Algorithmica (2024) 86:825–851
https://doi.org/10.1007/s00453-023-01141-z

Reducing Graph Parameters by Contractions and Deletions

Felicia Lucke[1] · **Felix Mann**[1]

Received: 7 October 2022 / Accepted: 6 June 2023 / Published online: 9 July 2023

Abstract
We consider the following problem: for a given graph G and two integers k and d, can we apply a fixed graph operation at most k times in order to reduce a given graph parameter π by at least d? We show that this problem is NP-hard when the parameter is the independence number and the graph operation is vertex deletion or edge contraction, even for fixed $d = 1$ and when restricted to chordal graphs. We give a polynomial time algorithm for bipartite graphs when the operation is edge contraction, the parameter is the independence number and d is fixed. Further, we complete the complexity dichotomy for H-free graphs when the parameter is the clique number and the operation is edge contraction by showing that this problem is NP-hard in $(C_3 + P_1)$-free graphs even for fixed $d = 1$. When the operation is edge deletion and the parameter is the chromatic number, we determine the computational complexity of the associated problem for cographs and complete multipartite graphs. Our results answer several open questions stated in Diner et al. (Theor Comput Sci 746:49–72, 2012, https://doi.org/10.1016/j.tcs.2018.06.023).

Keywords Blocker problems · Edge contraction · Vertex deletion · Edge deletion · Chromatic number · Independence number · Clique number

1 Introduction

Blocker problems are a type of graph modification problems which are characterised by a set $\mathcal{O}$ of graph modification operations (for example, vertex deletion or edge contraction), a graph parameter π and an integer threshold $d \geq 1$. The aim of the problem is to determine, for a given graph G, the smallest sequence of operations from $\mathcal{O}$ which transforms G into a graph G' such that $\pi(G') \leq \pi(G) - d$.

✉ Felix Mann
felix.mann@unifr.ch

Felicia Lucke
felicia.lucke@unifr.ch

1 Université de Fribourg, Boulevard de Pérolles 90, 1700 Fribourg, Switzerland

As in the case of regular graph modification problems, we often consider a set of operations consisting each of a single graph operation, typically vertex deletion, edge contraction, edge addition or edge deletion. Amongst the parameters which have been studied are the chromatic number χ (see [16]), the matching number μ (see [18]), the length of a longest path (see [5, 14]), the (total or semitotal) domination number γ (γ_t and γ_{t2}, respectively) (see [9–11]), the clique number ω (see [15]) and the independence number α (see [3]).

In this paper, the set of allowed graph operations will always consist of only one operation, either *vertex deletion*, *edge contraction* or *edge deletion*. Given a graph G, we denote by $G - U$ the graph from which a subset of vertices $U \subseteq V(G)$ has been deleted. Given an edge $uv \in E(G)$, contracting the edge uv means deleting the vertices u and v and replacing them with a single new vertex which is adjacent to every neighbour of u or v. We denote by G/S the graph in which every edge from an edge set $S \subseteq E(G)$ has been contracted. Further, we denote by $G - S$ the graph G from which a subset of edges $S \subseteq E(G)$ has been deleted. We consider the following problems, where $d \geq 1$ is a fixed integer.

d-Deletion Blocker (π)	
Instance:	A graph G and an integer k.
Question:	Is there a set $U \subseteq V(G)$, $\|U\| \leq k$, such that $\pi(G-U) \leq \pi(G) - d$?

d-Contraction Blocker (π)	
Instance:	A graph G and an integer k.
Question:	Is there a set $S \subseteq E(G)$, $\|S\| \leq k$, such that $\pi(G/S) \leq \pi(G) - d$?

d-Edge Deletion Blocker (π)	
Instance:	A graph G and an integer k.
Question:	Is there a set $S \subseteq E(G)$, $\|S\| \leq k$, such that $\pi(G-S) \leq \pi(G) - d$?

When d is not fixed but part of the input, the problems are called Deletion Blocker(π), Contraction Blocker(π) and Edge Deletion Blocker(π), respectively.

When $\pi = \alpha$ or $\pi = \omega$, we know from [8] that Deletion Blocker(π) and Contraction Blocker(π) are NP-hard for general graphs. From [2] we know that Edge Deletion Blocker(χ) is NP-hard for general graphs. So it is natural to ask if these problems remain NP-hard when the input is restricted to special graph classes.

The authors of [8] show that Contraction Blocker(α) in bipartite and chordal graphs as well as Deletion Blocker(α) in chordal graphs are NP-hard when the threshold d is part of the input. However, as an open question, they ask for the complexity of both problems when d is fixed. In this paper, we show that Contraction Blocker(α) in bipartite graphs is solvable in polynomial time if d is fixed and that both problems are NP-hard for chordal graphs even if $d = 1$. An overview of the complexities in some graph classes is given in Table 1.

A *monogenic* graph class is characterised by a single forbidden induced subgraph H. For a given graph parameter π, it is interesting to establish a *complexity dichotomy for monogenic graph classes*, that is, to determine the complexity of (d-)Deletion

Table 1 The table of complexities for some graph classes

Class	Contraction Blocker(π)		Deletion Blocker(π)	
	$\pi = \alpha$	$\pi = \omega$	$\pi = \alpha$	$\pi = \omega$
Tree	P	P	P	P
Bipartite	NP-h **d fixed: P**	P	P	P
Cobipartite	$d = 1$: NP-c	NP-c d fixed: P	P	P
Cograph	P	P	P	P
Split	NP-c d fixed: P	NP-c d fixed: P	NP-c d fixed: P	NP-c d fixed: P
Interval	?	P	?	P
Chordal	**d=1: NP-c**	$d = 1$: NP-c	**d=1: NP-c**	$d = 1$: NP-c
Perfect	$d = 1$: NP-h	$d = 1$: NP-h	**d=1: NP-c**	$d = 1$: NP-c

Here, P means solvable in polynomial time, whereas NP-h and NP-c mean NP-hard and NP-complete, respectively. A question mark means that the case is open. Everything in **bold** are new results from this paper, all other cases are referenced in [8], where an older version of this table is given

Blocker(π) or (d-)Contraction Blocker(π) in H-free graphs, for every graph H. For example, such a dichotomy has been established for Deletion Blocker(π) for all $\pi \in \{\alpha, \omega, \chi\}$ and Contraction Blocker(π) for $\pi \in \{\alpha, \chi\}$ (all [8]), Contraction Blocker(γ_{t2}) (for $d = k = 1$, [11]), Contraction Blocker(γ_t) (for $d = k = 1$, [9]) and Contraction Blocker(γ) (for $d = k = 1$, [10]). In [8], the computational complexity of Contraction Blocker(ω) in H-free graphs has been determined for every H except $H = C_3 + P_1$. We show that this case is NP-hard even when $d = 1$ and complete hence the dichotomy. For the problem Edge Deletion Blocker(χ), the authors of [8] observe that the complexity of the problem is known for H-free graphs for all H except $H = P_4$ and $H = P_2 + P_1$. We show that Edge Deletion Blocker(χ) is NP-complete for ($P_2 + P_1$)-free graphs (and thus, for P_4-free graphs as well). For P_4-free graphs, we show that the problem is solvable in polynomial time when the difference between d and the chromatic number of the input graph is bounded. We also solve d-Edge Deletion Blocker(χ) for P_4-free graphs in polynomial time for any fixed d.

2 Preliminaries

Throughout this paper, we assume that all graphs are connected unless stated differently.

We refer the reader to [7] for any terminology not defined here.

For any natural number n, we denote by $[n]$ the set $\{1, \ldots, n\}$ and by $[0..n]$ the set $\{0, \ldots, n\}$. For a graph G, we denote by $V(G)$ the *vertex set* of the graph and by $E(G)$ its *edge set*. For two graphs G and H, we denote by $G + H$ the *disjoint union* of G and H. For two graphs G and H with disjoint vertex sets, we denote by $G \times H$ the graph with vertex set $V(G) \cup V(H)$ and edge set $E(G) \cup E(H) \cup \{uv | u \in V(G), v \in V(H)\}$. For two vertices $u, v \in V(G)$, we denote by $\text{dist}_G(u, v)$ the *distance* between u and v, which is the number of edges in a shortest path between u and v. For two sets of vertices $U, W \subseteq V(G)$, the *distance between* U *and* W, denoted by $\text{dist}_G(U, W)$, is given by $\min_{u \in U, w \in W} \text{dist}_G(u, w)$. For a set of edges $S \subseteq E(G)$, we denote by $V(S)$ the set of vertices in $V(G)$ which are endpoints of at least one edge of S. Let $v \in V(G)$, then the *(open) neighbourhood of* v, denoted by $N_G(v)$, is the set $\{u \in V(G) : \text{dist}_G(u, v) = 1\}$. The *closed neighbourhood of* v, denoted by $N_G[v]$, is the set $N_G(v) \cup \{v\}$. For a set $U \subseteq V(G)$, we define the (open) neighbourhood of U as $N_G(U) = \bigcup_{v \in U} N(v)$ and the closed neighbourhood of U as $N_G[U] = N_G(U) \cup U$. If the graph G is clear from the context, we can omit the index. For a vertex $v \in V(G)$ and a set of vertices $U \subseteq V(G)$, we say that v *is complete to* U if v is adjacent to every vertex of U. Let G be a graph and $S \subseteq E(G)$. We denote by $G\big|_S$ the graph whose vertex set is $V(G)$ and whose edge set is S. For any $U \subseteq V(G)$, we denote by $G[U]$ the subgraph of G induced by U. For any $U \subseteq V(G)$, we denote by $G - U$ the graph $G[V(G) \setminus U]$. For any vertex $v \in V(G)$, we denote by $G - v$ the graph $G - \{v\}$.

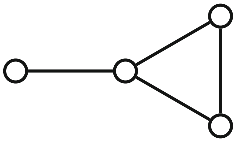
Let $S \subseteq E(G)$. We denote by $G - S$ the graph with vertex set $V(G)$ and edge set $E(G) \setminus S$. Further, we denote by G/S the graph whose vertices are in one-to-one correspondence to the connected components of $G\big|_S$ and two vertices $u, v \in V(G/S)$ are adjacent if and only if their corresponding connected components A, B of $G\big|_S$ satisfy $\text{dist}_G(V(A), V(B)) = 1$. This is equivalent to the regular notion of contracting the edges in S. However, this definition allows us to make the notation in the proofs simpler and less confusing. Let $S, S' \subseteq E(G)$ such that for every connected component A of $G\big|_S$, there is a connected component A' of $G\big|_{S'}$ with $V(A) = V(A')$. Then, $G/S = G/S'$ and thus we get the following corollary, which we will use later.

Corollary 1 *Let* G *be a graph and* $S \subseteq E(G)$ *a minimal* π*-contraction-critical set of edges. Then,* $G\big|_S$ *is a forest.*

An *h-colouring* of G is a map from $V(G)$ to $[h]$. For an h-colouring c of G and a set $U \subseteq V(G)$, we denote by $c(U)$ the set $\bigcup_{v \in U} c(v)$.

We say that a set $I \subseteq V(G)$ is *independent* if the vertices contained in it are pairwise non-adjacent. We denote by $\alpha(G)$ the size of a maximum independent set in G. The decision problem Independent Set takes as input a graph G and an integer k and outputs Yes if and only if there is an independent set of size at least k in G. We say that a set $U \subseteq V(G)$ is a *clique* if every two vertices in U are adjacent. We denote by $\omega(G)$ the size of a maximum clique in G. We call a set $U \subseteq V(G)$ a *vertex cover*, if, for every edge $uv \in E(G)$, we have that $u \in U$ or $v \in U$. The decision problem Vertex Cover takes as input a graph G and an integer k and outputs Yes if and only if there is a vertex cover of size at most k in G. We denote by $\tau(G)$ the size of a minimum vertex cover in G. Furthermore, we call a graph M a *matching* of a graph G, if $V(M) \subseteq V(G)$, $E(M) \subseteq E(G)$ and each vertex in $V(M)$ has exactly one neighbour in M. We say that a matching is a *maximum matching* if it contains the maximum possible number of edges and denote this number by $\mu(G)$. Observe that we did not use the standard definition of a matching as a set of non-adjacent edges. This was done in order to simplify the notation in the proofs. However, the edge set of a matching in our definition follows the conventional definition.

Fig. 1 The paw



A graph without cycles is called a *forest* and a connected forest is a *tree*. It is well-known that a tree has one more vertex than it has edges. Let T be a tree. We call a vertex of T a *leaf* if it has exactly one neighbour. A vertex which is not a leaf is called an *interior* vertex of T. A tree T is called *rooted* if there is a designated vertex called the *root* of T. The children of a vertex $v \in T$ are those neighbours of v whose distance to the root s is larger than $\text{dist}(v, s)$. A *rooted binary tree* is a rooted tree in which every interior vertex has exactly two children. A graph is said to be *chordal*, if it has no induced cycle of length at least four. A graph G is *bipartite*, if we can find a partition of the vertices into two sets $V(G) = U \cup W$ such that U and W are both independent sets. For a given graph H, we say that the graph G is H*-free* if it does not contain H as an induced subgraph.

A graph G is called *complete multipartite* if we can partition the vertex set $V(G)$ into disjoint independent sets $I_1, \dots, I_\ell$ and G is isomorphic to $I_1 \times \cdots \times I_\ell$. We call the independent sets $I_1, \dots, I_\ell$ the *parts* of G. It is easy to see that complete multipartite graphs are exactly the $(P_2 + P_1)$-free graphs.

A graph G is called a *cograph* if one of the following conditions holds:

- $G = K_1$,
- there are cographs H, H' such that $G = H + H'$, or
- there are cographs H, H' such that $G = H \times H'$.

A graph is a cograph if and only if it is P_4-free (see [6]).

Let T be a rooted binary tree with root s whose interior (or non-leaf) vertices are labelled either 0 or 1. We call the vertices of T *nodes* and the interior vertices *0-* or *1-node*, according to their label. To every node p of T we associate a cograph T_p as follows:

- if p is a leaf, then $T_p = K_1$,
- if p is a 0-node with children q and r, then $T_p = T_q + T_r$,
- if p is a 1-node with children q and r, then $T_p = T_q \times T_r$.

If T_s is isomorphic to a cograph G, then we say that T is a *cotree* corresponding to G. For a node $p \in V(T)$, we denote by $T_{\overline{p}}$ the vertex set $V(G) \setminus V(T_p)$. It was shown in [4] that every cograph has a corresponding cotree. Let p be a node of a cograph T with children q and r. It is easy to see that $\chi(T_p) = \max\left\{\chi(T_q), \chi(T_r)\right\}$ if p is a 0-node and $\chi(T_p) = \chi(T_q) + \chi(T_r)$ if p is a 1-node.

For a positive integer i, we denote by P_i and C_i the *path* and the *cycle* on i vertices, respectively. We call the graph which is given in Fig. 1 a *paw*.

For a given graph parameter π, we say that a set $S \subseteq E(G)$ is π*-contraction-critical* if $\pi(G/S) < \pi(G)$. We say that a set $U \subseteq V(G)$ is π*-deletion-critical* if $\pi(G - U) < \pi(G)$.

We will use the following two results. The first one is due to Kőnig, the second one is well-known and easy to see.

Lemma 1 (see [7]) *Let G be a bipartite graph. Then, $\mu(G) = \tau(G)$.*

Lemma 2 *Let G be a graph and let $I \subseteq V(G)$ be a maximum independent set. Then, $V(G) \setminus I$ is a minimum vertex cover and hence $\tau(G) + \alpha(G) = |V(G)|$.*

In [17] it was shown that INDEPENDENT SET is NP-complete in C_3-free graphs. This and Lemma 2 imply the following corollary.

Corollary 2 VERTEX COVER *is NP-complete in C_3-free graphs.*

3 Edge Contractions

3.1 Algorithms

In this section we give a polynomial-time algorithm for d-CONTRACTION BLOCKER(α) in bipartite graphs. In Theorem 1, we consider the case of a graph with at least $2d+2$ vertices where we are allowed to contract at least $2d+1$ edges. We show that this case always leads to a YES-instance. In Theorem 2, we consider the remaining cases, in which the graph has few vertices or we contract only few edges. We will see that we can solve these cases with brute force, by contracting each allowed subset of the edge set and computing the independence number of the resulting graph. We have to be careful here because the contraction can lead to odd cycles, so the resulting graph is not necessarily bipartite. We will use the fact that most of the resulting graph will still be bipartite.

Algorithm 1

Input: A bipartite graph G, a maximum matching M in G, an integer $d \geq 1$
Output: A tree T
Choose an arbitrary edge $uu' \in E(M)$.
Set $V(T) = \{u, u'\}$, $E(T) = \{uu'\}$.
while $|E(T)| \leq 2d - 1$ **do**
    Choose two vertices $w \in N_G(T) \setminus V(T)$, and $w' \in N_G(w) \cap V(T)$.
    if $w \in V(M)$ **then**
        Let $v \in V(M)$ s.t. $vw \in E(M)$.
        $V(T) = V(T) \cup \{v, w\}$, $E(T) = E(T) \cup \{w'w, vw\}$
    else $V(T) = V(T) \cup \{w\}$, $E(T) = E(T) \cup \{w'w\}$
    end if
end while
return T

Theorem 1 *Let G be a connected, bipartite graph with $|V(G)| \geq 2d+2$, where $d \geq 1$ is an integer. Then $(G, 2d+1)$ is a* YES*-instance of d-*CONTRACTION BLOCKER*(α).*

Proof Let G be a bipartite graph with $|V(G)| \geq 2d+2$. Let M be a maximum matching of G. Since G is connected, M is non-empty. Note that G has an independent set of size at least $d+1$ since it is bipartite and $|V(G)| \geq 2d+2$. Consider Algorithm 1,

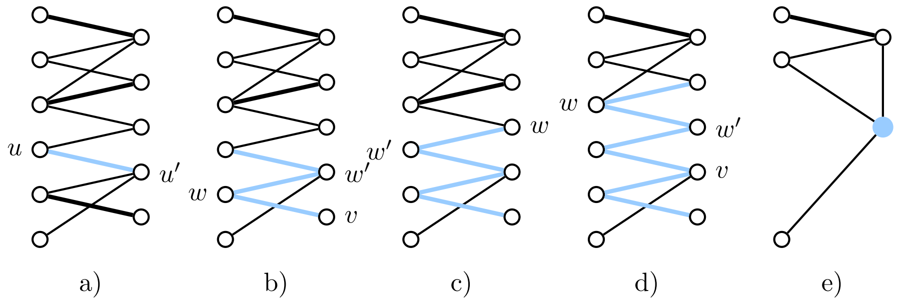


Fig. 2 An example of Algorithm 1 for bipartite graphs, when $d = 3$ and we are allowed to contract at least $2d + 1$ edges. In this example the algorithm does four steps, which are shown in **a**–**d**. In **a** the four thick edges form a maximum matching of the graph. The algorithm starts with one edge uu' (see **a**) and ends when there are $2d = 6$ edges in T (see **d**). The graph resulting from the contraction of T is shown in **e**. We can see that the resulting graph is not bipartite anymore

which constructs a tree T, which is a subgraph of G. An example of Algorithm 1 is given in Fig. 2.

We claim that the resulting graph T is a tree. Indeed, the initial graph is a single edge and thus a tree. Observe that every time there are vertices and edges added to T in lines 7 or 8, the resulting graph remains connected. Further, the number of added vertices and added edges is the same. It follows that T is connected and has exactly one more vertex than it has edges and is thus a tree. It is easy to see that T has $2d$ or $2d + 1$ edges.

We consider the graph $G' = G - V(T)$. For every $v \in V(M) \cap V(T)$, the unique vertex $u \in V(M)$ with $uv \in E(M)$ is also contained in $V(T)$ and $uv \in E(T)$. Thus, there are at most $\left\lfloor \frac{|V(T)|}{2} \right\rfloor$ edges in $E(M)$ which have an endvertex in T. Since $M - V(T)$ is a matching in G' we have that $\mu(G') \geq \mu(G) - \left\lfloor \frac{|V(T)|}{2} \right\rfloor$. Applying Lemma 1 and Lemma 2, we get for the independence number of G':

$$\alpha(G') = |V(G')| - \mu(G') \leq |V(G)| - |V(T)| - \mu(G) + \left\lfloor \frac{|V(T)|}{2} \right\rfloor$$
$$= \alpha(G) - \left\lceil \frac{|V(T)|}{2} \right\rceil = \alpha(G) - d - 1.$$

Let $G^* = G/E(T)$. Observe that $G\big|_{E(T)}$ contains exactly one connected component, say A, which has more than one vertex, namely the connected component corresponding to T. Let $v^* \in V(G^*)$ be the vertex which corresponds to A. Since $G^* - v^*$ is isomorphic to G', we obtain that $\alpha(G^*) \leq \alpha(G') + 1 \leq \alpha(G) - d$. □

Theorem 2 *d-*Contraction Blocker*(α) is solvable in polynomial time in bipartite graphs.*

Proof Let G be a bipartite graph and k a positive integer. If $|V(G)| \leq 2d + 1$ there are at most $2^{d(d+1)}$ subsets of $E(G)$ and at most 2^{2d+1} subsets of $V(G)$. We can check for every subset $S \subseteq E(G)$ if $\alpha(G/S) \leq \alpha(G) - d$ in constant time by computing the graph G/S and checking for each subset of $V(G/S)$ if it is independent. Thus, we can check in constant time if G is a YES-instance for d-CONTRACTION BLOCKER(α).

In the following we may assume that $|V(G)| \geq 2d+2$. By Theorem 1, we know that for $k \geq 2d+1$, we can reduce the independence number by at least d by contracting at most k edges. Thus, we can further assume that $k \leq 2d$.

Algorithm 2

Input: A bipartite graph G, an integer k, a fixed integer d
Output: YES if (G, k) is a YES-instance of d-CONTRACTION BLOCKER(α),
NO if not
2: **for** every $S \subseteq E(G)$ of size at most k **do**
Let $\beta = 0$.
4: Let $G' = G/S$.
Let $U = \{v \in V(G') : v$ *corresponds to a connected component of* $G|_S$ *which contains at least 2 vertices*$\}$.
6: **for** every subset $U' \subseteq U$ **do**
if U' is independent **then**
8: $\beta = \max(\beta, \alpha(G' - (U \cup N_{G'}(U'))) + |U'|)$
end if
10: **end for**
if $\beta \leq \alpha(G) - d$ **then**
12: **return** YES
end if
14: **end for**
return NO

Consider now Algorithm 2 which takes as input G, k and d and outputs YES or NO. Algorithm 2 considers every subset $S \subseteq E(G)$ of edges of cardinality at most k and computes $\alpha(G/S)$. If there is some S such that $\alpha(G/S) \leq \alpha(G) - d$ then we return YES, and NO otherwise. In order to compute $\alpha(G/S)$ for such a subset S of edges, we first set $G' = G/S$. We then consider the set of vertices $U \subseteq V(G')$ which have been formed by contracting some edges in S (see line 4 of the algorithm). Observe that $G[V(G') \setminus U]$ is isomorphic to $G - V(S)$ and induces thus a bipartite graph. Every independent set of G' can be partitioned into a set $U' \subseteq U$ and a set $W \subseteq V(G') \setminus (U \cup N_{G'}(U'))$. Thus, we can find the independence number of G' by considering every independent subset U' of U and computing $\alpha(G' - (U \cup N_{G'}(U'))) + |U'|$. The largest of these values is then $\alpha(G')$. The independence number of the bipartite graph $G' - (U \cup N_{G'}(U'))$ can be computed in polynomial time, see Lemma 2 and [1].

The number of subsets of $E(G)$ of cardinality at most k is in $O(|E(G)|^k) = O(|V(G)|^{4d})$. For any such subset S, the number of subsets $U' \subseteq U$ is at most $2^k \leq 2^{2d}$. Thus, the running time of Algorithm 2 is polynomial. □

3.2 Hardness proofs

In this section, we answer several questions asked in [8]. Indeed, Theorem 3 settles the missing case of [8, Theorem 24] and completes the complexity dichotomy for H-free graphs, which is given after the theorem. We further settle the computational complexity of 1-Contraction Blocker(α) in chordal graphs, which is an open case in Table 1.

Theorem 3 *The decision problem* 1- Contraction Blocker*(ω) is NP-hard in* $(C_3 + P_1)$*-free graphs.*

Proof We use a reduction from Vertex Cover in C_3-free graphs which is NP-complete due to Corollary 2. Let (G, k) be an instance of Vertex Cover where G is a C_3-free graph. Since Vertex Cover is trivial to solve for a graph without edges, we can assume that $E(G)$ is non-empty. We construct an instance (G', k) of 1- Contraction Blocker(ω) such that (G, k) is a Yes-instance of Vertex Cover if and only if (G', k) is a Yes-instance of 1- Contraction Blocker(ω) and G' is $(C_3 + P_1)$-free. Let G' be a graph with $V(G') = V(G) \cup \{w\}$, $w \notin V(G)$, and $E(G') = E(G) \cup \{wv, v \in V(G)\}$. In other words, we add a universal vertex w to G in order to obtain G'. See Fig. 3 for an example.

Since G is C_3-free, every copy of C_3 in G' has to contain w. Furthermore, since w is adjacent to every other vertex in $V(G')$, it follows that every vertex of G' has distance at most one to every copy of C_3. Thus, G' is $(C_3 + P_1)$-free. Also, note that $\omega(G') = 3$ and that every maximum clique in G' is a copy of C_3 which contains w and exactly two vertices of $V(G)$.

Let us assume that (G, k) is a Yes-instance of Vertex Cover. Let $\{v_1, \dots, v_k\} \subseteq V(G)$ be a vertex cover of G. Set $S = \{v_i w \colon i \in \{1, \dots, k\}\}$ and let $G^* = G'/S$. We claim that S is ω-contraction-critical. Notice that the contraction of an edge $vw \in S$ is equivalent to deleting the vertex v, since the new vertex remains adjacent to all other vertices. Thus, G^* is isomorphic to $G - (V(S) \setminus \{w\})$. Since $\{v_1, \dots, v_k\}$ is a minimum vertex cover of G, there are no edges in $G^* - w$. This means that G^* is C_3-free and thus $\omega(G^*) \leq 2$. Hence (G', k) is a Yes-instance of 1- Contraction Blocker(ω).

For the other direction, assume that (G', k) is a Yes-instance of 1- Contraction Blocker(ω). Let $S \subseteq E(G')$ be a minimum ω-contraction-critical set of edges with $|S| \leq k$ and let $G^* = G'/S$.

We construct a set U of vertices of G as follows: For the connected component T of $G'\big|_S$ that contains w, add every vertex of $V(T)$ except w to U. For every other connected component T of $G'\big|_S$, we add to U all vertices of $V(T)$ except one, which can be chosen arbitrarily. We claim that U is a vertex cover of G of size at most k.

To see that $|U| \leq k$, observe that for every connected component T of $G'\big|_S$, we have added $|V(T)| - 1$ vertices to U. Since T is a tree (see Corollary 1), we have that $|V(T)| - 1 = |E(T)|$. Thus, we have added as many vertices to U as there are edges in S and hence $|U| = |S| \leq k$.

In order to show that U is a vertex cover, suppose for a contradiction that there is an edge $uv \in E(G)$ for which neither u nor v is contained in U. Consider the connected components A_u, A_v and A_w of $G'\big|_S$ which contain u, v and w, respectively. It follows from the construction of U that in every connected component T of $G'\big|_S$ there is at most one vertex of T which is not contained in U. Hence, $A_u \neq A_v$. We have that $w \notin U$ by construction, so the same argument can be used to show that $A_u \neq A_w$ and $A_v \neq A_w$. Thus, A_u, A_v, A_w correspond to three different vertices in G^*. Since the

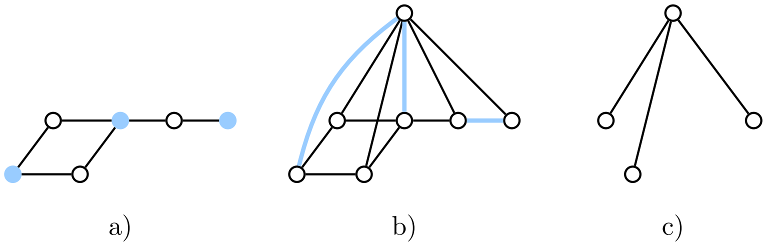


Fig. 3 **a** shows a graph G, where a minimum vertex cover is highlighted with blue vertices. In **b** the graph G' constructed in Theorem 3 for G is shown. The thick blue edges are a 1-contraction blocker of G'. Finally, **c** shows the graph G' after the contraction of the edges in the 1-contraction blocker

components are pairwise at distance one, their corresponding vertices induce a C_3 in G^*, a contradiction to S being ω-contraction-critical. Thus, U is a vertex cover in G and (G, k) a YES-instance of VERTEX COVER. □

Theorem 4 *Let H be a graph. If H is an induced subgraph of P_4 or of the paw, then* CONTRACTION BLOCKER(ω) *is polynomial-time solvable for H-free graphs. Otherwise, it is NP-hard or co-NP-hard for H-free graphs.*

In order to simplify the notation of the proof of the following theorem, we restate VERTEX COVER as a satisfiability problem.

WEIGHTED POSITIVE 2- SAT	
Instance:	A variable set X, a clause set C in which all clauses contain exactly two literals and every literal is positive, as well as an integer k.
Question:	Is there a truth assignment of the variables (that is, a mapping $f: X \to \{\text{true}, \text{false}\}$) such that at least one literal in each clause is true and there are at most k variables which are true.

If $\Phi = (G, k)$ is an instance of VERTEX COVER then taking $X = V(G)$ as the variable set and $C = \{(u \vee w): uw \in E(G)\}$ as the set of clauses yields an instance (X, C, k) of WEIGHTED POSITIVE 2- SAT which is clearly equivalent to Φ. Since VERTEX COVER is known to be NP-hard (see Corollary 2), it follows that WEIGHTED POSITIVE 2- SAT is NP-hard, too.

Theorem 5 1-CONTRACTION BLOCKER*(α) is* NP*-complete in chordal graphs.*

Proof It was shown in [13] that INDEPENDENT SET can be solved in polynomial time for chordal graphs. Since the family of chordal graphs is closed under edge contractions, for a given chordal graph G and a set $S \subseteq E(G)$, it is possible to check in polynomial time whether S is α-contraction-critical. It follows that 1-CONTRACTION BLOCKER(α) is in NP for chordal graphs. In order to show NP-hardness, we reduce from WEIGHTED POSITIVE 2- SAT, which was shown to be NP-hard above. Let $\Phi = (X, C, k)$ be an instance of WEIGHTED POSITIVE 2- SAT. We construct a

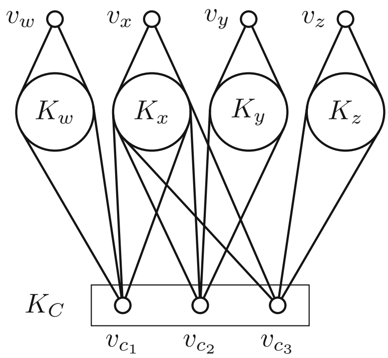


Fig. 4 This is the graph corresponding to the instance of WEIGHTED POSITIVE 2- SAT given by the variables w, x, y, z and the clauses $c_1 = w \vee x$, $c_2 = x \vee y$ and $c_3 = x \vee z$. The rectangular box corresponds to $G[K_C]$, the vertices contained in it induce a clique. Every set K_i induces a clique and the lines between a vertex and a set K_i mean that this vertex is complete to K_i

chordal graph G such that (G, k) is a YES-instance for 1-CONTRACTION BLOCKER(α) if and only if Φ is a YES-instance for WEIGHTED POSITIVE 2- SAT, as follows:

For every variable $x \in X$, we introduce a set of vertices G_x with $G_x = \{v_x\} \cup K_x$. Here, K_x is a set of $2k + 1$ vertices which induce a clique. We make v_x complete to K_x. For every clause $c \in C$, we introduce a vertex v_c. We define $K_C = \bigcup_{c \in C} \{v_c\}$. We add edges so that $G[K_C]$ is a clique. For every clause $c \in C$, $c = (x \vee y)$, we make v_c complete to K_x and K_y (see Fig. 4 for an example).

Observe first that the graph G is indeed chordal: if a cycle of length at least four contains at least three vertices of K_C, it follows immediately that the cycle cannot be induced, since K_C induces a clique. Otherwise, such a cycle contains at most two vertices of K_C. Assume that there are two vertices w and w' of the cycle which are contained in G_x and G_y, respectively, with $x, y \in X$, $x \neq y$. Then, the cycle has to contain a chord in $G[K_C]$ and is thus not induced. If all vertices of the cycle are in $K_C \cup G_x$ for some fixed $x \in X$, then there are at least two vertices w and w' contained in K_x. Hence, the cycle cannot be induced since w and w' are adjacent and have the same neighbourhood. It follows that G cannot have any induced cycle of length at least 4 and is thus chordal.

Since G_x induces a clique for every $x \in X$, it can contain at most one vertex in any independent set; the same applies to K_C. Thus, $\alpha(G) \leq |X| + 1$. Let $c \in C$. Since the set $\{v_x : x \in X\} \cup \{v_c\}$ is an independent set of size $|X| + 1$, it follows that $\alpha(G) = |X| + 1$.

Let us assume that Φ is a YES-instance of WEIGHTED POSITIVE 2- SAT. Let X_+ be the set of positive variables of a satisfying assignment of Φ. For each $x \in X_+$, let e_x be an edge incident to v_x and let $S = \{e_x | x \in X_+\}$. Let $G' = G/S$. We claim that $\alpha(G') < \alpha(G)$. To see this, observe first that for any $x \in X_+$, contracting e_x is equivalent to deleting the vertex v_x, since $N_G(v_x) = K_x$ induces a clique. Therefore, we have that $G' \simeq G - \{v_x : x \in X_+\}$. Suppose for a contradiction that there is an independent set I of G' of size $|X| + 1$. Since $|I \cap K_x| \leq 1$ (for $x \in X_+$) and $|I \cap G_x| \leq 1$ (for all $x \in X \setminus X_+$), it follows that there exists $c \in C$ such that $v_c \in K_C \cap I$. Furthermore, the inequalities above all have to be equalities. By the choice of X_+, it follows that there is $x \in X_+$ such that x is a literal in c. Since $|I \cap K_x| = 1$, there is a vertex $w \in I \cap K_x$ which is adjacent to v_c, contradicting the fact that I is independent. It follows that S is α-contraction-critical.

For the other direction, assume that $\Phi' = (G, k)$ is a YES-instance of 1-CONTRACTION BLOCKER(α). Let S be a minimum α-contraction-critical set of edges such that $|S| \leq k$. By Corollary 1, the graph $G\big|_S$ is a forest.

For any $x \in X$, there is a vertex $u_x \in K_x \setminus V(S)$. This follows from the fact that k edges can be incident to at most $2k$ vertices and $|K_x| = 2k + 1$. Let H be the graph with vertex set $V(H) = K_C$ and edge set $E(H) = \{uv \in S : u, v \in K_C\}$.

We will show that for every connected component of H, there is a variable $x \in X$ with $\text{dist}_G(G_x, V(T)) = 1$ and $G_x \cap V(S) \neq \emptyset$. Suppose for a contradiction that there is a connected component T of H such that for every $x \in X$ with $\text{dist}_G(G_x, V(T)) = 1$, we have $G_x \cap V(S) = \emptyset$. In other words, for every $c = (x \vee y) \in C$ with $v_c \in V(T)$, we have $G_x \cap V(S) = G_y \cap V(S) = \emptyset$. So we have that $N_G[V(T)] \cap V(S) \subseteq V(T)$, and thus T is also a connected component in $G\big|_S$. For every $x \in X$, the set $\{u_x\}$ is a connected component in $G\big|_S$, that is, u_x is not incident to any edge in S. Further, for every $x \in X$ where $\text{dist}_G(G_x, V(T)) = 1$, we have that $G_x \cap V(S) = \emptyset$. Thus, $\{v_x\}$ is a connected component in $G\big|_S$. Let $X_1 = \{x \in X\colon \text{dist}_G(u_x, V(T)) = 1\}$ and $X_2 = X \setminus X_1$. Consider the set $I = T \cup \{\{v_x\} : x \in X_1\} \cup \{\{u_x\} : x \in X_2\}$ of connected components of $G\big|_S$. Each two connected components in the set correspond to vertices in G/S who are at distance at least two. In other words, I corresponds to an independent set in G/S of cardinality $|X| + 1$, a contradiction to the assumption that S is α-contraction-critical. It follows that there is no connected component T of H such that for every $x \in X$ with $\text{dist}_G(G_x, V(T)) = 1$, we have $G_x \cap V(S) = \emptyset$.

We can obtain a truth assignment of the variables satisfying Φ as follows: Set every x to true for which $G_x \cap V(S)$ is non-empty. For every clause $c = (x \vee y) \in C$ for which both $G_x \cap V(S)$ and $G_y \cap V(S)$ are empty, set one of its variables to true. This assignment is clearly satisfying, it remains to show that we set at most $|S| \leq k$ variables to true. Consider a connected component T of H. Recall that T is a tree, and so its number of vertices is one more than its number of edges. We have shown that there is a vertex $v_c \in V(T)$, $c = (x \vee y)$, for which $G_x \cap V(S) \neq \emptyset$. Thus, there are at most $|E(T)|$ vertices $v_c \in T$, $c = (x \vee y)$, for which both $G_x \cap V(S)$ and $G_y \cap V(S)$ are empty. This implies that for every connected component T of H, we set at most $|E(T)|$ variables to true. Further, the number of variables $x \in X$ which we set to true because $G_x \cap V(S)$ is non-empty cannot be larger than the number of edges of S which are not contained in $G[K_C]$. This shows that, in total, we set at most $|S|$ variables to true, which concludes the proof. □

4 Vertex Deletions

In this section, we settle another open case of Table 1. Interestingly, 1- DELETION BLOCKER(α) and 1-CONTRACTION BLOCKER(α) are equivalent on the instance Φ' constructed in the proof of Theorem 5 and thus the same construction can be used to show NP-hardness of 1- DELETION BLOCKER(α) in chordal graphs.

Theorem 6 1- DELETION BLOCKER*(α) is* NP*-complete in chordal graphs.*

Proof It has been shown in [13] that it is possible to determine the independence number of chordal graphs in polynomial time. Since chordal graphs are closed under vertex deletion, it is possible to check in polynomial time whether the deletion of a given set of vertices reduces the independence number. Hence 1- DELETION BLOCKER(α) is in NP for chordal graphs.

In order to show NP-hardness, we reduce from WEIGHTED POSITIVE 2- SAT. Let Φ be an instance of WEIGHTED POSITIVE 2- SAT, $\Phi = (X, C, k)$. Let $\Phi' = (G, k)$ be the instance of 1-CONTRACTION BLOCKER(α) which is described in Theorem 5 and which has been shown to be equivalent to Φ. Further, let K_x, G_x and v_x for each $x \in X$, K_C, and v_c for each $c \in C$ be as in the proof of Theorem 5. Recall that we have shown that $\alpha(G) = |X| + 1$ and that G is chordal.

We show that Φ' is a YES-instance of 1- DELETION BLOCKER(α) if and only if Φ is a YES-instance of WEIGHTED POSITIVE 2- SAT.

Assume first that Φ is a YES-instance of WEIGHTED POSITIVE 2- SAT and that X_+ is the set of positive variables in a satisfying assignment of Φ. We have shown in the proof of Theorem 5 that $\alpha(G - \{v_x : x \in X_+\}) < \alpha(G)$. Hence, (G, k) is a YES-instance of 1- DELETION BLOCKER(α).

Conversely, assume that Φ' is a YES-instance of 1- DELETION BLOCKER(α) and let W be an α-deletion-critical set of vertices of cardinality at most k. For every $x \in X$, there exists a vertex $u_x \in K_x \setminus W$, since $|W| < |K_x|$. Define a set $Z = \{x \in X : v_x \in W\}$ and initialize a set $Z' = \varnothing$. For every clause $c \in C$ with $v_c \in W$, we choose one of the variables contained in c and add it to Z'. We claim that setting the variables of $Z \cup Z'$ to true yields a satisfying assignment of Φ with at most k true variables. Observe first that $|Z \cup Z'| \leq |W| \leq k$ by construction. Suppose for a contradiction that there is a clause $c \in C$, $c = (x \vee y)$, such that neither x nor y is contained in $Z \cup Z'$. It follows that $v_x, v_y, v_c \notin W$. But then $\{v_c, v_x, v_y\} \cup \{u_z : z \in X \setminus \{x, y\}\} \subseteq G - W$ is an independent set of size $|X| + 1$, a contradiction to the α-deletion-criticalness of W. Hence, the assignment is satisfying and has at most k true variables, which implies the theorem. □

Since perfect graphs are a superclass of chordal graphs, we obtain the following corollary.

Corollary 3 1- DELETION BLOCKER*(α) is* NP-*complete in perfect graphs.*

Observe that Corollary 3 could also be shown as follows. Complements of perfect graphs are again perfect graphs. Further, 1- DELETION BLOCKER(α) is a YES-instance for a graph G if and only if 1- DELETION BLOCKER(ω) is a YES-instance for $\overline{G}$. Since it was shown in [8] that 1- DELETION BLOCKER(ω) is NP-hard in perfect graphs, the corollary follows.

5 Edge Deletions

Given a colouring c of the vertices of a graph G, we say that an edge uv, with $u, v \in V(G)$, is a *monochromatic edge* of c if $c(u) = c(v)$. Using this terminology, a *proper colouring* is a colouring without monochromatic edges. The following problem is a generalization of the well-known h-CHROMATIC NUMBER, in the sense that we ask if there is a colouring with few monochromatic edges.

h-CHROMATIC NUMBER	
Instance:	A graph G.
Question:	Is there an h-colouring of G without monochromatic edges?

h-MONOCHROMATIC EDGES	
Instance:	A graph G and an integer m.
Question:	Is there an h-colouring of G with at most m monochromatic edges?

As above, we sometimes consider h to be part of the input. The problem is then called MONOCHROMATIC EDGES.

To keep the notation more simple, we will focus on MONOCHROMATIC EDGES instead of EDGE DELETION BLOCKER(χ) in this chapter. This is justified by the following proposition.

Lemma 3 *The tuple* (G, m) *is a* YES*-instance for* h-MONOCHROMATIC EDGES *if and only if* (G, m) *is a* YES*-instance for* $(\chi(G) - h)$-EDGE DELETION BLOCKER(χ).

Proof Let G be a graph and m an integer. If (G, m) is a YES-instance for h-MONOCHROMATIC EDGES, then there is an h-colouring c of G with at most m monochromatic edges. Let $S \subseteq E(G)$ be the set of monochromatic edges of c. Then, c is a proper h-colouring of $G - S$. It follows that deleting $|S| \leq m$ edges from G yields a graph whose chromatic number is at most h. In other words, (G, m) is a YES-instance for $(\chi(G) - h)$-EDGE DELETION BLOCKER(χ). For the other direction, assume that (G, m) is a YES-instance for $(\chi(G) - h)$-EDGE DELETION BLOCKER(χ). Thus, there is a set of edges $S \subseteq E(G)$ such that $|S| \leq m$ and $\chi(G - S) \leq \chi(G) - (\chi(G) - h) = h$. Let c be a proper h-colouring of $G - S$. When we colour the vertices of G according to c, then the only monochromatic edges can be the edges in S. Thus, c is an h-colouring of G with at most m monochromatic edges, which completes the proof. □

The following lemma is a simple observation about reducing the number of monochromatic edges by recolouring the vertices.

Lemma 4 *Let G be a graph, and $I \subseteq V(G)$ an independent set such that for any $v \in I$, we have $N(v) = N(I)$. If c is an h-colouring of G, then there is a colour $j \in c(I)$ such that recolouring every vertex of I with j yields an h-colouring of G which has at most as many monochromatic edges as c.*

Proof For each $i \in [h]$, we denote by n_i the number of vertices in I which receive colour i by c. Similarly, for every $i \in [h]$, we denote by n'_i the number of vertices in $N(I)$ which receive colour i by c. Since I is independent, no colouring can have any monochromatic edges between two vertices of I. The number of monochromatic edges between I and $N(I)$ is $\sum_{i\in[h]} n_i n'_i$. Let $j \in c(I)$ be such that n'_j is minimum amongst all colours in $c(I)$. After recolouring all vertices in I with colour j, the number of monochromatic edges between I and $N(I)$ is $|I|n'_j = \sum_{i\in[h]} n_i n'_j \le \sum_{i\in[h]} n_i n'_i$. This concludes the proof. □

5.1 Algorithms

Theorem 7 *For a fixed integer h, the decision problem h-MONOCHROMATIC EDGES is solvable in polynomial time for cographs.*

Proof Let G be a cograph with associated cotree T. For every $p \in V(T)$, we define a function f^p which takes as input an h-tuple of non-negative integers $a^p = (a_1^p, \ldots, a_h^p)$ whose entries sum up to $|V(T_p)|$. Below, we will give a definition of f^p and show that its output can be computed in polynomial time. Then, we will show the following claim:

Claim. For every $p \in V(T)$ and every h-tuple $a^p = (a_1^p, \ldots, a_h^p)$ of non-negative integers with

$$a_1^p + \ldots + a_h^p = |T_p|, \tag{1}$$

the value of $f^p(a^p)$ is the minimum number of monochromatic edges of all h-colourings of T_p, in which colour i appears a_i^p times, for every $i \in [h]$.

We will now give the definition of f^p. If p is a leaf, then $f^p(a^p) = 0$ for any valid input. If p is not a leaf, let q and r be the children of p. If p is a 0-node, then

$$f^p(a^p) = \min_{\substack{a^q, a^r \in [0..n]^h \\ a_1^q + \ldots + a_h^q = |T_q| \\ a_1^r + \ldots + a_h^r = |T_r| \\ a_i^q + a_i^r = a_i^p,\, i \in [h]}} \left(f^q(a^q) + f^r(a^r) \right).$$

If p is a 1-node, then

$$f^p(a^p) = \min_{\substack{a^q, a^r \in [0..n]^h \\ a_1^q + \ldots + a_h^q = |T_q| \\ a_1^r + \ldots + a_h^r = |T_r| \\ a_i^q + a_i^r = a_i^p,\, i \in [h]}} \left(f^q(a^q) + f^r(a^r) + \sum_{i=1}^{h} a_i^q a_i^r \right).$$

This defines the values of f^p for every $p \in V(T)$. Observe that for every node $p \in V(T)$, there are at most $O(n^h)$ possible inputs a^p. To compute the values of f^p, we consider every pair of h-tuples $a^q, a^r \in [0..n]^h$, of which there are $O(n^{2h})$. Checking whether they sum to the correct values and computing the term given in the formula above takes constant time. So, we can compute the function f^p in polynomial time.

Proof of the claim. Observe first that the claim holds when p is a leaf, since then there are no edges in T_p. If p is not a leaf, let q and r be the children of p and assume that the claim holds for q and r.

Let $a^p = (a_1^p, \ldots, a_h^p)$ be an h-tuple which satisfies (1). Let c_p be an h-colouring of T_p which assigns colour i to exactly a_i^p vertices, for every $i \in [h]$, and which minimizes the number of monochromatic edges amongst all such colourings. Let m_p be the number of monochromatic edges in c_p. Let c_q and c_r be the h-colourings of T_q and T_r, respectively, which we obtain by restricting c_p to T_q and T_r, respectively. For every $i \in [h]$, let a_i^q be the number of vertices in T_q which receive the colour i from c_q. Define a_i^r analogously for all $i \in [h]$. Clearly, $a_i^p = a_i^q + a_i^r$ for every $i \in [h]$, as well as $\sum_{i=1}^h a_i^q = |T_q|$ and $\sum_{i=1}^h a_i^r = |T_r|$.

It follows from the definition of f^p that $f^p(a^p) \le f^q(a^q) + f^r(a^r)$ if p is a 0-node. Analogously, $f^p(a^p) \le f^q(a^q) + f^r(a^r) + \sum_{i=1}^h a_i^q a_i^r$ if p is a 1-node.

Let m_q be the number of monochromatic edges of c_q, and define m_r analogously. Since the claim holds for q and r, it follows that $f^q(a^q) \le m_q$ and $f^r(a^r) \le m_r$. If p is a 0-node then the monochromatic edges of c_p are exactly the disjoint union of the monochromatic edges of c_q and those of c_r. Thus, the number of monochromatic edges of c_p is $m_p = m_q + m_r$ which implies that $f^p(a^p) \le m_q + m_r = m_p$. If p is a 1-node, the monochromatic edges of c_p are the monochromatic edges of c_q, those of c_r and the monochromatic edges between T_q and T_r. For each colour $i \in [h]$, there are $a_i^q a_i^r$ monochromatic edges between T_q and T_r whose endpoints are coloured i. Thus, in total, the number of monochromatic edges of c_p is $m_p = m_q + m_r + \sum_{i=1}^h a_i^q a_i^r$. This implies that $f^p(a^p) \le m_q + m_r + \sum_{i=1}^h a_i^q a_i^r = m_p$.

Suppose for a contradiction that $f^p(a^p) < m_p$. Then there are h-tuples of non-negative integers $b^q = (b_1^q, \ldots, b_h^q)$ and $b^r = (b_1^r, \ldots, b_h^r)$ such that

- $b_1^q + \ldots + b_h^q = |T_q|$,
- $b_1^r + \ldots + b_h^r = |T_r|$,
- $b_i^q + b_i^r = a_i^p$ for all $i \in [h]$,
- $f^q(b^q) + f^r(b^r) = f^p(a^p) < m_p = m_q + m_r$ if p is a 0-node and
- $f^q(b^q) + f^r(b^r) + \sum_{i=1}^h b_i^q b_i^r = f^p(a^p) < m_p = m_q + m_r + \sum_{i=1}^h a_i^q a_i^r$ if p is a 1-node.

We assume that the claim holds for q and r. It follows that there is an h-colouring c'_q of T_q which assigns colour i to b_i^q vertices in T_q, for every $i \in [h]$. There is also an analogously defined function c'_r.

Let us colour T_p by colouring the vertices in T_q according to c'_q and the vertices in T_r according to c'_r. This yields a well-defined h-colouring c'_p of T_p since $V(T_p)$ is a disjoint union of $V(T_q)$ and $V(T_r)$. The number of monochromatic edges of c'_p is $f^q(b^q) + f^r(b^r) = f^p(a^p)$ if p is a 0-node. Similarly, the number of monochromatic edges of c'_p is $f^q(b^q) + f^r(b^r) + \sum_{i=1}^h b_i^q b_i^r = f^p(a^p)$ if p is a 1-node. Since $f^p(a^p) < m_p$, it follows that c'_p is an h-colouring of T_p which assigns the colour i to $b_i^q + b_i^r = a_i^p$ vertices, for every $i \in [m]$. Also, c'_p has less monochromatic edges than c_p, a contradiction. It follows that $f^p(a^p)$ is the number of monochromatic edges of c_p and thus that the claim holds for p.

We have shown that for every node $p \in V(T)$ and every h-tuple a^p, we can compute in polynomial time the smallest number of monochromatic edges of all h-colourings of T_p whose i-th colour class has size a_i^p, for each $i \in [h]$. Let s be the root of T. The value of $f^s(a^s)$ is the minimum number of monochromatic edges of all h-colourings of T whose i-th colour class has size a_i^s for each $i \in [h]$. There are $O(n^h)$ possibilities how to fix the sizes of h colour classes which partition n vertices. We can determine the minimum number of monochromatic edges of h-colourings for each partition and then take the minimum of all of them. This can be done in polynomial time and shows thus the theorem. □

We have shown that we can solve Monochromatic Edges if the number of available colours is bounded. We will now consider the case where the number of available colours is almost the chromatic number of the graph. To be precise, the difference between the available colours and the chromatic number is bounded by a constant. In order to tackle this problem, we will first show that, for any h, there are h-colourings of the graph which minimize the number of monochromatic edges and which have a very specific structure. In particular, for a cotree T of a graph and a node $p \in V(T)$ with children q and r, we will give conditions for colours which appear in both T_q and T_r, depending on whether p is a 0- or a 1-node. We will then show in Lemmas 5 and 6 that there are h-colourings of the graph which minimize the number of monochromatic edges and which satisfy these conditions for every node of the corresponding cotree.

We will need the following definitions. Let ℓ be a positive integer, $a = (a_1, \dots, a_\ell)$ and $a' = (a'_1, \dots, a'_\ell)$ two ℓ-tuples of real numbers and $\lambda \leq \ell$ a positive integer. We want to associate λ entries of one tuple to λ entries of the other tuple.

We say that a *λ-matching μ of two ℓ-tuples* is a λ-tuple $((i_1, i'_1), \dots, (i_\lambda, i'_\lambda)) \in ([\ell] \times [\ell])^\lambda$ such that for all $j, j' \in [\lambda]$, $j \neq j'$, we have that $i_j \neq i_{j'}$ and $i'_j \neq i'_{j'}$. Intuitively, we can imagine that a λ-matching encodes the edges of a matching in a complete bipartite graph whose vertices correspond to the entries of two ℓ-tuples, see Fig. 5 for an example. For any $\mu_j = (i_j, i'_j)$, $j \in [\lambda]$, we say that the index i_j is matched with the index i'_j. For the ℓ-tuples a and a', we say that the entry a_{i_j} is matched with the entry $a'_{i'_j}$. The *value of a λ-matching μ applied to the ℓ-tuples a and a'* is denoted by $\mathrm{val}(\mu, a, a')$ and is defined as $\sum_{j=1}^{\lambda} a_{i_j} a'_{i'_j}$.

We now want to create a new tuple from a and a' and a λ-matching μ. A *μ-merge* of a and a' is a $(2\ell - \lambda)$-tuple whose j-th entry is $a_{i_j} + a'_{i'_j}$ for all $j \in [\lambda]$. The next $\ell - \lambda$ entries contain the unmatched entries of a, and the last $\ell - \lambda$ entries contain the unmatched entries of a', see Fig. 5 for an example. We obtain a *sorted μ-merge of a and a'* if we sort the entries of any μ-merge of a and a' in ascending order. Observe that there might be several (unsorted) μ-merges of a and a', but there is only one sorted μ-merge. Given a λ-matching μ and two ℓ-tuples a and a', we denote the sorted μ-merge of a and a' by $\mathrm{merge}(\mu, a, a')$. We denote the i-th entry of $\mathrm{merge}(\mu, a, a')$ by $\mathrm{merge}(\mu, a, a', i)$, for every $i \in [2\ell - \lambda]$.

Property 1 Let G be a cograph with corresponding cotree T. A colouring c of T is said to have *Property* 1 if, for every 0-node p of T with children q and r, the i-th

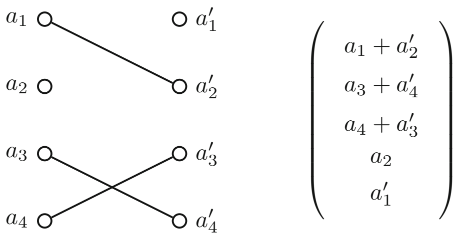


Fig. 5 Left: The 3-matching $\mu = ((1, 2), (3, 4), (4, 3))$ visualized on the 4-tuples $a = (a_1, \dots, a_4)$ and $a' = (a_1', \dots, a_4')$. Right: A corresponding μ-merge

largest colour class of T_q has the same colour as the i-th largest colour class of T_r for all $i \in [\chi(T_p)]$, unless one of them is empty.

Lemma 5 *Let G be a cograph with associated cotree T. Let d be a non-negative integer. There is a $(\chi(G) - d)$-colouring c of G with the minimum possible number of monochromatic edges which satisfies Property 1.*

Proof Let $p \in V(T)$ be a 0-node with children q and r. Suppose that there is an $i \in [\chi(T_p)]$ such that the i-th largest colour class of T_q and the i-th largest colour class of T_r are both non-empty and have different colours. Assume further that i is the smallest such number. By construction of the cotree, all vertices in T_p have the same neighbourhood in $T_{\overline{p}}$. Denote by n_q and n_r the number of vertices in $T_{\overline{p}}$ which have the same colour as the vertices in the i-th largest colour class of T_q and T_r, respectively, and which are adjacent to the vertices in T_p. Assume, without loss of generality, that $n_q \geq n_r$. Denote by a_q and a_r the number of vertices of the i-th largest colour class of T_q and T_r, respectively. Denote by a_q' the number of vertices in T_q which have the colour of the i-th largest colour class of T_r. Denote by a_r' the number of vertices in T_r which have the colour of the i-th largest colour class of T_q. Observe that $a_q' \leq a_q$ and $a_r' \leq a_r$, since we assumed that i is the smallest index for which the i-th largest colour classes of T_q and T_r do not have the same colour. There are $n_q(a_q + a_r') + n_r(a_r + a_q')$ monochromatic edges between T_p and $T_{\overline{p}}$ whose endpoints have the colours of the i-th colour classes of T_q or T_r. Recolour the vertices in the i-th largest colour class of T_q with the colour of the i-th largest colour class of T_r. Similarly, recolour the vertices in T_q which have the colour of the i-th largest colour class in T_r with the colour of the i-th largest colour class of T_q. Observe that we only exchanged the colours of two colour classes in T_q. Thus, we can only have changed the number of monochromatic edges between T_p and $T_{\overline{p}}$ which have the colours of the i-th largest colour classes of T_p and T_q. After recolouring, this number is now $n_r(a_q + a_r) + n_q(a_q' + a_r')$. The difference of the number of monochromatic edges is $n_q(a_q + a_r') + n_r(a_r + a_q') - n_r(a_q + a_r) + n_q(a_q' + a_r') = (n_r - n_q)(a_q' - a_q)$, which is the product of two non-positive numbers and thus non-negative. This shows that the recolouring did not increase the number of monochromatic edges. Repeating this step for each $i \in [\chi(T_p)]$ as above yields a colouring which satisfies Property 1 above.

Observe that if Property 1 holds for any descendant q of p in the original colouring, then it still holds for q after the recolouring, since any two vertices in T_q having the same colour in the original colouring will still have the same colour after the recolouring step. We can thus apply the recolouring procedure starting at the leaves and working our way up to the root. We end up with a colouring of the graph for which Property 1 holds for every node of the cotree. □

Lemma 6 *Let G be a cograph with associated cotree T whose root p is a 1-node with children q and r. Let $a^p = (a_1^p, \ldots, a_\ell^p)$ be an ℓ-tuple of non-negative integers and let λ, Δ be non-negative integers. Assume that there is a $(\chi(T_p) - \Delta)$-colouring c_p of T_p for which the following four conditions hold:*

- *Property 1 holds,*
- *the i-th smallest colour class of c_p has at most a_i^p vertices for every $i \in [\ell]$,*
- *there are exactly λ colours which appear in both T_q and T_r,*
- *c_p has the minimum number of monochromatic edges of all $(\chi(T_p)-\Delta)$-colourings of T_p which satisfy the first three conditions above.*

Then, there is a $(\chi(T_p) - \Delta)$-colouring c'_p of T_p which satisfies all of the four conditions above and this following condition, too:

- *there is no $j > \ell + \lambda$ such that the colour of the j-th smallest colour class of T_q (or T_r) appears in T_r (or T_q, respectively).*

Proof Let c_p be a $(\chi(T_p) - \Delta)$-colouring of T_p which satisfies the first four conditions in the statement of the lemma. If there is no $j > \ell + \lambda$ such that the j-th smallest colour class of T_q shares its colour with some vertices in T_r, or vice versa, we are done. If not, suppose, without loss of generality, that there are $j > \ell + \lambda$ and j' such that the j-th smallest colour class of T_q and the j'-th smallest colour class of T_r have the same colour. Besides the j-th smallest colour class, there are $\lambda - 1$ other colour classes in T_q whose colour also appears in T_r. Since $j \geq \ell + \lambda + 1$, there is an $i \in [\ell + \lambda]$ such that the i-th smallest colour class in T_q has a colour which does not appear in T_r and which is not one of the ℓ smallest colour classes of T_p. Recolour the vertices of the i-th and the j-th smallest colour classes of c_q by exchanging their colours. This does not change the number of monochromatic edges with both ends in T_q. The number of monochromatic edges between T_q and T_r cannot increase since the i-th smallest colour class does not contain more vertices than the j-th smallest one. It follows that the new colouring did not increase the number of monochromatic edges or the number of colours used. Further, it did not change the sizes of the ℓ smallest colour classes. It is also clear that the new colouring still has Property 1. Repeating this process for every j as above yields a colouring as desired. □

Theorem 8 *For a fixed integer d, $(\chi(G) - d)$-*Monochromatic Edges *is solvable in polynomial time for cographs.*

Proof Let G be a cograph with associated cotree T and let $d \leq \chi(G)$ be a fixed integer. For every $\ell \in [0..d]$ and every node $p \in V(T)$, we define a function f_ℓ^p which takes as input an ℓ-tuple of non-negative integers $a^p = (a_1^p, \ldots, a_\ell^p)$ with $a_i^p \leq a_j^p$, for every $i \leq j$, and a non-negative integer Δ with $\Delta \leq d - \ell$.

Below, we will give a definition of f_l^p and show that its output can be computed in polynomial time. Then, we will show the following claim:

Claim. Let $\ell \in [d]$, $p \in V(T)$ be such that (a^p, Δ) is a valid input of f_ℓ^p. Then, $f_\ell^p(a^p, \Delta)$ is the smallest number of monochromatic edges of all $(\chi(T_p) - \Delta)$-colourings of T_p which have Property 1 and whose i-th smallest colour class has size at most a_i^p, for all $i \in [\ell]$.

Observe that the correctness of the claim would show the theorem: Indeed, let s be the root of T and () an empty tuple of length 0. The correctness of the claim implies that $f_0^s((), d)$ is the smallest number of monochromatic edges that any $(\chi(G) - d)$-colouring of G satisfying Property 1 can have. It follows then from Lemma 5 that this coincides with the smallest number of monochromatic edges amongst *all* $(\chi(G) - d)$-colourings of G.

We will now give the definition of f_l^p. If p is a leaf, then f_ℓ^p will output 0 for every ℓ and every input. If p is not a leaf, let q and r be the children of p. Let ℓ and Δ be non-negative integers with $\ell + \Delta \leq d$ and let $a^p = (a_1^p, \ldots, a_\ell^p)$ be an ℓ-tuple as above.

If p is a 0-node, then assume, without loss of generality, that $\chi(T_q) \geq \chi(T_r)$ and let $\delta = \chi(T_q) - \chi(T_r)$.

We distinguish the following three cases:

Case 1: If $\Delta \geq \delta$ we set

$$f_\ell^p(a^p, \Delta) = \min_{\substack{a^q, a^r \in [0..n]^\ell \\ a_i^q + a_i^r \leq a_i^p, i \in [\ell]}} f_\ell^q(a^q, \Delta) + f_\ell^r(a^r, \Delta - \delta).$$

Case 2: If $\Delta < \delta$ and $\Delta + \ell \geq \delta$ we set

$$f_\ell^p(a^p, \Delta) = \min_{\substack{a^q, a^r \in [0..n]^{\ell-\delta+\Delta} \\ a^q_{\delta-\Delta+i} + a_i^r \leq a^p_{\delta-\Delta+i}, i \in [\ell-\delta+\Delta]}} f_\ell^q(a_1^p, \ldots, a_{\delta-\Delta}^p, a^q, \Delta) + f_{\ell-\delta+\Delta}^r(a^r, 0).$$

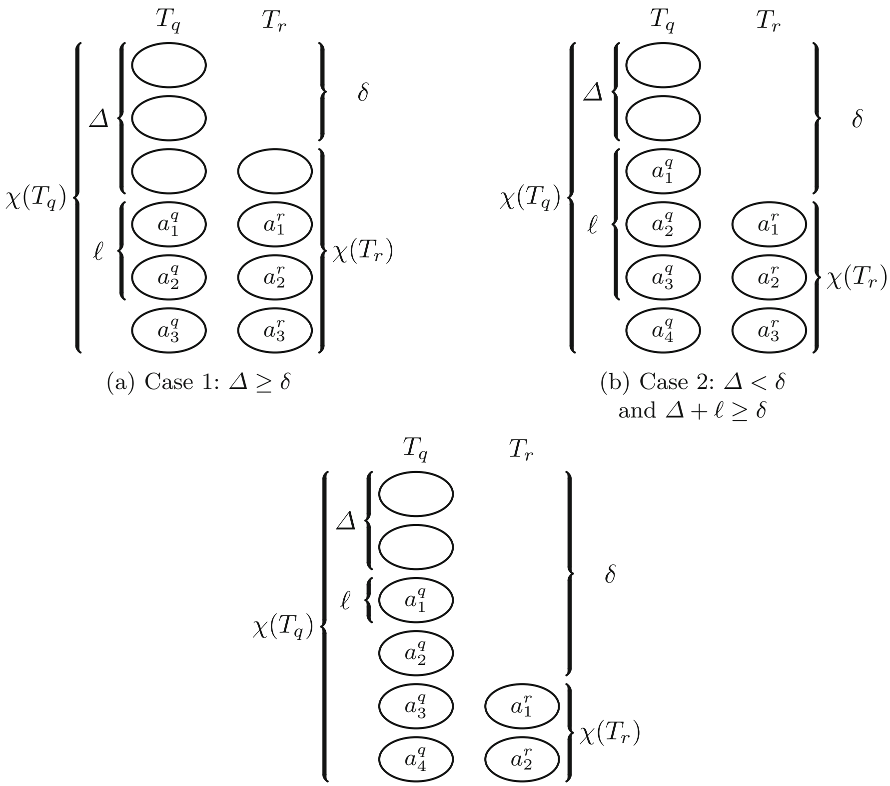
Case 3: If $\Delta < \delta$ and $\Delta + \ell < \delta$ we set

$$f_\ell^p(a^p, \Delta) = f_\ell^q(a^p, \Delta).$$

If p is a 1-node, then

$$f_\ell^p(a^p, \Delta) = \min_{\substack{\Delta_q, \Delta_r, \lambda \geq 0 \\ \Delta_q + \Delta_r + \lambda = \Delta \\ a^q, a^r \in [0..n]^{\ell+\lambda} \\ \mu\lambda\text{-matching of } (\ell+\lambda)\text{-tuples} \\ \mathrm{merge}(\mu, a^q, a^r, i) \leq a_i^p, i \in [\ell]}} f_{\ell+\lambda}^q(a^q, \Delta_q) + f_{\ell+\lambda}^r(a^r, \Delta_r) + \mathrm{val}(\mu, a^q, a^r).$$

This defines the values of f_ℓ^p for every $\ell \in [0..d]$ and $p \in V(T)$. Observe that for every f_ℓ^p, there are at most $(n+1)^\ell(d - \ell + 1)$ possible inputs. In order to compute the value of f_ℓ^p for one of them, we consider at most $(n+1)^{2(\ell+\Delta)} = O(n^{2d})$ possible pairs of tuples a^q and a^r if p is a 0-node. Analogously, there are at most $\Delta^3 n^{2(\ell+\lambda)} \ell^{2\lambda} = O(n^{2d})$ choices for $\Delta_q, \Delta_r, \lambda, a^q, a^r$ and μ if p is a 1-node. Thus, we can compute the function f_ℓ^p in polynomial time.



(a) Case 1: $\Delta \geq \delta$

(b) Case 2: $\Delta < \delta$ and $\Delta + \ell \geq \delta$

(c) Case 3: $\Delta < \delta$ and $\Delta + \ell < \delta$

Fig. 6 Illustration of the relation between δ, Δ and ℓ in the three cases we consider in the proof if p is a 0-node. We represent a $(\chi(T_p) - \Delta)$-colouring with Property 1 as a $\chi(T_p)$-colouring with Property 1 whose smallest Δ colour classes are empty. Each ellipse represents a colour class in T_q or T_r and they are sorted ascendingly in size from top to bottom. By Property 1, two horizontally adjacent ellipses correspond to the same colour class in T_p

It remains to show the correctness of the claim.

Proof of the Claim. Let $p \in V(T)$ and let ℓ and Δ be non-negative integers with $\ell + \Delta \leq d$ and $a^p = (a_1^p, \dots, a_\ell^p) \in [0..n]^\ell$.

Observe first that the value of f_ℓ^p is correct when p is a leaf since then there are no edges in T_p. If p is not a leaf let q and r be the children of p and assume that the claim holds for q and r.

Let c_p be a $(\chi(T_p) - \Delta)$-colouring of T_p which satisfies Property 1, whose i-th smallest colour class has at most a_i^p vertices, for every $i \in [\ell]$, and which has the smallest number of monochromatic edges amongst all such colourings. We need to show that the number of monochromatic edges of c_p, which we denote by m_p, is identical to the value of $f_\ell^p(a^p, \Delta)$. Let c_q and c_r be the colourings which we obtain by restricting c_p to T_q and T_r, respectively. Let Δ_q be an integer such that $\chi(T_q) - \Delta_q$ is the exact number of colours we use in c_q. We define Δ_r analogously. For every

$i \in [\ell]$, denote by a_i^q and a_i^r the number of vertices of the i-th smallest colour class of c_q and c_r, respectively. Observe that the i-th smallest colour classes of c_q and c_r do not necessarily have the same colour. Let m_q and m_r be the numbers of monochromatic edges of c_q and c_r, respectively.

Assume first that p is a 0-node, set $\delta = \chi(T_q) - \chi(T_r)$ and assume without loss of generality that $\delta \geq 0$. Recall that $\chi(T_p) = \chi(T_q)$ and $m_q + m_r = m_p$, since p is a 0-node. It follows from Property 1 that for every $i > \chi(T_r)$, the i-th largest colour class of c_p is entirely contained in T_q. To prove the claim, we distinguish the following three cases, see Fig. 6 for an illustration.

Case 1: $\Delta \geq \delta$

We first show that $f_\ell^p(a^p, \Delta) \leq m_p$. The colouring c_p uses $\chi(T_q) - \Delta$ colours on T_p. We have that $\chi(T_q) - \Delta = \chi(T_r) - (\Delta - \delta) \leq \chi(T_r)$. It follows that c_q is a $(\chi(T_q) - \Delta)$-colouring of T_q and c_r is a $(\chi(T_r) - (\Delta - \delta))$-colouring of T_r. By Property 1 we know that the i-th largest colour class of c_p on T_p consists of the i-th largest colour class of c_q and the i-th largest colour class of c_r, for every $i \in [\chi(T_q) - \Delta]$. This implies in particular that $a_i^p \geq a_i^q + a_i^r$ for every $i \in [\ell]$.

We have that c_q is a $(\chi(T_q) - \Delta)$-colouring of T_q whose i-th smallest colour class contains a_i^q vertices and that the claim holds for q. It follows that $f_\ell^q(a^q, \Delta) \leq m_q$ and analogously $f_\ell^r(a^r, \Delta - \delta) \leq m_r$. Hence

$$f_\ell^p(a^p, \Delta) \leq f_\ell^q(a^q, \Delta) + f_\ell^r(a^r, \Delta - \delta) \leq m_q + m_r = m_p.$$

It remains to show that $f_\ell^p(a^p, \Delta) = m_p$. We suppose for a contradiction that there are $a^q, a^r \in [0..n]^\ell$ such that $a_i^q + a_i^r \leq a_i^p$, for all $i \in [\ell]$, and $f_\ell^p(a^p, \Delta) = f_\ell^q(a^q, \Delta) + f_\ell^r(a^r, \Delta - \delta) < m_p$. Since the claim holds for q and r, it follows that there is a $(\chi(T_q) - \Delta)$-colouring c'_q of T_q and a $(\chi(T_r) - (\Delta - \delta))$-colouring c'_r of T_r, such that:

- c'_q and c'_r both have Property 1,
- the i-th smallest colour class of c'_q (c'_r, respectively) contains at most a_i^q vertices (a_i^r vertices, respectively), for each $i \in [\ell]$,
- the sum of the number of monochromatic edges in c'_q and c'_r is strictly less than m_p.

Construct a $(\chi(T_p) - \Delta)$-colouring c'_p of T_p with Property 1 such that the i-th largest colour class of c'_p contains exactly the vertices of the i-th largest colour class of c'_q and the vertices of the i-th largest colour class of c'_r, for every $i \in [\chi(T_q) - \Delta]$. It follows that the size of the i-th smallest colour class of c'_p is at most a_i^p, for every $i \in [\ell]$. Since p is a 0-node, the number of monochromatic edges of c'_p is exactly the sum of the numbers of monochromatic edges of c'_q and c'_r and thus less than m_p. By construction, c'_p has Property 1. Thus, c'_p is a $(\chi(T_p) - \Delta)$-colouring of T_p which has Property 1, whose i-th smallest colour class has size at most a_i^p, for every $i \in [\ell]$, and which has less monochromatic edges than c_p, a contradiction to the choice of c_p.

Case 2: $\Delta < \delta$ and $\Delta + \ell \geq \delta$

Again we first show that $f_\ell^p(a^p, \Delta) \leq m_p$. Since c_p uses $\chi(T_q) - \Delta$ colours, it follows from Property 1 that the smallest $\chi(T_q) - \Delta - \chi(T_r) = \delta - \Delta$ colour classes of c_p are entirely contained in T_q. Thus, $a_i^q \leq a_i^p$, for each $i \in [\delta - \Delta]$. Further, c_r uses $\chi(T_r)$ colours. For every $i \in \{\delta - \Delta + 1, \dots, \ell\}$, the i-th smallest colour class of c consists of the i-th smallest colour class of c_q and the $(i - \delta + \Delta)$-th smallest colour class of c_r. It follows that $a_i^p \geq a_i^q + a_{i-\delta+\Delta}^r$. This implies that $f_\ell^q(a_1^p, \dots, a_{\delta-\Delta}^p, a_{\delta-\Delta+1}^q, \dots, a_\ell^q, \Delta) \leq f_\ell^q(a^q, \Delta) \leq m_q$ and $f_{\ell-\delta+\Delta}^r(a^r, 0) \leq m_r$. It follows that

$$\begin{aligned} f_\ell^p(a^p, \Delta) &\leq f_\ell^q(a_1^p, \dots, a_{\delta-\Delta}^p, a_{\delta-\Delta+1}^q, \dots, a_\ell^q, \Delta) + f_{\ell-\delta+\Delta}^r(a^r, 0) \\ &\leq m_q + m_r = m_p. \end{aligned}$$

It remains to show that $f_\ell^p(a^p, \Delta) = m_p$. Again, we suppose for a contradiction that there are $a^q, a^r \in [0..n]^{\ell-\delta+\Delta}$ such that $a_{i+\delta-\Delta}^q + a_i^r \leq a_{i+\delta-\Delta}^p$, for all $i \in [\ell-\delta+\Delta]$, and $f_\ell^q(a_1^p, \dots, a_{\delta-\Delta}^p, a^q, \Delta) + f_{\ell-\delta+\Delta}^r(a^r, 0) < m_q + m_r$. Since we assumed that the claim holds for q and r, it follows that there is a $(\chi(T_q) - \Delta)$-colouring c'_q of T_q and a $\chi(T_r)$-colouring c'_r of T_r such that

- c'_q and c'_r both have Property 1,
- the i-th smallest colour class of c'_q contains at most a_i^p vertices, for every $i \in [\delta - \Delta]$,
- the $(i + \delta - \Delta)$-th smallest colour class of c'_q contains at most $a_{i+\delta-\Delta}^q$ vertices, for each $i \in [\ell - \delta + \Delta]$,
- the i-th smallest colour class of c'_r contains at most a_i^r vertices, for each $i \in [\ell - \delta + \Delta]$,
- the sum of monochromatic edges in c'_q and c'_r is less than $m_q + m_r$.

Construct a $(\chi(T_q) - \Delta)$-colouring c'_p of T_p with Property 1 as follows:

- for every $i \in [\delta - \Delta]$, the i-th smallest colour class of c'_p contains exactly the vertices of the i-th smallest colour class of c'_q,
- for every $i \in \{\delta - \Delta + 1, \dots, \chi(T_q) - \Delta\}$, the i-th smallest colour class of c'_p contains exactly the vertices of the i-th smallest colour class of c'_q and the vertices of the $(i - \delta + \Delta)$-th smallest colour class of c'_r.

It follows that the size of the i-th smallest colour class of c'_p has size at most a_i^p, for every $i \in [\ell]$. It is also clear that the number of monochromatic edges of c'_p is exactly the sum of the numbers of monochromatic edges of c'_q and c'_r and thus less than $m_q + m_r$. So c'_p is a $(\chi(T_q) - \Delta)$-colouring of T_p which has Property 1, whose i-th colour class has size at most a_i^p, for every $i \in [\ell]$, and which has less monochromatic edges than c_p, a contradiction to the choice of c_p.

Case 3: $\Delta < \delta$ and $\Delta + \ell < \delta$

In this case, the colours used in the ℓ smallest colour classes of c do not appear in T_r. Clearly, $f_\ell^r(a^r, \Delta) = 0$ and thus $f_\ell^p(a^p, \Delta) \leq f_\ell^q(a^q, \Delta) + f_\ell^r(a^r, \Delta) \leq m_q$.

Suppose for a contradiction that $f_\ell^q(a^q, \Delta) < m_q$. Then, there is a $(\chi(T_q) - \Delta)$-colouring c'_q of T_q with Property 1 such that the i-th smallest colour class of c'_q contains at most a_i^p vertices, for each $i \in [\ell]$, and the number of monochromatic edges of c'_q is less than m_q. Let c'_r be a proper colouring of T_r with Property 1 using $\chi(T_r)$ colours. Construct a $(\chi(T_p) - \Delta)$-colouring c'_p of T_p with Property 1 as follows:

- for each $i \in [\delta - \Delta]$, the i-th smallest colour class of c'_p contains exactly the vertices of the i-th smallest colour class of c'_q,
- for $i \in \{\delta - \Delta + 1, \ldots, \chi(T_r) - \Delta\}$, the i-th smallest colour class of c'_p contains exactly the vertices of the i-th smallest colour class of c'_q and the $(i - \delta + \Delta)$-th smallest colour class of c'_r.

Since $\ell < \delta - \Delta$, it follows that the ℓ smallest colour classes of c'_p are all contained in T_q. Thus, c'_p is a $(\chi(T_p) - \Delta)$-colouring of T_p whose i-th smallest colour class contains at most a_i^p vertices, for each $i \in [\ell]$. Since the number of monochromatic edges of c'_p is equal to the number of monochromatic edges of c'_q, it follows that c'_p has less monochromatic edges than c_p, a contradiction. This completes the third case and thus the claim holds for p if p is a 0-node.

Assume that p is a 1-node. Let λ be the number of colours that are used by both c_q and c_r. Following Lemma 6 we can assume that only the $\ell + \lambda$ smallest colour classes have colours which appear in both T_q and T_r. The number of colours used by c_p is $\chi(T_q) - \Delta_q + \chi(T_r) - \Delta_r - \lambda = \chi(T_p) - (\Delta_q + \Delta_r + \lambda)$, and so $\Delta_q + \Delta_r + \lambda = \Delta$. Let μ be a λ-matching on tuples of length $\ell + \lambda$ which matches i with j if and only if the i-th smallest colour class of c_q and the j-th smallest colour class of c_r have the same colour. It follows that $\mathrm{merge}(\mu, (a_1^q, \ldots, a_{\ell+\lambda}^q), (a_1^r, \ldots, a_{\ell+\lambda}^r), i)$ is the size of the i-th smallest colour class of c_p, for every $i \in [\ell]$. Observe that c_q (and c_r, respectively) is a $(\chi(T_q) - \Delta_q)$-colouring of T_q with Property 1 (or a $(\chi(T_r) - \Delta_r)$-colouring of T_r with Property 1, respectively) whose i-th colour class has exactly a_i^q vertices (a_i^r vertices, respectively), for every $i \in [\ell + \lambda]$. Thus, by the assumption that the claim holds for q and r, $f_{\ell+\lambda}^q((a_1^q, \ldots, a_{\ell+\lambda}^q), \Delta_q) \le m_q$ and $f_{\ell+\lambda}^r((a_1^r, \ldots, a_{\ell+\lambda}^r), \Delta_r) \le m_r$. It is easy to see that the number of monochromatic edges of c_p is $m_q + m_r + \mathrm{val}(\mu, a^q, a^r)$. This implies that $f_\ell^p(a, \Delta)$ is less than or equal to the number of monochromatic edges of c_p.

Suppose for a contradiction that $f_\ell^p(a, \Delta)$ is strictly smaller than the number of monochromatic edges of c_p. This implies that there are

- $\Delta'_q, \Delta'_r, \lambda' \ge 0$ with $\Delta'_q + \Delta'_r + \lambda' = \Delta$,
- tuples $b^q, b^r \in [0..n]^{\ell+\lambda'}$,
- a λ'-matching μ' of $(\ell + \lambda')$-tuples such that $\mathrm{merge}(\mu', b^q, b^r)_i \le a_i^p$, for each $i \in [\ell]$, and
- $f_{\ell+\lambda}^q(b^q, \Delta'_q) + f_{\ell+\lambda}^r(b^r, \Delta'_r) + val(\mu', b^q, b^r)$ is strictly less than the number of monochromatic edges of c_p.

By the assumption of correctness of the claim for q and r, there are

- a $(\chi(T_q) - \Delta'_q)$-colouring c'_q of T_q with Property 1,
- a $(\chi(T_r) - \Delta'_r)$-colouring c'_r of T_r with Property 1, such that
- the number of monochromatic edges of c'_q and c'_r are m'_q and m'_r, respectively,
- the i-th smallest colour class of c'_q (or of c'_r, respectively) has at most b_i^q vertices (b_i^r vertices, respectively), for each $i \in [\ell + \lambda]$,
- $m'_q + m'_r + \mathrm{val}(\mu', b^q, b^r)$ is strictly less than the number of monochromatic edges of c_p.

Construct a $(\chi(T_p) - \Delta)$-colouring c'_p of T_p with Property 1 as follows:

- Two vertices in T_q (in T_r, respectively) obtain the same colour if and only if they obtain the same colour by c'_q (by c'_r, respectively).
- Let $w_q \in T_q$ and $w_r \in T_r$ be two vertices which are contained in the i-th smallest colour class of c'_q and in the i'-th smallest colour class of T_r, respectively. They obtain the same colour if and only if i and i' are both at most $\ell + \lambda'$ and μ' matches i with i'.

It follows immediately from the definition of this new colouring that c'_p is a $(\chi(T_p)-\Delta)$-colouring of T_p with Property 1 such that its i-th smallest colour class contains at most a_i^p vertices, for every $i \in [\ell]$. Further, it has less monochromatic edges than c_p, a contradiction to the choice of c_p. This shows the claim. □

5.2 Hardness Proofs

Theorem 9 MONOCHROMATIC EDGES *is NP-hard for complete multipartite graphs.*

Proof We reduce from MINIMUM SUM OF SQUARES, which takes as input an integer ℓ, an ℓ-tuple $a = (a_1, \ldots, a_\ell)$ of integers, an integer h and an integer J. It asks whether $[\ell]$ can be partitioned into h sets A_i, $i \in [h]$, such that

$$\sum_{i=1}^{h} \left(\sum_{j \in A_i} a_j \right)^2 \leq J.$$

It was shown in [12] that this problem is NP-hard, when we consider $\sum_{j \in [\ell]} a_j$ as the size of an instance. Given an instance (ℓ, a, h, J) of MINIMUM SUM OF SQUARES, we set $D = \frac{1}{2} \sum_{j=1}^{\ell} a_j^2$. We construct a complete multipartite graph G as follows: for every $j \in [\ell]$, let U_j be a set of a_j vertices such that U_j and $U_{j'}$ are disjoint for every $j, j' \in [\ell]$, $j \neq j'$. Set $V(G) = \bigcup_{j \in [\ell]} U_j$ and let $E(G)$ be such that two vertices $v \in U_j$ and $w \in U_{j'}$ are adjacent if and only if $j \neq j'$. We claim that $(G, h, \frac{1}{2}J - D)$ is a YES-instance for MONOCHROMATIC EDGES if and only if (ℓ, a, h, J) is a YES-instance for MINIMUM SUM OF SQUARES.

Assume first that $(G, h, \frac{1}{2}J - D)$ is a YES-instance for MONOCHROMATIC EDGES. We know from Lemma 4 that there is an h-colouring c of G minimizing the number of monochromatic edges such that $|c(U_j)| = 1$, for every $j \in [\ell]$.

For every $i \in [h]$, let $A_i \subseteq [\ell]$ be such that $j \in [\ell]$ is contained in A_i if and only if the vertices of U_j are coloured with colour i.

Then the number of monochromatic edges whose both ends are coloured with colour i equals $\frac{1}{2}(\sum_{j \in A_i} a_j)^2 - \sum_{j \in A_i} a_j^2$ and the total number of monochromatic edges is

$$\frac{1}{2} \sum_{i=1}^{h} \left(\left(\sum_{j \in A_i} a_j \right)^2 - \sum_{j \in A_i} a_j^2 \right) = \frac{1}{2} \sum_{i=1}^{h} \left(\sum_{j \in A_i} a_j \right)^2 - D.$$

Thus,
$$\sum_{i=1}^{h}\left(\sum_{j\in A_i} a_j\right)^2 \leq 2\left(\frac{1}{2}J - D + D\right) = J.$$

It follows that (ℓ, a, h, J) is a YES-instance for MINIMUM SUM OF SQUARES.

Assume now that (ℓ, a, h, J) is a YES-instance for MINIMUM SUM OF SQUARES. For every $j \in A_i$, $i \in [h]$, we colour the corresponding vertex set U_j in G with colour i. As above, the number of monochromatic edges between vertices with colour i is $\frac{1}{2}\sum_{i=1}^{h}\left(\sum_{j\in A_i} a_j\right)^2 - D \leq \frac{1}{2}J - D$.

Thus, $(G, h, \frac{1}{2}J - D)$ is a YES-instance for MONOCHROMATIC EDGES. □

Acknowledgements The authors thank Carolina Lucía Gonzalez for helpful discussions about Theorem 8.

Funding Open access funding provided by University of Fribourg

Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows-Theory, Algorithms and Applications, pp. 461–509. Prentice Hall, New Jersey (1993)
2. Bazgan, C., Bentz, C., Picouleau, C., Ries, B.: Blockers for the stability number and the chromatic number. Graph. Comb. **31**(1), 73–90 (2015)
3. Bentz, C., Costa, M.C., de Werra, D., Picouleau, C., Ries, B.: Minimum d-transversals of maximum-weight stable sets in trees. Electron. Notes Discrete Math. **38**, 129–134 (2011). https://doi.org/10.1016/j.endm.2011.09.022, the Sixth European Conference on Combinatorics, Graph Theory and Applications, EuroComb 2011
4. Bodlaender, H.L., Möhring, R.H.: The pathwidth and treewidth of cographs. In: Scandinavian Workshop on Algorithm Theory. pp. 301–309. Springer (1990)
5. Cerioli, M.R., Lima, P.T.: Intersection of longest paths in graph classes. Discrete Appl. Math. **281**, 96–105 (2020). https://doi.org/10.1016/j.dam.2019.03.022, LAGOS'17: IX Latin and American Algorithms, Graphs and Optimization Symposium, C.I.R.M., Marseille, France - 2017
6. Corneil, D.G., Lerchs, H., Burlingham, L.S.: Complement reducible graphs. Discret. Appl. Math. **3**(3), 163–174 (1981)
7. Diestel, R.: Graph Theory, p. 37. Springer, Heidelberg (2016)
8. Diner, Ö.Y., Paulusma, D., Picouleau, C., Ries, B.: Contraction and deletion blockers for perfect graphs and H-free graphs. Theoret. Comput. Sci. **746**, 49–72 (2018). https://doi.org/10.1016/j.tcs.2018.06.023
9. Galby, E., Mann, F., Ries, B.: Blocking total dominating sets via edge contractions. Theoret. Comput. Sci. **877**, 18–35 (2021). https://doi.org/10.1016/j.tcs.2021.03.028

10. Galby, E., Mann, F., Ries, B.: Reducing the domination number of $(P_3 + kP_2)$-free graphs via one edge contraction. Discret. Appl. Math. **305**, 205–210 (2021). https://doi.org/10.1016/j.dam.2021.09.009
11. Galby, E., Lima, P.T., Mann, F., Ries, B.: Using edge contractions to reduce the semitotal domination number (2021). https://doi.org/10.48550/ARXIV.2107.03755, arXiv:2107.03755
12. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
13. Gavril, F.: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM J. Comput. **1**(2), 180–187 (1972). https://doi.org/10.1137/0201013
14. Paik, D., Reddy, S., Sahni, S.: Deleting vertices to bound path length. IEEE Trans. Comput. **43**(9), 1091–1096 (1994). https://doi.org/10.1109/12.312117
15. Paulusma, D., Picouleau, C., Ries, B.: Reducing the clique and chromatic number via edge contractions and vertex deletions. In: Cerulli, R., Fujishige, S., Mahjoub, A.R. (eds.) Combinatorial Optimization, pp. 38–49. Springer International Publishing, Cham (2016)
16. Picouleau, C., Paulusma, D., Ries, B.: Reducing the chromatic number by vertex or edge deletions. Electron. Notes Discrete Math. **62**, 243–248 (2017). https://doi.org/10.1016/j.endm.2017.10.042, LAGOS'17 - IX Latin and American Algorithms, Graphs and Optimization
17. Poljak, S.: A note on stable sets and colorings of graphs. Comment. Math. Univ. Carol. **15**(2), 307–309 (1974)
18. Zenklusen, R., Ries, B., Picouleau, C., de Werra, D., Costa, M.C., Bentz, C.: Blockers and transversals. Discret. Math. **309**(13), 4306–4314 (2009). https://doi.org/10.1016/j.disc.2009.01.006