Parameterized Complexity of Edge-Coloured and Signed Graph Homomorphism Problems

Florent Foucaud
Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067 Orléans Cedex 2, France
Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France

Hervé Hocquard
Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France

Dimitri Lajou
Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France

Valia Mitsou
Université Paris-Diderot, IRIF, CNRS, 75205, Paris, France

Théo Pierron
Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, F-33400 Talence, France
DIMEA, Masaryk University, 60200 Brno, Czech republic

Abstract

We study the complexity of graph modification problems with respect to homomorphism-based colouring properties of edge-coloured graphs. A homomorphism from an edge-coloured graph G to an edge-coloured graph H is a vertex-mapping from G to H that preserves adjacencies and edge-colours. We consider the property of having a homomorphism to a fixed edge-coloured graph H, which generalises the classic vertex-colourability property. The question we are interested in is the following: given an edge-coloured graph G, can we perform k graph operations so that the resulting graph admits a homomorphism to H? The operations we consider are vertex-deletion, edge-deletion and switching (an operation that permutes the colours of the edges incident to a given vertex). Switching plays an important role in the theory of signed graphs, that are 2-edge-coloured graphs whose colours are the signs $+$ and $-$. We denote the corresponding problems (parameterized by k) by Vertex Deletion-H-Colouring, Edge Deletion-H-Colouring and Switching-H-Colouring. These problems generalise the extensively studied H-Colouring problem (where one has to decide if an input graph admits a homomorphism to a fixed target H). For 2-edge-coloured H, it is known that H-Colouring already captures the complexity of all fixed-target Constraint Satisfaction Problems.

Our main focus is on the case where H is an edge-coloured graph of order at most 2, a case that is already interesting since it includes standard problems such as Vertex Cover, Odd Cycle Transversal and Edge Bipartization. For such a graph H, we give a PTime/NP-complete complexity dichotomy for all three Vertex Deletion-H-Colouring, Edge Deletion-H-Colouring and Switching-H-Colouring problems. Then, we address their parameterized complexity. We show that all Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring problems for such H are FPT. This is in contrast with the fact that already for some H of order 3, unless PTime = NP, none of the three considered problems is in XP, since 3-Colouring is NP-complete. We show that the situation is different for Switching-H-Colouring: there are three 2-edge-coloured graphs H of order 2 for which Switching-H-Colouring is W[1]-hard, and assuming the ETH, admits no algorithm in time $f(k)n^{o(k)}$ for inputs of size n and for any computable function f. For the other cases, Switching-H-Colouring is FPT.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis; Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Graph homomorphism, Graph modification, Edge-coloured graph, Signed graph

Digital Object Identifier 10.4230/LIPIcs.IPEC.2019.15

14th International Symposium on Parameterized and Exact Computation (IPEC 2019).
Editors: Bart M. P. Jansen and Jan Arne Telle; Article No. 15; pp. 15:1–15:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Related Version A full version of the paper is available at `https://arxiv.org/abs/1910.01099`.

Funding This research was financed by the ANR project HOSIGRA (ANR-17-CE40-0022) and the IFCAM project "Applications of graph homomorphisms" (MA/IFCAM/18/39).

1 Introduction

Graph colouring problems such as k-Colouring are among the most fundamental problems in algorithmic graph theory. Problem H-Colouring is a homomorphism-based generalisation of k-Colouring that is extensively studied [8, 14, 18, 25]. Considering a fixed graph H, in H-Colouring one asks whether an input graph G admits a homomorphism (an edge-preserving vertex-mapping) to H. k-Colouring is the same problem as K_k-Colouring, where K_k is the complete graph of order k (the order of a graph is its number of vertices).

We will consider parameterized variants of H-Colouring where H is an edge-coloured graph. We say that a graph is *t-edge-coloured* if its edges are coloured with at most t colours. We allow loops and multiple edges, but multiple edges of the same colour are irrelevant in H. We sometimes give actual colour names to the colours: red, blue, green. For 2-edge-coloured graphs, we will use red and blue as the two edge colours. A standard uncoloured graph can be seen as 1-edge-coloured. For two edge-coloured graphs G and H, a homomorphism from G to H is a vertex-mapping $\varphi : V(G) \to V(H)$ such that, if xy is an edge of colour i in G, then $\varphi(x)\varphi(y)$ is an edge of colour i in H. Whenever such a φ exists, we say that G maps to H, and we write $G \xrightarrow{ec} H$.

The H-Colouring problems are well-studied, see for example [1, 2, 3, 4, 5]. They are special cases of *Constraint Satisfaction Problems* (CSPs). A large set of CSPs can be modeled by homomorphisms of general relational structures to a fixed relational structure H [14]. The corresponding decision problem is noted H-CSP. When H has only binary relations, H can be seen as an edge-coloured graph (a relation corresponds to the set of edges of a given colour) and H-CSP is exactly H-Colouring. The complexity of H-CSP has been the subject of intensive research in the last decades, since Feder and Vardi conjectured in [14] that H-CSP is either PTime or NP-complete – a statement that became known as the Dichotomy Conjecture. The latter conjecture was recently solved in [7, 30] independently; the criterion for H-CSP to be in PTime is based on certain algebraic properties of H. Nevertheless, determining whether a structure H satisfies this criterion is not an easy task (even for targets as simple as oriented trees [8]). Thus, the study of more simple and elegant complexity classifications for relevant special cases is of high importance.

The complexity of H-Colouring when H is uncoloured is well-understood: it is in PTime if H contains a loop or is bipartite; otherwise it is NP-complete [18]. This was one of the early dichotomy results in the area. On the other hand, when H is a 2-edge-coloured graph, it was proved that the class of H-Colouring problems captures the difficulty of the whole class of H-CSP problems [5], and thus the dichotomy classification for this class of problems is expected to be much more intricate.

Our goal is to study generalisations of H-Colouring problems for edge-coloured graphs by enhancing them as *modification problems*. In this setting, given a graph property $\mathcal{P}$ and a graph operation π, the graph modification problem for $\mathcal{P}$ and π asks whether an input graph G can be made to satisfy property $\mathcal{P}$ after applying operation π a given number k of times. This is a classic setting studied extensively both in the realms of classical and parameterized complexity, see for example [9, 22, 23, 28]. In this context, the most studied graph operations are vertex deletion (VD) and edge-deletion (ED), see the seminal papers [23, 28].

For a fixed graph H, let $\mathcal{P}(H)$ denote the property of admitting a homomorphism to H. Certain standard computational problems can be stated as graph modification problems to $\mathcal{P}(H)$. For example, VERTEX COVER is the graph modification problem for property $\mathcal{P}(K_1)$ and operation VD. Similarly, ODD CYCLE TRANSVERSAL and EDGE BIPARTIZATION are the graph modification problems for $\mathcal{P}(K_2)$ and VD, and $\mathcal{P}(K_2)$ and ED, respectively.

When considering edge-coloured graphs with only two edge-colours, another operation of interest is *switching*: to switch at a vertex v is to change the colour of all edges incident with v. (Note that a loop does not change its colour under switching.) This operation is of prime importance in the context of signed graphs. A *signed graph* is a 2-edge-coloured graph in which the two colours are denoted by signs ($+$ and $-$). A graph is called *balanced* if it can be switched to be all-positive. The concepts of signed graphs, balance and switching, were introduced and developed in [17, 29] and have many interesting applications, in particular in social networks and biological dynamical systems (see [19] and the references therein).

The switching operation plays an important role in the study of homomorphisms of signed graphs, a concept defined in [26] which has many connections to deep questions in structural graph theory. In their definition, before mapping the vertices, one may perform any number of switchings. (Note that when switching at a set S of vertices of a signed graph G, the order does not matter: ultimately, only the edges between S and its complement $V(G) \setminus S$ change their sign.) The algorithmic complexity of this problem was studied in [5, 6, 16]. Herein, we will consider edge-coloured graph modification problems for property $\mathcal{P}(H)$ (for fixed edge-coloured graphs H) and for graph operations VD, ED and SW.

A parameterized problem is a decision problem with a parameter of the input. It is *fixed parameter tractable* (FPT) if for any input I with parameter value k, it can be solved in time $O(f(k)|I|^c)$ for a computable function f and integer c. It is in the class XP if it can be solved in time $|I|^{g(k)}$ for a computable function g. It is W[1]-hard if all problems in the class W[1] can be reduced in FPT time to it. For more details, see the books [11, 12]. Let us now formally define the problems of interest to us (the parameter is always k).

> VERTEX DELETION-(RESP. EDGE DELETION)-H-COLOURING **Parameter:** k.
> **Input:** An edge-coloured graph G, an integer k.
> **Question:** Is there a set S of k vertices (resp. edges) of G such that $(G - S) \xrightarrow{ec} H$?

> SWITCHING-H-COLOURING **Parameter:** k.
> **Input:** A 2-edge-coloured graph G, an integer k.
> **Question:** Is there a set S of k vertices of G such that the 2-edge-coloured graph G' obtained from G by switching at every vertex of S satisfies $G' \xrightarrow{ec} H$?

In the study of the three above problems, one may assume that H is a *core* (that is, H does not have a homomorphism to a proper subgraph of itself). Indeed, it is well-known that for any subgraph H' of H with $H \xrightarrow{ec} H'$, we have $G \xrightarrow{ec} H$ if and only if $G \xrightarrow{ec} H'$ [3].

Of course, whenever H-COLOURING is NP-complete, all three above problems are NP-complete, even when $k = 0$, and so they are not in XP (unless PTime = NP). This is for example the case when H is a monochromatic triangle: then we have 3-COLOURING. Thus, from the point of view of parameterized complexity, it is of primary interest to consider these problems for edge-coloured graphs H such that H-COLOURING is in PTime. (In that case a simple brute-force algorithm iterating over all k-subsets of vertices of G implies that the three problems are in XP.) For classic graphs, the only cores H for which H-COLOURING is in PTime are the three connected graphs with at most one edge (a single vertex with no edge, a single vertex with a loop, two vertices joined by an edge), so in that case the interest of these problems is limited. However, for many interesting families of edge-coloured graphs H, the problem H-Colouring is in PTime, and the class of such graphs H is not very well-understood, see [2, 3, 4]. Even when H is a 2-edge-coloured cycle, tree or complete graph, there are infinitely many H with H-Colouring NP-complete and infinitely many H where it is in PTime [2].

Recall that when H is a single vertex with no loop, Vertex Deletion-H-Colouring is exactly Vertex Cover. If H has a single edge, Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring are Odd Cycle Transversal and Edge Bipartization, respectively. For H consisting of a single (blue) loop, Switching-H-Colouring for $k = |V(G)|$ consists in checking whether the given 2-edge-coloured graph G is balanced (a problem that is in PTime [5]). More generally, Switching-H-Colouring for 2-edge-coloured graphs H and $k = |V(G)|$ is exactly the problem Signed H-Colouring studied in [5, 6, 16].

Related work. Several works address the parameterized complexity of graph colouring problems. In [25], the vertex-deletion variant of H-List-Colouring is studied. Graph modification problems for Colouring in specific graph classes and for operations VD and ED are considered, for example in [10] (bipartite graphs, split graphs) and [27] (comparability graphs). Graph colouring problems parameterized by structural parameters are considered in [20]. Algorithmic problems relative to the operation of *Seidel switching* have been considered. Given a (classic) graph G, the Seidel switching operation performed at a vertex exchanges all adjacencies and non-adjacencies of v. This can be seen as performing a switching operation in a 2-edge-coloured complete graph, where blue edges are the actual edges of G, and red edges are its non-edges. In [13, 21], the complexity of graph modification problems with respect to the Seidel switching operation and the property of being a member of certain graph classes has been studied. Our work on Switching-H-Colouring problems can be seen as a variation of these problems, generalised to arbitrary 2-edge-coloured graphs.

Our results. We study the classic and parameterized complexities of the three problems Vertex Deletion-H-Colouring, Edge Deletion-H-Colouring and Switching-H-Colouring. Our focus is on t-edge-coloured graphs H of order at most 2 with t an integer ($t = 2$ for Switching-H-Colouring). Despite having just two vertices, H-Colouring for such H is interesting and nontrivial; it is proved to be in PTime by two different nontrivial methods, see [1, 4]. Thus, the three considered problems are in XP for such H. (Recall that for suitable 1-edge-coloured graphs H of order 1 or 2, Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring include Vertex Cover and Odd Cycle Transversal.)

We completely classify the classical complexity of Vertex Deletion-H-Colouring when H is a t-edge-coloured graph of arbitrary order: it is either trivially in PTime or NP-complete. It turns out that all Vertex Deletion-H-Colouring problems are FPT when H has order at most 2. To prove this, we extend a method from [4] and reduce the problem to an FPT variant of 2-Sat.

For Edge Deletion-H-Colouring, a classical complexity dichotomy seems more difficult to obtain, as there are nontrivial PTime cases. We perform such a classification when H is a t-edge-coloured graph of order at most 2. Similar 2-Sat-based arguments as for Vertex Deletion-H-Colouring give a FPT algorithm for Edge Deletion-H-Colouring when H has order at most 2.

For Switching-H-Colouring when H is a 2-edge-coloured graph, the classical dichotomy is again more difficult to obtain. We perform such a classification by using some characteristics of the switch operation and by giving some reductions to well-known NP-complete problems. In contrast to the two previous cases for the parameterized complexity, we show that for three graphs H of order 2, Switching-H-Colouring is already W[1]-hard

Table 1 Overview of our main results, sorted by problem and by type of classification.

Problem	Vertex-Del.-H-Col.	Edge-Del.-H-Col.	Switching-H-Col.
P vs NP	Dichotomy for all graphs (Cor. 7)	Dichotomy when $\|V(H)\| \leq 2$ (Th. 8)	Dichotomy when $\|V(H)\| \leq 2$ (Th. 9)
FPT vs W-hard when $\|V(H)\| \leq 2$	All FPT (Th. 12)	All FPT (Th. 12)	Dichotomy (Th. 13 and 14)

(and cannot be solved in time $f(k)|G|^{o(k)}$ for any function f, assuming the ETH[1]). For all other 2-edge-coloured graphs of order 2, we prove that Switching-H-Colouring is FPT. Table 1 presents a brief overview of our results.

Our paper is structured as follows. In Section 2, we state some definitions and make some preliminary observations in relation with the literature. In Section 3, we study the classical complexity of the three considered problems. We address their parameterized complexity in Section 4. Finally, we conclude in Section 5.

2 Preliminaries and known results

2.1 Some known complexity dichotomies

Recall that whenever H-Colouring is NP-complete, Vertex Deletion-H-Colouring, Edge Deletion-H-Colouring and Switching-H-Colouring are NP-complete (even for $k = 0$), and thus are not in XP, unless PTime = NP. For example, this is the case when H is a monochromatic triangle. When Signed H-Colouring (this is Switching-H-Colouring for $k = |V(G)|$, see [5]) is NP-complete, then Switching-H-Colouring is NP-complete (but could still be in XP or FPT).

On the other hand, when H-Colouring is in PTime, all three problems are in XP for parameter k (by a brute-force algorithm iterating over all k-subsets of vertices of G, performing the operation on these k vertices, and then solving H-Colouring):

▶ **Proposition 1.** *Let H be an edge-coloured graph such that H-Colouring is in PTime. Then, Vertex Deletion-H-Colouring, Edge Deletion-H-Colouring and Switching-H-Colouring can be solved in time $|G|^{O(k)}$.*

When $k = 0$ and H is 1-coloured, we have the following classic theorem.

▶ **Theorem 2** (Hell and Nešetřil [18]). *Let H be a 1-edge-coloured graph. H-Colouring is in PTime if the core of H has at most one edge (H is bipartite or has a loop), and NP-complete otherwise.*

There is no analogue of Theorem 2 for edge-coloured graphs. In fact, it is proved in [5] that a dichotomy classification for H-Colouring restricted to 2-edge-coloured H would imply a dichotomy for all fixed-target CSP problems. Thus, no simple combinatorial classification is expected to exist. In fact, even for trees, cycles or complete graphs, such classifications are not easy to come by [2]. However, some classifications exist for certain classes of graphs H, such as those of order at most 2 [1, 4] or paths [3].

[1] The Exponential Time Hypothesis, ETH, postulates that 3-SAT cannot be solved in time $2^{o(n)}(n+m)^c$, where n and m are the input's number of variables and clauses, and c is any integer [24].

For SWITCHING-H-COLOURING with $k = |V(G)|$, (that is, SIGNED H-COLOURING), we have the following (where the *switching core* of a 2-edge-coloured graph is a notion of core where an arbitrary number of switchings can be performed before the self-mapping):

▶ **Theorem 3** (Brewster et al. [5, 6]). *Let H be a signed graph.* SIGNED H-COLOURING *is in PTime if the switching core of H has at most two edges, and NP-complete otherwise.*

Note that 2-edge-coloured graphs where the switching core has at most two edges either have one vertex (with zero loop, one loop or two loops of different colours), or two vertices (with either one edge or two parallel edges of different colours joining them) [5]. (If there are two vertices joined by one edge and a loop at one of the vertices, we can switch at the non-loop vertex if necessary to obtain one edge-colour, and then retract the whole graph to the loop-vertex, so this is not a core.)

2.2 Homomorphism dualities and FPT time

For a t-edge-coloured graph H, we say that H has the *duality property* if there is a set $\mathcal{F}(H)$ of t-edge-coloured graphs such that, for any t-edge-coloured graph G, $G \xrightarrow{ec} H$ if and only if no graph F of $\mathcal{F}(H)$ satisfies $F \xrightarrow{ec} G$. If $\mathcal{F}(H)$ is finite, we say that H has the *finite duality property*. If checking whether any graph F in $\mathcal{F}(H)$ satisfies $F \xrightarrow{ec} G$ (for an input edge-coloured graph G) is in PTime, we say that H has the *polynomial duality property*. This is in particular the case when $\mathcal{F}(H)$ is finite. For such H, H-COLOURING is in PTime. This topic is explored in detail for edge-coloured graphs in [1]. By a simple bounded search tree argument, we get the following:

▶ **Proposition 4.** *Let H be an edge-coloured graph with the finite duality property. Then,* VERTEX DELETION-H-COLOURING, EDGE DELETION-H-COLOURING *and* SWITCHING-H-COLOURING *are FPT.*

Proof. First, we search for all appearances of homomorphic images of graphs in $\mathcal{F}(H)$ (there are at most $f(\mathcal{F}(H))$ such images for some exponential function f), which we call *obstructions*. This takes time at most $f(\mathcal{F}(H))n^{m_v}$, where $m_v = \max\{|V(F)|, F \in \mathcal{F}(H)\}$. Then, we need to get rid of each obstruction. For VERTEX DELETION-H-COLOURING (resp. EDGE DELETION-H-COLOURING), we need to delete at least one vertex (resp. edge) in each obstruction, thus we can branch on all m_v (resp. m_v^2) possibilities. For SWITCHING-H-COLOURING, we need to switch at least one of the vertices of the obstruction (but then update the list of obstructions, as we may have created a new one). In all cases, this gives a search tree of height k and degree bounded by a function of $\mathcal{F}(H)$, which is FPT. ◀

3 PTime/NP-complete complexity dichotomies

In this section, we prove some results about the classical complexity of VERTEX DELETION-H-COLOURING, EDGE DELETION-H-COLOURING and SWITCHING-H-COLOURING. We first adapt a general method from [23] to show that VERTEX DELETION-H-COLOURING is either trivial, or NP-complete in Section 3.1.

For EDGE DELETION-H-COLOURING and SWITCHING-H-COLOURING, we cannot use this technique (in fact there exist nontrivial PTime cases). Thus, we turn our attention to edge-coloured graphs of order 2 (note that for every edge-coloured graph H of order at most 2, H-COLOURING is in PTime [1, 4]). Recall that SWITCHING-H-COLOURING is defined only on 2-edge-coloured graphs, so our focus is on this case (but for EDGE DELETION-H-COLOURING our results hold for any number of colours). In Section 3.2, we prove a dichotomy result for graphs of order at most 2 for the Edge Deletion-H-Colouring problem. The Switching-H-Colouring problem is treated in Section 3.3, where we also prove a dichotomy result.

The twelve 2-edge-coloured graphs of order at most 2 that are cores (up to symmetries of the colours) are depicted in Figure 1. The two colours are red (dashed edges) and blue (solid edges). We use the terminology of [1]: for $\alpha \in \{-, r, b, rb\}$, the 2-edge-coloured graph H^1_α is the graph of order 1 with no loop, a red loop, a blue loop, and both kinds of loops, respectively. Similarly, for $\alpha \in \{-, r, b, rb\}$ and $\beta, \gamma \in \{-, r, b\}$, the graph $H^{2\alpha}_{\beta,\gamma}$ denotes the graph of order 2 with vertex set $\{0, 1\}$. The string α indicates the presence of an edge between 0 and 1: no edge, a red edge, a blue edge and both edges for $-$, r, b and rb, respectively. Similarly, β and γ denote the presence of a loop at vertices 0 and 1, respectively ($-$ for no loop, r for a red loop, b for a blue loop).

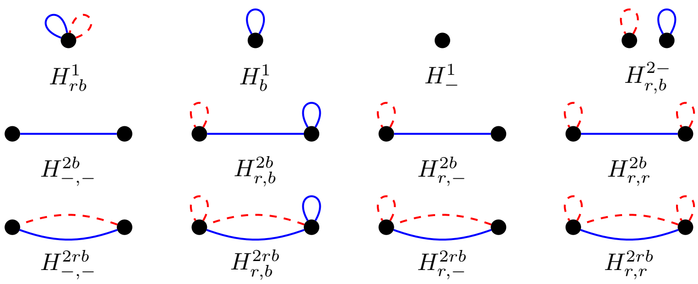


Figure 1 The twelve 2-edge-coloured cores of order at most 2 considered in this paper.

3.1 Dichotomy for Vertex Deletion-H-Colouring

Graph modification problems for operations VD and ED have been studied extensively. For a graph property $\mathcal{P}$, we denote by Vertex Deletion-$\mathcal{P}$ the graph modification problem for property $\mathcal{P}$ and operation VD. Lewis and Yannakakis [23] defined a non-trivial property $\mathcal{P}$ on graphs as a property true for infinitely many graphs and false for infinitely many graphs. They showed the following general result:

▶ **Theorem 5** (Lewis and Yannakakis [23]). *The* Vertex Deletion-$\mathcal{P}$ *problem for nontrivial graph-properties $\mathcal{P}$ that are hereditary on induced subgraphs is NP-hard.*

By modifying the proof of Theorem 5, we can prove the two following results (the proof is omitted due to space restrictions and is included in the full version of the manuscript).

▶ **Theorem 6.** *Let $\mathcal{P}$ be a nontrivial property of loopless edge-coloured graphs that is hereditary for induced subgraphs and true for all independent sets. Then,* Vertex Deletion-$\mathcal{P}$ *is NP-hard.*

For a t-edge-coloured graph, the only case where the property of mapping to H is trivial (in this case, always true) is when H has a vertex with all t kinds of loops attached (in which case the core of H is that vertex). Thus we obtain the following dichotomy.

▶ **Corollary 7.** *Let H be a t-edge-coloured graph.* Vertex Deletion-H-Colouring *is in PTime if H contains a vertex having all t kinds of loops, and NP-complete otherwise.*

3.2 Dichotomy for Edge Deletion-H-Colouring when H has order 2

No analogue of Theorem 5 for operation ED exists nor is expected to exist [28]. We thus restrict our attention to the case of edge-coloured graphs H of order at most 2. For this case we classify the complexity of Edge Deletion-H-Colouring. Since multiple edges of the same colour are irrelevant, if H has order 2, for each edge-colour there are three possible edges.

▶ **Theorem 8.** *Let H be an edge-coloured core of order at most 2. If each colour of H contains only loops or contains all three possible edges, then* Edge Deletion-H-Colouring *is in PTime; otherwise it is NP-complete.*

Proof. The NP-completeness proofs are by reductions from Vertex Cover, based on vertex- and edge-gadgets constructed using obstructions to the corresponding homomorphisms from [1]. They are available in the full version of the manuscript. We now present the PTime part.

First note that if colour i has all three possible edges in H, we can simply ignore this colour by removing it from H and G without decreasing the parameter, as it does not provide any constraint on the homomorphisms.

We can therefore suppose that H contains only loops. If two colours induce the same subgraph of H, then we can identify these two colours in both G and H as they give the same constraints.

If G has colours that H does not have, then remove each edge with this colour and decrease the parameter for each removed edge. If it goes below zero then we reject.

We can now assume that H has only loops and G has the same colours as H. We are left with only a few cases, as H is a core (there is no vertex whose set of loops is included in the set of loops of the other).

- H has a single loop. Then, $G \xrightarrow{ec} H$ as G has the same colours as H.
- H contains two non-incident loops with different colours and two non-incident loops of a third colour. Up to symmetry, suppose that H has one blue loop and one green loop on the first vertex and has one red loop and one green loop on the second vertex. We will reduce to the problem where we have removed the green loops. We construct G' from G by replacing each green edge by a blue edge and a red edge. We claim that Edge Deletion-H-Colouring with parameter k and input G is true if and only if Edge Deletion-$H^{2-}_{r,b}$-Colouring with parameter k plus the number of green edges of G on input G' is true. (See full version of the article for details.) Using this method we can reduce the problem to Edge Deletion-$H^{2-}_{r,b}$-Colouring, which is our last case.
- H contains two non-incident loops with different colours; then $H = H^{2-}_{r,b}$. Indeed if there were any other kind of loop, then we would be in the previous case or we could identify two colours. Note that a 2-edge-coloured graph maps to $H^{2-}_{r,b}$ if and only if it has no red edge incident to a blue edge. Thus, solving Edge Deletion-$H^{2-}_{r,b}$-Colouring amounts to splitting G into disconnecting red and blue connected components. This can be done by constructing the following bipartite graph: put a vertex for each edge of G; two are adjacent if the corresponding edges in G are adjacent and of different colours. Solving Edge Deletion-$H^{2-}_{r,b}$-Colouring is the same as solving Vertex Cover on this bipartite graph, which is PTime.

There is no other case as otherwise the set of loops of one vertex would be included in the set of loops of the other. ◀

3.3 Dichotomy for Switching-H-Colouring when H has order 2

▶ **Theorem 9.** *Let H be a 2-edge-coloured graph from Figure 1. If H is one of $H^{2b}_{r,b}$, $H^{2b}_{r,-}$, $H^{2rb}_{r,b}$, $H^{2rb}_{r,-}$ or $H^{2rb}_{r,r}$, then* Switching-H-Colouring *is* NP*-complete. Otherwise, it is in* PTime*.*

Proof. We begin with the PTime cases.

- Every 2-edge-coloured graph maps to H^1_{rb}, thus Switching-H^1_{rb}-Colouring is trivially in PTime.
- No graph with an edge can be mapped to H^1_- (regardless of switchings).
- For H^1_b, we need to test if the graph can be switched to an all-blue graph in less than k switchings. There are only two sets of switchings that achieve this signature (one is the complement of the other). It is in PTime to test if the graph can be switched to an all-blue graph (see [5, Proposition 2.1]). Doing that also gives us one of the two switching sets; we then need to check if its size is at most k or at least $|V(G)| - k$. So, Switching-H^1_b-Colouring is in PTime.
- For $H^{2-}_{r,b}$, we just apply the algorithm for H^1_b and H^1_r to each connected component, one of the two must accept for each of them.
- For $H^{2br}_{-,-}$, a graph G is a YES-instance if and only if G (without considering edge-colours) is bipartite, which is PTime testable.
- For $H^{2b}_{-,-}$ a graph G is a YES-instance if and only if it is bipartite and maps to H^1_b. We just need to check the two properties, which are both PTime.
- For $H^{2b}_{r,r}$, a graph G maps to $H^{2b}_{r,r}$ if and only if it has no cycles with an odd number of blue edges [1]. This property is preserved under the switching operation. Thus, switching the graph does not impact the nature of the instance. It is thus in PTime (we can test with $k = 0$) since $H^{2b}_{r,r}$-Colouring is in PTime [1, 4].

We now consider the NP-complete cases. For every H, Switching-H-Colouring clearly lies in NP. NP-hardness follows from the above-stated Theorem 3 (proved in [5, 6]) in all but one case: indeed, the switching cores of $H^{2b}_{r,b}$, $H^{2rb}_{r,b}$, $H^{2rb}_{r,-}$ and $H^{2rb}_{r,r}$ have at least three edges, and thus when H is one of these, Switching-H-Colouring is NP-complete (even with $k = |V(G)|$).

The last case is $H^{2b}_{r,-}$. We give a reduction from Vertex Cover to Switching-$H^{2b}_{r,-}$-Colouring. Given instance G of Vertex Cover, we construct an all-red copy G' of G, and we attach to each vertex v of G a blue edge vv', with a red loop on v' (see Figure 2).

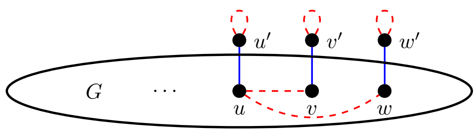


Figure 2 Reduction from Vertex Cover to Switching-$H^{2b}_{r,-}$-Colouring.

Denote by x the vertex of $H^{2b}_{r,-}$ with a loop, and by y the other one. Assume that G has a vertex cover C of size at most k. Denote by G'' the graph obtained from G' by switching at the vertices of C. We map every vertex v' to x, every vertex of C to x and the remaining ones to y. Since C is a vertex cover, each red edge of G'' is either a loop on some vertex v', an edge vv' with $v \in C$ or an edge uv with $u, v \in C$. In each case, both endpoints are mapped on x. The blue edges of G'' are then either vv' with $v \notin C$ or uv with $u \in C$ and $v \notin C$. In both cases, the two endpoints are mapped to different vertices of $H^{2b}_{r,-}$; thus, $G'' \xrightarrow{ec} H^{2b}_{r,-}$.

Conversely, assume that we can switch G' at vertices from a set S such that the resulting graph G'' maps to $H^{2b}_{r,-}$. Let C be the set of vertices v of G such that v or v' lies in S. Note that C has size at most $|S|$. We claim that C is a vertex cover of G. Assume that there is an edge uv in G with $u, v \notin C$. By construction, $u, u', v, v' \notin S$, so uu', vv' are blue in G'', and uv is red. Thus, u, v have to be mapped to x, and u', v' to y, a contradiction since u' has a incident red loop in G''. Therefore C is a vertex cover of G. ◀

4 Parameterized complexity results

4.1 Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring

For many edge-coloured graphs H of order at most 2, we can show that Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring are FPT by giving ad-hoc reductions to Vertex Cover, Odd Cycle Transversal or a combination of both. However, a more powerful method is to generalise a technique from [4] used to prove that H-Colouring is in PTime by reduction to 2-Sat (see also [2]):

▶ **Theorem 10** (Brewster et al. [4]). *Let H be an edge-coloured graph of order at most 2. Then, for each instance G of H-Colouring, there exists a PTime computable 2-Sat formula $F(G)$ that is satisfiable if and only if $G \xrightarrow{ec} H$. Thus, H-Colouring is in PTime.*

The formula $F(G)$ from Theorem 10 contains a variable x_v for each vertex v of G, and for each edge uv, a set of clauses that depends on H. The idea is to see the two vertices of H as "true" and "false", and for each edge uv of a certain colour, to express the possible assignments of x_u and x_v based on the edges of that colour that are present in H.

We will show how to generalise this idea to Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring. We will need the following parameterized variant of 2-Sat:

Variable Deletion Almost 2-Sat	**Parameter:** k.
Input: A 2-CNF Boolean formula F, an integer k.	
Question: Is there a set of k variables that can be deleted from F (together with the clauses containing them) so that the resulting formula is satisfiable?	

Variable Deletion Almost 2-Sat and another similar variant, Clause Deletion Almost 2-Sat (where instead of k variables, k clauses may be deleted), are known to be FPT (see [11, Chapter 3.4]). We need to introduce a more general variant, that we call Group Deletion Almost 2-Sat, defined as follows.

Group Deletion Almost 2-Sat	**Parameter:** k.
Input: A 2-CNF Boolean formula F, an integer k, and a partition of the clauses of F into groups such that each group has a variable which is present in all of its clauses.	
Question: Is there a set of k groups of clauses that can be deleted from F so that the resulting formula is satisfiable?	

By a generalisation of [11, Exercise 3.21] for Clause Deletion Almost 2-Sat, we obtain the following complexity result for Group Deletion Almost 2-Sat. Its proof is included in the full version of the paper.

▶ **Proposition 11.** Group Deletion Almost 2-Sat *is FPT.*

We are now able to prove the following theorem.

▶ **Theorem 12.** *For every edge-coloured graph H of order at most 2,* Vertex Deletion-H-Colouring *and* Edge Deletion-H-Colouring *are FPT.*

Proof. For an instance G, k of Vertex Deletion-H-Colouring or Edge Deletion-H-Colouring, we consider the formula $F(G)$ from Theorem 10 (see [4]). In $F(G)$, to each vertex of G corresponds a variable x_v. Deleting v from G when mapping G to H has the same effect as deleting x_v when satisfying $F(G)$. Thus, this is an FPT reduction from Vertex Deletion-H-Colouring to Variable Deletion Almost 2-Sat.

Moreover, each edge uv of G corresponds to one or two clauses of $F(G)$. This naturally defines the groups of Group Deletion Almost 2-Sat by grouping the clauses corresponding to the same edge. Removing an edge is equivalent to remove its corresponding group. To finish, we have to make sure that we can have one variable common to all the clauses of each group. This is the case in the reduction in [4] for every case except when $E_i(H)$ (the set of edges of colour i in H) is just a loop. Assume without loss of generality that the loop is on vertex 1 (the other loop can be treated the same way). Suppose uv has colour i in G; then uv must be mapped to the loop on vertex 1. The original reduction added the clauses $(x_u)(x_v)$; we modify this part and add instead the clauses $(c + x_u)(c + x_v)(\overline{c})$ where c is a new variable. This is now a valid and equivalent instance of Group Deletion Almost 2-Sat, which is FPT by Proposition 11. ◀

4.2 Switching-H-Colouring: FPT cases

We now consider the parameterized complexity of Switching-H-Colouring. By Theorem 9, there are five 2-edge-coloured graphs H of order at most 2 with Switching-H-Colouring NP-complete. We first show that two of them are FPT:

▶ **Theorem 13.** *Switching-$H^{2b}_{r,b}$-Colouring and Switching-$H^{2b}_{r,-}$-Colouring are FPT.*

Proof. The graph $H^{2b}_{r,b}$ has the finite duality property by [1], which implies FPT time for Switching-$H^{2b}_{r,b}$-Colouring by a simple bounded search tree algorithm (Proposition 4).

For the graph $H^{2b}_{r,-}$, the duality set $\mathcal{F}(H)$ discovered in [1] is composed of paths of the form $RB^{2p-1}R$ (where R is a red edge, B a blue edge and $p \geq 1$ is an integer) and of cycles with an odd number of blue edges. As seen before, if the graph G has such a cycle then switching will not remove it, thus we can reject.

If the graph has a $RB^{2p-1}R$ path and is a positive instance, then we claim that we need to switch one of the four vertices of the red edges. Indeed, if we switch only at the vertices inside the blue path (those not incident with one of the red edges) then the parity of the number of blue edges will not change and we will still have some maximal odd blue subpath, the two edges next to the extremities being red. Thus we would still have a $RB^{2q-1}R$ path.

Thus, since we need to switch at one of these four vertices, we branch on this configuration using the classic bounded search tree technique. This is an FPT algorithm. ◀

4.3 Switching-H-Colouring: W[1]-hard cases

The remaining cases, $H^{2rb}_{r,b}$, $H^{2rb}_{r,-}$ and $H^{2rb}_{r,r}$, yield W[1]-hard Switching-H-Colouring problems, even for input graphs of large girth (recall that the *girth* of a graph is the smallest length of one of its cycles, and by the girth of an edge-coloured graph we mean the girth of its underlying uncoloured graph):

▶ **Theorem 14.** *Let $x \in \{r, b, -\}$. Then for any integer $q \geq 3$, the problem Switching-$H^{2br}_{r,x}$-Colouring is W[1]-hard, even for graphs G' with girth at least q. Under the same conditions, Switching-$H^{2br}_{r,x}$-Colouring cannot be solved in time $f(k)|G|^{o(k)}$ for any function f, assuming the ETH.*

We will prove Theorem 14 by three reductions from Multicoloured Independent Set, which is W[1]-complete [15]:

Multicoloured Independent Set	**Parameter:** k.
Input: A graph G, an integer k and a partition of $V(G)$ into k sets $V_1,\ldots,V_k$.	
Question: Is there a set S of exactly k vertices of G, such that each V_i contains exactly one element of S, that forms an independent set of G?	

Our three reductions (one for each possible choice of x) follow the same pattern. In Section 4.3.1, we describe this idea, together with the required properties of the gadgets. In Sections 4.3.2, 4.3.3 and 4.3.4, we show how to construct the gadgets. Since the reduction preserves the parameter and is actually polynomial, the ETH-based lower bound follows.

4.3.1 Generic reduction

Let (G, k) be an instance of Multicoloured Independent Set, and denote by $V_1, \ldots, V_k$ the partition of G. We begin by replacing each V_i by a *partition gadget* G_i. This gadget must have $|V_i|$ special vertices, in order to associate a vertex of G_i to each vertex of V_i. Moreover, G_i must satisfy the following:

($P1$) We do not have $G_i \xrightarrow{ec} H^{2rb}_{r,x}$.

($P2$) If we switch G_i at exactly one vertex v, then the obtained graph maps to $H^{2rb}_{r,x}$ (without switching) if and only if v is one of the special vertices of G_i.

($P3$) G_i has girth at least q.

Let uv be an edge of G. Recall that u and v can be seen as vertices of G'. We then add an *edge gadget* G_{uv} between u and v. This gadget must satisfy the following:

($E1$) Let H be the graph obtained from G_{uv} by switching at a subset S of $\{u, v\}$. Then, $H \xrightarrow{ec} H^{2rb}_{r,x}$ if $S \neq \{u, v\}$.

($E2$) Assume that $u \in V_i$ and $v \in V_j$ and let H be the graph obtained from $G_{uv} \cup G_i \cup G_j$ by switching u and v. Then, we do *not* have $H \xrightarrow{ec} H^{2rb}_{r,x}$.

($E3$) G_e has girth at least q.

($E4$) In G_e, u and v are at distance at least q.

Let G' be the graph obtained from G by replacing each V_i by a partition gadget G_i, and each edge uv by an edge gadget G_{uv} such that for every $u \in V_i$ and v such that uv is an edge, we identify the special vertex u in G_i with the special vertex u in G_{uv}. (Note in particular that every vertex of G is present in G'.)

We say that a set S of vertices of G is *valid* if, when seen in G', it contains at most one special vertex in each edge gadget. We need a last condition about G':

(SP) If, after switching a valid set in G', the obtained graph does not map to $H^{2rb}_{r,x}$, then this is because a partition gadget or an edge gadget does not map to $H^{2rb}_{r,x}$ (that is, each minimal obstruction is entirely contained in an edge gadget or a partition gadget).

With this Property (SP), we can prove that $(G, k) \mapsto (G', k)$ is a valid reduction.

▶ **Proposition 15.** *(G', k) is a positive instance of* Switching-$H^{2rb}_{r,x}$-Colouring *if and only if (G, k) is a positive instance of* Multicoloured Independent Set.

Proof. Assume we can switch at most k vertices of G' such that the obtained graph maps to $H^{2rb}_{r,x}$. Let S be the set of those vertices. We claim that S is a valid set of G'. First note that, due to ($P1$), S must contain at least one vertex in each V_i. This enforces $|S| = k$, thus S contains exactly one vertex v_i in each V_i. By ($P2$), each of these v_i has to be one of the special vertices of G_i. This means that S contains only vertices that are present in G.

We claim that S induces an independent set in G. Assume by contradiction that there is an edge uv in G with $u, v \in S$. Then, by construction, there is an edge gadget whose special vertices are u and v, such that the edge gadget and the two partition gadgets associated with u and v map to $H^{2rb}_{r,x}$ when we switch only at u and v, contradicting $(E2)$. (Note that S does not contain any other vertex of the edge gadget nor any other vertex of the partition gadgets.) Therefore, G has an independent set of size k containing exactly one vertex in each set V_i.

Conversely, assume that G has an independent set S intersecting each V_i at one vertex. Then, we denote by H the graph obtained by switching all vertices of S in G'. By construction, this is a valid set, hence by (SP) every obstruction for mapping to $H^{2rb}_{r,x}$ in H is actually contained in some gadget. However, it cannot be contained in a partition gadget due to $(P2)$, nor in an edge gadget due to $(E1)$. Therefore, we have $H \xrightarrow{ec} H^{2rb}_{r,x}$. ◀

Observe moreover that, due to $(P3)$, $(E3)$ and $(E4)$, G' has girth at least q. Thus to prove Theorem 14 it suffices to construct the gadgets.

4.3.2 Gadgets for $H^{2rb}_{r,r}$

(a) Partition gadget for $V_i = \{x_0, x_1, x_2, x_3\}$. **(b)** Edge gadget for uv.

Figure 3 Partition and edge gadgets in the $H^{2rb}_{r,r}$-reduction when $q = 3$.

We now describe the gadgets for Switching-$H^{2rb}_{r,r}$-Colouring. Note that for every graph G, we have $G \xrightarrow{ec} H^{2rb}_{r,r}$ if and only if it does not contain an all-blue odd cycle.

The partition gadget G_i is an all-blue cycle of length $2q$ if q and $|V_i|$ have the same parity (resp. $2q + 2$ if they do not have the same parity) with a chord of order $|V_i|$ between two antipodal vertices. The special vertices are those on the chord (see Figure 3a).

Property $(P3)$ directly follows from the construction. Moreover, since G_i contains an all-blue odd cycle, we have $(P1)$. If we switch G_i at exactly one vertex, then either this vertex is a special vertex and the obtained graph does not have any all-blue odd cycle (and thus maps to $H^{2rb}_{r,r}$), or it is not a special vertex and there is still an all-blue odd cycle. Therefore, property $(P2)$ also holds.

We now consider the edge gadget. It is formed by an all-blue odd cycle of length $2q + 1$ where two vertices u, v at distance q have been switched (see Figure 3b). These vertices are the special vertices of the gadget. By construction, properties $(E3)$ and $(E4)$ hold. Moreover, consider a set $S \subset \{u, v\}$. The only way for switching the vertices of S to yield a graph containing an all-blue odd cycle is to switch both u and v. This proves $(E1)$. If we switch at both special vertices then we do not have $G_{uv} \xrightarrow{ec} H^{2rb}_{r,r}$, which implies $(E2)$.

It remains to prove Property (SP). Let S be a valid set, and let H be the graph obtained from G' when switching all vertices of S. Assume that H contains an all-blue odd cycle. Since S is valid set, at most one vertex has been switched in each edge gadget. Therefore, no all-blue odd cycle of H can contain an edge from an edge gadget. It is thus contained in some partition gadget, ensuring that (SP) holds.

Figure 4 The edge gadget for uv in the $H^{2rb}_{r,-}$-reduction when $q = 6$.

4.3.3 Gadgets for $H^{2rb}_{r,-}$

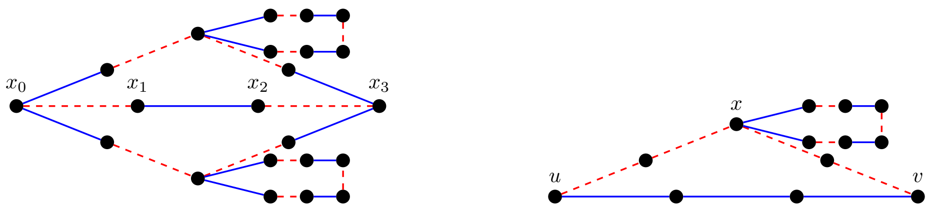
We now describe the gadgets for Switching-$H^{2rb}_{r,-}$-Colouring. Note that for every graph G, we have $G \xrightarrow{ec} H^{2rb}_{r,-}$ if and only if it does not contain a *bad* walk, i.e. a walk v_0, v_1, …, v_{2j}, v_0, v_{2j+2}, …, v_{2p-1}, v_0 such that all edges $v_{2i}v_{2i+1}$ are blue [1].

The partition gadget G_i is the same as in the previous case (see Figure 3a).

The edge gadget is an odd path of length at least q, whose edges are all blue except for the two first and two last ones (see Figure 4).

The proofs of validity for this case can be found in the full version of the paper.

4.3.4 Gadgets for $H^{2rb}_{r,b}$



(a) Partition gadget for $V_i = \{x_0, x_1, x_2, x_3\}$.

(b) Edge gadget for uv. The vertex x is where the two alternating cycles were identified.

Figure 5 Partition and edge gadgets in the $H^{2rb}_{r,b}$-reduction when $q = 3$.

We now describe the gadgets for Switching-$H^{2rb}_{r,b}$-Colouring. Note that for every graph G, we have $G \xrightarrow{ec} H^{2rb}_{r,b}$ if and only if it does not contain another type of *bad* walks, i.e. an alternating walk v_0, v_1, …, v_{2j}, v_0, v_{2j+2}, …, v_{2p-1}, v_0 for some integers j and p [1].

The partition gadget G_i is defined by gluing two obstructions with large girth along a path of length $|V_i|$ (see Figure 5a). More precisely, consider an alternating odd cycle C of size $|V_i| + q$ (or $|V_i| + q + 1$). Note that C contains a vertex u adjacent to two red edges. We attach an alternating odd cycle C' of length q (or $q + 1$) to u, such that the edges of C' adjacent to u are blue. To obtain G_i, we take two copies of this obstruction, and glue their respective largest cycle along a path of length $|V_i|$. The vertices of this path are the special vertices of G_i.

The edge gadget is formed by identifying the vertices with monochromatic neighbourhood of two alternating odd cycles of length $2q + 1$, in such a way that the common vertex has two blue edges in one cycle and two red edges in the other one. To obtain the edge gadget, we switch this graph at two vertices u, v in the same cycle, at distance q from each other (see Figure 5b).

The proofs of validity for this case can be found in the full version of the paper.

5 Conclusion and perspectives

We have introduced Vertex Deletion-H-Colouring, Edge Deletion-H-Colouring and Switching-H-Colouring and characterised their complexity for some small H. The full complexity landscape still needs to be determined. We have fully classified the classic complexity of Vertex Deletion-H-Colouring problems. It remains to do the same for Edge Deletion-H-Colouring and Switching-H-Colouring.

We proved that both Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring are FPT when H has order at most 2. However, if H has order 3, for example if H is a monochromatic triangle, we obtain 3-Colouring, which is not in XP. Switching-H-Colouring seems particularly interesting, since we obtained an FPT/W[1]-hard dichotomy when H has order at most 2 (in which case the problem is always in XP). But again for some H of order 3, Switching-H-Colouring is not in XP. It would be very interesting to obtain FPT/W[1]/XP trichotomies for Vertex Deletion-H-Colouring, Edge Deletion-H-Colouring and Switching-H-Colouring.

Finally, we note that it could be interesting to study analogues of Vertex Deletion-H-Colouring and Edge Deletion-H-Colouring for arbitrary fixed-template CSP problems. Up to our knowledge this has not been done.

References

1 Z. Bawar, R. C. Brewster, and D. A. Marcotte. Homomorphism duality in edge-coloured graphs. *Annales des sciences mathématiques du Québec*, 29(1):21–34, 2005.

2 R. C. Brewster. *Vertex colourings of edge-coloured graphs.* PhD thesis, Simon Fraser University, Canada, 1993.

3 R. C. Brewster. The complexity of colouring symmetric relational systems. *Discrete Applied Mathematics*, 49(1):95–105, 1994.

4 R. C. Brewster, R. Dedić, F. Huard, and J. Queen. The recognition of bound quivers using edge-coloured homomorphisms. *Discrete Mathematics*, 297:13–25, 2005.

5 R. C. Brewster, F. Foucaud, P. Hell, and R. Naserasr. The complexity of signed and edge-coloured graph homomorphisms. *Discrete Mathematics*, 340(2):223–235, 2017.

6 R. C. Brewster and M. H. Siggers. A complexity dichotomy for signed H-colouring. *Discrete Mathematics*, 341(10):2768–2773, 2018.

7 A. A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *IEEE Computer Society*, pages 319–330, FOCS 2017, 2017. Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science.

8 J. Bulín. On the complexity of H-coloring for special oriented trees. *European Journal of Combinatorics*, 69:54–75, 2018.

9 L. Cai. Fixed parameter tractability of graph modification problem for hereditary properties. *Information Processing Letters*, 58:171–176, 1996.

10 L. Cai. Parameterized complexity of vertex colouring. *Discrete Applied Mathematics*, 127:415–429, 2003.

11 M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms.* Springer, 2015.

12 R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity.* Springer, 2013.

13 A. Ehrenfeucht, J. Hage, T. Harju, and G. Rozenberg. Complexity issues in switching of graphs. In *Lecture Notes in Computer Science*, pages 59–70, Proceedings of the International Workshop on Theory and Application of Graph Transformations, TAGT'98, 1764, 2000.

14 T. Feder and M. Y. Vardi. The computational structure of monotone monadic snp and constraint satisfaction: a study through datalog and group theory. *SIAM Journal on Computing*, 28(1):57–104, 1998.

15 M. R. Fellows, D. Hermelin, F. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 40(1):53–61, 2009.

16 F. Foucaud and R. Naserasr. The complexity of homomorphisms of signed graphs and signed constraint satisfaction. In *Lecture Notes in Computer Science*, pages 526–537, Proceedings of the 11th Latin American Symposium on Theoretical Informatics 2014, LATIN'14. 8392, 2014.

17 F. Harary. On the notion of balance of a signed graph. *Michigan Mathematical Journal*, 2(2):143–146, 1953-1954.

18 P. Hell and J. Nešetřil. On the complexity of H-coloring. *Journal of Combinatorial Theory Series B*, 48(1):92–110, 1990.

19 F. Hüffner, N. Betzler, and R. Niedermeier. Separator-based data reduction for signed graph balancing. *Journal of Combinatorial Optimization*, 20(4):335–360, 2010.

20 L. Jaffke and B. M. P. Jansen. Fine-grained parameterized complexity analysis of graph coloring problems. In *Lecture Notes in Computer Science*, pages 345–356, Proceedings of the 10th International Conference on Algorithms and Complexity, CIAC'17. 10236, 2017.

21 E. Jelínková, O. Suchý, P. Hliněný, and J. Kratochvíl. Parameterized problems related to Seidel's switching. *Discrete Mathematics and Theoretical Computer Science*, 13(2):19–42, 2011.

22 S. Khot and V. Raman. Parameterized complexity of finding subgraphs with hereditary properties. *Theoretical Computer Science*, 289(2):997–1008, 2002.

23 J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980.

24 D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–71, 2011.

25 D. Marx. Parameterized coloring problems on chordal graphs. *Theoretical Computer Science*, 351(3):407–424, 2006.

26 R. Naserasr, E. Rollová, and É. Sopena. Homomorphisms of signed graphs. *Journal of Graph Theory*, 79(3):178–212, 2015.

27 Y. Takenaga and K. Higashide. Vertex coloring of comparability $+ke$ and $-ke$ graphs. In *Lecture Notes in Computer Science*, pages 102–112, Proceedings of the 32nd International Worksop on Graph-Theoretic Concepts in Computer Science, WG'06. 4271, 2006.

28 M. Yannakakis. Edge-deletion problems. *SIAM Journal on Computing*, 10(2):297–309, 1981.

29 T. Zaslavsky. Signed graphs. *Discrete Applied Mathematics*, 4(1):47–74, 1982.

30 D. Zhuk. A proof of CSP dichotomy conjecture. In *IEEE Computer Society*, pages 331–342, FOCS 2017, 2017. Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science.