement (9), w must be attached to u. If there is an edge from u to w, then we are done. If there is no such edge, then $\mathcal{X}$ has a bag $X_k = \{u, w\}$, and k must be a child of j. As k is in the head subtree, X_k is precut on (u, w) or (w, u) by Claim 19. By Statements (5) and (6), since $x_u = x_w = \texttt{true}$, $x_{w'} = \texttt{true}$ for every $w' \in V[k] \setminus \{u, w\}$ and thus $V[k] \subseteq V_1$. Since $G[V[k]]$ is connected and contains both u and w, we are done.

Now we consider the case when X_j is not in the head subtree. Thus X_j is in a subtree that has at its root a bag containing a single vertex u and, by Statement (7), every vertex in the bags of the subtree are in the same subset V_1 or V_2 as u and these vertices together induce a connected subgraph. Thus, as u is also in the parent of the root of the subtree (since G is connected, the parent is not empty), we are done.

Thus we have a polynomial-time algorithm to decide 2-Colourful Partition on the instance (G, c). First we compute in polynomial time a tree-decomposition $(T, \mathcal{X})$ of G with Properties 1, 2 and 3. Then for every pair of adjacent vertices a and b in G, we check whether there is a colourful (a, b)-partition of size 2 in polynomial time using the corresponding 2-Satisfiability formula ϕ. ◀

4 Parameterized Complexity

Guided by some observations on the aforementioned hardness constructions, we present three new FPT results in this section. The first one generalizes Theorem 6.

▶ **Theorem 21 (♠).** *Colourful Partition and Colourful Components are FPT when parameterized by treewidth plus the number of colours.*

The proof of our second FPT result uses similar arguments to the proof sketch of Theorem 8 given in [10]. However, its details are different, as optimal solutions for Colourful Partition do not necessarily translate into optimal solutions for Colourful Components.

▶ **Theorem 22.** Colourful Partition *is* FPT *when parameterized by vertex cover number.*

Proof Sketch. In fact we will show the result for the optimization version of Colourful Partition. Let (G, c) be a coloured graph. We will prove that we can find the size of a *minimum* colourful partition (one with smallest size) in FPT time. By a simple greedy argument, we can find a vertex cover S of G that contains at most $2\,\mathrm{vc}(G)$ vertices. It is therefore sufficient to show that Colourful Partition is FPT when parameterized by $|S|$. If two vertices of G have the same colour then they will always be in different colourful components of G. Thus if two vertices with the same colour are adjacent, we can delete the edge that joins them, that is, we may assume that c is a proper colouring of G (note that deleting edges from G maintains the property that S is a vertex cover). Since S is a vertex cover, $T = V(G) \setminus S$ is an independent set. Let C be the set of colours used on G. We let $s = |S|$ and for a set $S' \subseteq S$, we let $T_i(S')$ be the set of vertices with colour i whose neighbourhood in S is S', that is, for all $u \in T_i(S')$, we have that $N_S(u) = S'$ and $c(u) = i$.

▷ Rule 1. If there is a colour $i \in C$ and a set $S' \subseteq S$ such that $|T_i(S')| \geq s + 1$, then delete $|T_i(S')| - s$ (arbitrary) vertices of $T_i(S')$ from G.

We claim that we can safely apply Rule 1 (♠) and apply it exhaustively. We again denote the resulting instance by (G, c) and let $T = V(G) \setminus S$. Note that now $|T_i(S')| \leq s$ for every $i \in C$ and every $S' \subseteq S$. Consequently, the number of vertices of T with colour i is at most $s2^s$. Note that this means that $|T| \leq |C|s2^s$ and thus the total number of vertices in G is at most $s + |C|s2^s$. Hence, it remains to bound the size of C by a function of s.

We let C_T denote the set of colours that appear on vertices of T but not on vertices of S. For two colours $i, j \in C_T$, if $T_i(S') = T_j(S')$ holds for all $S' \subseteq S$, then we say that i and j are *clones* and note that these colours are interchangeable. For a colourful partition $P = (V_1, \ldots, V_k)$ for G, we let $P_S = (V_1 \cap S, \ldots, V_k \cap S)$ be the partition of S *induced by* P (note that in this case we allow some of the blocks to be empty).

We consider each partition Q of S. If a block of Q is not colourful, then we discard Q. Otherwise we determine a minimum colourful partition P for G with $P_S = Q$; note that such a partition P may not exist, as it may not be possible to make the blocks of Q connected (by using vertices T in addition to edges both of whose endpoints lie in S). We will choose the colourful partition for G that has minimum size overall. Let Q be a partition of S in which each block is colourful.

▷ Rule 2. If there are s distinct colours $i_1, \ldots, i_s$ in C_T that are pairwise clones, then delete all vertices with colour i_s from G.

We claim that we can safely apply Rule 2 (♠) and apply it exhaustively; again call the resulting graph G and define S and T as before.

▷ Claim 23 (♠). $|C_T| \leq (s-1)(s+1)^{2^s}$.

We continue as follows. The number of different colours used on vertices of S is at most s. Hence $C \setminus C_T$ has size at most s. Recall that for every colour $i \in C$, the number of vertices of T with colour i is at most $s2^s$. We combine these two facts with Claim 23 to deduce that $|V| = |S| + |T| \leq s + |C|s2^s = s + |C \setminus C_T|s2^s + |C_T|s2^s \leq s + s^2 2^s + (s-1)(s+1)^{2^s} s2^s$. Then by brute force we can compute a minimum colourful partition P for G subject to the restriction that $P_S = Q$ in $f(s)$ time for some function f that only depends on s.

The correctness of our FPT-algorithm follows from the above description. It remains to analyze the running time. Applying Rule 1 exhaustively takes $O(2^s|C|) = O(2^s n)$ time, as the number of different subsets $S' \subseteq S$ is 2^s. We then branch into at most s^s directions by considering every partition of S. Applying Rule 2 exhaustively takes $O(2^s n^2)$ time, as for each colour $i \in C_T$ we first calculate the values of $|T_i(S')|$ for every $S' \subseteq S$, which can be done in $O(2^s n)$ time. Doing this for every colour takes a total of $O(2^s n^2)$ time and partitioning the colours into sets that are clones can be done in $O(2^{2s} n^2)$ time, and deleting colours can be done in $O(sn)$ time. As every auxiliary graph F has at most n vertices, we can compute a maximum matching in F in $O(n^{\frac{5}{2}})$ time by using the Hopcroft-Karp algorithm [9]. Finally, translating a minimum solution into a minimum solution for the graph in which we restored the vertices we removed due to exhaustive application of Rules 1 and 2 takes $O(n)$ time. This means that the total running time is $O(2^s n) + s^s(O(2^{2s} n^2) + O(s) + O(n^{\frac{5}{2}}) + f(s) + O(n)) = f'(s)O(n^{\frac{5}{2}})$ for some function f' that only depends on s, as desired. ◀

The Disjoint Connected Subgraphs problem takes as input a graph G with r pairwise disjoint subsets $Z_1, \ldots, Z_r$ of $V(G)$ for some $r \geq 1$. It asks whether we can partition $V(G) \setminus (Z_1 \cup \cdots \cup Z_r)$ into sets $S_1, \ldots, S_r$ such that every $S_i \cup Z_i$ induces a connected subgraph of G. Robertson and Seymour introduced this problem in their graph minor project and proved that it is cubic-time solvable as long as $Z_1 \cup \cdots \cup Z_r$ has constant size [11].

▶ **Theorem 24.** *When parameterized by the number of non-uniquely coloured vertices,* Colourful Components *is para-*NP*-complete, but* Colourful Partition *is* FPT*.*

Proof. The Multiterminal Cut problem is to test for a graph G, integer p and terminal set S, if there is a set E' with $|E'| \leq p$ such that every terminal in S is in a different component of $G - E'$. This problem is NP-complete even if $|S| = 3$ [6]. To prove the first part, give each of the three vertices in S colour 1 and the vertices in $G - S$ colours $2, \ldots, |V| - 2$.

To prove the second part, let (G, c) be a coloured graph and k be an integer. We assume without loss of generality that G is connected. Let Q with $|Q| = q$ be the set of non-uniquely coloured vertices. If $k \geq q$, then place each of the q vertices of Q in a separate component and assign the uniquely coloured vertices to components in an arbitrary way subject to maintaining connectivity of the q components. This yields a colourful partition of (G, c) of size at most k. Now assume that $k \leq q - 1$. We consider every possible partition of Q into k sets $Z_1, \ldots, Z_k$, where some of the sets Z_i may be empty. It remains to solve Disjoint Connected Subgraphs on the input $(G, Z_1, \ldots, Z_k)$. Note that $|Z_1| + \cdots + |Z_k|$ has size q. Hence, by the above result of Robertson and Seymour [11], solving Disjoint Connected Subgraphs takes cubic time. As there are $O(q^q)$ partitions to consider, the result follows. ◀

5 Conclusions

We showed that Colourful Partition and Colourful Components are NP-complete for coloured trees of maximum degree at most 6 (and colour-multiplicity 2). What is their complexity for coloured trees of maximum degree d for $3 \leq d \leq 5$? Colourful Components is known to be NP-complete for 3-coloured graphs of maximum degree 6 (Theorem 3); we also ask if one can prove a result analogous to Theorem 10: what is its complexity for coloured graphs of maximum degree 3? Our main result is that 2-Colourful Partition is NP-complete for coloured (planar bipartite) graphs of path-width 3 (and maximum degree 3), but polynomially solvable for coloured graphs of treewidth 2. We believe that the latter result can be extended to k-Colourful Partition ($k \geq 3$), but leave this for future research. A more interesting question is if the problem is FPT for treewidth 2 when parameterized by k.

References

1 Anna Adamaszek and Alexandru Popa. Algorithmic and Hardness Results for the Colorful Components Problems. *Algorithmica*, 73(2):371–388, 2015.

2 Sharon Bruckner, Falk Hüffner, Christian Komusiewicz, Rolf Niedermeier, Sven Thiel, and Johannes Uhlmann. Partitioning into Colorful Components by Minimum Edge Deletions. *Proc. CPM 2012, LNCS*, 7354:56–69, 2012.

3 Laurent Bulteau, Konrad K. Dabrowski, Guillaume Fertin, Matthew Johnson, Daniël Paulusma, and Stéphane Vialette. Finding a Small Number of Colourful Components. Manuscript, 2018. `arXiv:1808.03561`.

4 Laurent Bulteau, Guillaume Fertin, and Irena Rusu. Maximal strip recovery problem with gaps: Hardness and approximation algorithms. *Journal of Discrete Algorithms*, 19:1–22, 2013.

5 Janka Chlebikova and Clément Dallard. A Complexity Dichotomy for Colourful Components Problems on k-caterpillars and Small-Degree Planar Graphs. Manuscript, 2019. `arXiv:1902.11191`.

6 Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The Complexity of Multiterminal Cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.

7 Riccardo Dondi and Florian Sikora. Parameterized complexity and approximation issues for the colorful components problems. *Theoretical Computer Science*, 739:1–12, 2018.

8 George He, Jiping Liu, and Cheng Zhao. Approximation Algorithms for Some Graph Partitioning Problems. *Journal of Graph Algorithms and Applications*, 4(2):1–11, 2000.

9 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2(4):225–231, 1973.

10 Neeldhara Misra. On the Parameterized Complexity of Colorful Components and Related Problems. *Proc. IWOCA 2018, LNCS*, 10979:237–249, 2018.

11 Neil Robertson and Paul D. Seymour. Graph Minors. XIII. The Disjoint Paths Problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.

12 Thomas J. Schaefer. The Complexity of Satisfiability Problems. *Proc. STOC 1978*, pages 216–226, 1978.

13 Joseph A. Wald and Charles J. Colbourn. Steiner trees, partial 2–trees, and minimum IFI networks. *Networks*, 13(2):159–167, 1983.

14 Chunfang Zheng, Krister M. Swenson, Eric Lyons, and David Sankoff. OMG! Orthologs in multiple genomes – competing graph-theoretical formulations. *Proc WABI 2011, LNBI*, 6833:364–375, 2011.