http://dx.doi.org/10.1007/978-3-319-55911-7 34

Blocking Independent Sets for H-Free Graphs via Edge Contractions and Vertex Deletions

Daniël Paulusma[1], Christophe Picouleau[2], and Bernard Ries[3](✉)

[1] Durham University, Durham, UK
daniel.paulusma@durham.ac.uk
[2] CNAM, Laboratoire CEDRIC, Paris, France
christophe.picouleau@cnam.fr
[3] University of Fribourg, Fribourg, Switzerland
bernard.ries@unifr.ch

Abstract. Let d and k be two given integers, and let G be a graph. Can we reduce the independence number of G by at least d via at most k graph operations from some fixed set S? This problem belongs to a class of so-called blocker problems. It is known to be co-NP-hard even if S consists of either an edge contraction or a vertex deletion. We further investigate its computational complexity under these two settings:

- we give a sufficient condition on a graph class for the vertex deletion variant to be co-NP-hard even if $d = k = 1$;
- in addition we prove that the vertex deletion variant is co-NP-hard for triangle-free graphs even if $d = k = 1$;
- we prove that the edge contraction variant is NP-hard for bipartite graphs but linear-time solvable for trees.

By combining our new results with known ones we are able to give full complexity classifications for both variants restricted to H-free graphs.

1 Introduction

A graph modification problem aims to modify a graph G, via a small number of operations, into some other graph H that has a certain desired property, which usually describes a certain graph class to which H must belong. In this way a variety of classical graph-theoretic problems is captured. For instance, if only k vertex deletions are allowed and H must be an independent set or a clique, one obtains the Independent Set or Clique problem, respectively.

Instead of specifying a graph class we can specify a graph parameter. That is, given a graph G, a set S of one or more graph operations and an integer k, we ask whether G can be transformed into a graph G' by using at most k operations from S such that $\pi(G') \leq \pi(G) - d$ for some *threshold* $d \geq 0$. Such problems are called *blocker problems*. This is because the set of vertices or edges involved can be viewed as "blocking" π. Identifying such sets may gives us some important information on the structure of the graph.

Blocker problems have been well studied in the recent literature [1–3,5,7,13,14,16,18]; in particular, in [7,14] several relations to other graph problems were identified, such as Hadwiger Number, Club Contraction and a number of graph transversal problems. So far, the graph parameters considered were the chromatic number, the independence number, the clique number, the matching number and the vertex cover number, whereas the set S consisted of a single graph operation, which was either the vertex deletion, edge contraction, edge deletion or the edge addition operation. In this paper we keep the restriction on the size of S, and we let S consist of either a single vertex deletion or a single edge contraction. We mainly consider the independence number α, but for the deletion variant we will also take the clique number ω into account (for reasons we explain later).

Before we can define our problems formally we first need to give some terminology. The *contraction* of an edge uv of a graph G removes the vertices u and v from G, and replaces them by a new vertex made adjacent to precisely those vertices that were adjacent to u or v in G (neither introducing self-loops nor multiple edges). We say that G can be *k-contracted* or *k-vertex-deleted* into a graph G' if G can be modified into G' by a sequence of at most k edge contractions or vertex deletions, respectively. We let π denote the (fixed) graph parameter; as mentioned, in this paper π belongs to $\{\alpha, \omega\}$.

Contraction Blocker(π)
Instance: a graph G and two integers $d, k \geq 0$
Question: can G be k-contracted into a graph G' with $\pi(G') \leq \pi(G) - d$?

Deletion Blocker(π)
Instance: a graph G and two integers $d, k \geq 0$
Question: can G be k-vertex-deleted in a graph G' with $\pi(G') \leq \pi(G) - d$?

If we remove d from the input and fix it instead, we call the resulting problems d-Contraction Blocker(π) and d-Deletion Blocker(π), respectively. Note that 1-Deletion Blocker(α) is equivalent to testing whether the input graph contains a set of S of size at most k that intersects every maximum independent set. If $k = 1$, this is equivalent to testing whether the input graph contains a vertex that is in every maximum independent set. The intersection of all maximum independent sets is known as the core of a graph. Properties of the core have been well studied (see for example [10–12]). In particular, Boros, Golumbic and Levit [4] proved that computing if the core of a graph has size at least ℓ is co-NP-hard for every fixed $\ell \geq 1$. Taking $\ell = 1$ gives co-NP-hardness of 1-Deletion Blocker(α), whereas 1-Contraction Blocker(α) is known to be NP-hard [7].

Due to the above hardness results, it is natural to restrict the input to some special graph class. In a previous paper [7] we considered $\pi \in \{\alpha, \omega, \chi\}$, where

χ denotes the chromatic number of a graph, and we restricted the input to perfect graphs and subclasses of perfect graphs. We showed both new hardness results (e.g., for the class of perfect graphs itself) and tractable results (e.g., for cographs). In a follow-up paper [14] we extended the results of [7] by considering some more subclasses of perfect graphs for $\pi \in \{\omega, \chi\}$. Moreover, for every connected graph H and $\pi \in \{\omega, \chi\}$, we determined the computational complexity of Contraction Blocker(π) and Deletion Blocker(π) for H-free graphs, that is, graphs that do not contain an induced subgraph isomorphic to H.

Our Results

We settle the computational complexity of Contraction Blocker(α) and Deletion Blocker(α)restricted to H-free graphs for all graphs H (including those that are disconnected). We observe that Deletion Blocker(α) restricted to H-free graphs is equivalent to Deletion Blocker(ω) for $\overline{H}$-free graphs, where $\overline{H}$ denotes the complement of H. Hence, as a corollary, we obtain an extension of the aforementioned classification of [14] for Deletion Blocker(ω) for H-free graphs from connected graphs H to all graphs H.

To prove the above results we first show that Contraction Blocker(α) is NP-hard for bipartite graphs in Sect. 3. In the same section we complement this result by showing that Contraction Blocker(α) can be solved in linear time for trees. Then, in Sect. 4, we prove that Deletion Blocker(α) is co-NP-hard for triangle-free graphs even if $d = k = 1$ (in contrast the problem is polynomial-time solvable for bipartite graphs [2,5]). In Sect. 5 we extend our result for triangle-free graphs to other graph classes for which Independent Set is NP-complete. That is, we give a sufficient condition on such a graph class $\mathcal{G}$, such that Deletion Blocker(α) is co-NP-hard for $\mathcal{G}$ even if $d = k = 1$. This condition is similar to a previous condition when $\pi \in \{\chi, \omega\}$ [14]. In Sect. 6 we combine our new results from Sects. 4 and 5 with known ones to obtain the classifications for H-free graphs. In Sect. 7 we compare our new results with the results of our previous paper [14] and list some open problems.

Recall that the deletion variant for $k = d = 1$ is equivalent to asking whether a graph has a vertex that is in every maximum independent set. As such, our hardness results in Sects. 4 and 5 strengthen the aforementioned result of Boros, Golumbic and Levit [4], who proved co-NP-hardness of the latter problem for general graphs. Note that for graph classes, for which Independent Set is NP-complete, membership of our problems in NP is unknown. Contrary to those graph classes, for which Independent Set is polynomial-time solvable and which are closed under the graph operation under consideration, a certificate consisting of a sequence of edge contractions or vertex deletions no longer suffices.

2 Preliminaries

We only consider finite, undirected graphs that have no self-loops and no multiple edges (we recall that when we contract an edge no self-loops or multiple edges are created). See [6] for undefined terminology and notation.

Let $G = (V, E)$ be a graph. For a family $\{H_1, \ldots, H_p\}$ of graphs, G is said to be $(H_1, \ldots, H_p)$-*free* if G has no induced subgraph isomorphic to a graph in $\{H_1, \ldots, H_p\}$; if $p = 1$ we may write H_1-free instead of (H_1)-free. The *complement* of G is the graph $\overline{G} = (V, \overline{E})$ with vertex set V and an edge between two vertices u and v if and only if $uv \notin E$. For a subset $S \subseteq V$, we let $G[S]$ denote the subgraph of G *induced* by S, which has vertex set S and edge set $\{uv \in E \mid u, v \in S\}$. We write $H \subseteq_i G$ if a graph H is an induced subgraph of G.

Let G be a graph. For a vertex $v \in V$, we write $G - v = G[V \setminus \{v\}]$ and for a subset $V' \subseteq V$ we write $G - V' = G[V \setminus V']$. Recall that the contraction of an edge $uv \in E$ removes the vertices u and v from G and replaces them by a new vertex that is made adjacent to precisely those vertices that were adjacent to u or v in G. In that case we may also say that u is *contracted onto* v, and we use v to denote the new vertex resulting from the edge contraction. The *subdivision* of an edge $uv \in E$ removes the edge uv from G and replaces it by a new vertex w and two edges uw and wv.

Let G_1 and G_2 be two vertex-disjoint graphs. The *disjoint union* $G_1 + G_2$ has vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. The disjoint union of k copies of a graph G is denoted by kG. The *join* $G_1 \otimes G_2$ adds an edge between every vertex of G_1 and every vertex of G_2. For $r \geq 1$, the path, cycle and complete graph on r vertices are denoted by P_r, C_r and K_r respectively. The graph C_3 is also called the *triangle*. The *claw* $K_{1,3}$ is the 4-vertex star (that is, the graph with vertices u, v_1, v_2, v_3 and edges uv_1, uv_2, uv_3).

Let $G = (V, E)$ be a graph. A subset $K \subseteq V$ is called a *clique* of G if any two vertices in K are adjacent to each other. The *clique number* $\omega(G)$ is the number of vertices in a maximum clique of G. A subset $I \subseteq V$ is called an *independent set* of G if any two vertices in I are non-adjacent to each other. The *independence number* $\alpha(G)$ is the number of vertices in a maximum independent set of G. A subset of edges $M \subseteq E$ is called a *matching* if no two edges of M share a common end-vertex. The *matching number* $\mu(G)$ is the number of edges in a maximum matching of a graph G. A vertex v such that M contains an edge incident with v is *saturated* by M; otherwise v is *unsaturated* by M. A subset $S \subseteq V$ is a *vertex cover* of G if every edge of G is incident with at least one vertex of S.

The problems Clique and Independent Set are those of testing if a graph has a clique or independent set, respectively, of size at least k. The Vertex Cover problem is that of testing if a graph has a vertex cover of size at most k.

A graph is *cobipartite* if it is the complement of a *bipartite* graph, that is, a graph whose vertex set can be partitioned into two sets that each form a (possibly empty) independent set. A graph is a *split graph* if it has a *split partition*, which is a partition of its vertex set into a clique K and an independent set I. Split graphs coincide with $(2P_2, C_4, C_5)$-free graphs [8]. A P_4-free graph is also called a *cograph*.

3 Bipartite Graphs and Trees

Our first lemma below follows directly from a result of Golovach, Heggernes, van't Hof and Paul [9] on the so-called s- Club Contraction problem; see [7] for further details.

Lemma 1 **([7]).** *1-*Contraction Blocker*(α) is* NP*-complete for cobipartite graphs.*

If $\pi \in \{\chi, \omega\}$, Contraction Blocker(π) is trivial in bipartite graphs. To the contrary, for $\pi = \alpha$, we will show that Contraction Blocker(π) is NP-hard for bipartite graphs. The complexity of d-Contraction Blocker(α) remains open for bipartite graphs. Bipartite graphs are not closed under edge contraction. Therefore membership to NP cannot be established by taking a sequence of edge contractions as the certificate, even though due to König's Theorem (see, for example, [6]), Independent Set is polynomial-time solvable for bipartite graphs.

Theorem 1. Contraction Blocker*(α) is* NP*-hard on bipartite graphs.*

Proof. We know from Lemma 1 that 1-Contraction Blocker(α) is NP-complete on cobipartite graphs, which have independence number 2. Consider a cobipartite graph G with m edges and an integer k, which together form an instance of 1-Contraction Blocker(α). Subdivide each of the m edges of G in order to obtain a bipartite graph G'. We claim that (G, k) is a yes-instance of 1-Contraction Blocker(α) if and only if $(G', \alpha(G') - 1, k + m)$ is a yes-instance of Contraction Blocker(α).

First suppose that (G, k) is a yes-instance of 1-Contraction Blocker(α). In G' we first perform m edge contractions to get G back. We then perform k edge contractions to get independence number $\alpha(G) - 1 = 1 = \alpha(G') - (\alpha(G') - 1)$. Hence, $(G', \alpha(G') - 1, k + m)$ is a yes-instance of Contraction Blocker(α).

Now suppose that $(G', \alpha(G') - 1, k + m)$ is a yes-instance of Contraction Blocker(α). Then there exists a sequence of $k + m$ edge contractions that transform G' into a complete graph K. We may assume that K has size at least 4 (as we could have added without loss of generality three dominating vertices to G without increasing k). As K has size at least 4, each subdivided edge must be contracted back to the original edge again. This operation costs m edge contractions, so we contract G to K using at most k edge operations. Hence, (G, k) is a yes-instance of 1-Contraction Blocker(α). This proves the claim and hence the theorem. ☐

We complement Theorem 1 by showing that Contraction Blocker(α) is linear-time solvable on trees. In order to prove this result we make a connection to the matching number μ.

Theorem 2. Contraction Blocker*(α) is linear-time solvable on trees.*

Proof. Let (T, d, k) be an instance of Contraction Blocker(α), where T is a tree on n vertices. We first describe our algorithm and prove its correctness. Afterwards, we analyze its running time. Throughout the proof let M denote a maximum matching of T.

As $\alpha(T) + \mu(T) = n$ by König's Theorem (see, for example, [6]), we find that (T, d, k) is a no-instance if $d > n - \mu(T)$. Assume that $d \leq n - \mu(T)$. We observe that trees are closed under edge contraction. Hence, contracting an edge of T results in a new tree T'. Moreover, T' has $n - 1$ vertices and the edge contraction neither increased the independence number nor the matching number. As $\alpha(T) + \mu(T) = n$ and similarly $\alpha(T') + \mu(T') = n - 1$, this means that either $\alpha(T') = \alpha(T) - 1$ or $\mu(T') = \mu(T) - 1$.

First suppose that $d \leq n - 2\mu(T)$. There are exactly $\sigma(T) = n - 2\mu(T)$ vertices that are unsaturated by M. Let uv be an edge, such that u is unsaturated. As M is maximum, v must be saturated. Then, by contracting uv, we obtain a tree T' such that $\mu(T') = \mu(T)$. It follows from the above that $\alpha(T') = \alpha(T) - 1$. Say that we contracted u onto v. Then in T' we have that v is saturated by M, which is a maximum matching of T' as well. Thus, if $d \leq n - 2\mu(T)$, contracting d edges, one of the end-vertices of which is unsaturated by M, yields a tree T' with $\mu(T') = \mu(T)$ and $\alpha(T') = \alpha(T) - d$. Since an edge contraction reduces the independence number by at most 1, it follows that this is optimal. Hence, as $d \leq n - 2\mu(T)$, we find that (G, T, k) is a yes-instance if $k \geq d$ and a no-instance if $k < d$.

Now suppose that $d > n - 2\mu(T)$. Suppose that we first contract the $n - 2\mu(T)$ edges that have exactly one end-vertex that is unsaturated by M. It follows from the above that this yields a tree T' with $\mu(T') = \mu(T)$ and $\alpha(T') = \alpha(T) - (n - 2\mu(T))$. Since T' does not contain any unsaturated vertex, M is a perfect matching of T'. Then, contracting any edge in T' results in a tree T'' with $\mu(T'') = \mu(T') - 1$ and thus, $\alpha(T'') = \alpha(T')$. If we contract an edge $uv \in M$, the resulting vertex uv is unsaturated by $M' = M \setminus \{uv\}$ in T''. Hence, as explained above, if in addition we contract now an edge $(uv)w$, we obtain a tree T''' with $\alpha(T''') = \alpha(T'') - 1$ and $\mu(T''') = \mu(T'')$. Repeating this procedure, we may reduce the independence number of T by d with $n - 2\mu(T) + 2(d - n + 2\mu(T)) = 2(d + \mu(T)) - n$ edge contractions. Below we show that this is optimal.

Suppose that we contract p edges in T. Let T' be the resulting tree. We have $\alpha(T') + \mu(T') = n - p$. As $\mu(T') \leq \frac{1}{2}(n - p)$, this means that $\alpha(T') \geq \frac{1}{2}(n - p)$. If $p < 2(d + \mu(T)) - n$ we have $-\frac{p}{2} > -(d + \mu(T)) + \frac{n}{2}$, and thus

$$\begin{aligned} \alpha(T') &\geq \tfrac{1}{2}(n - p) \\ &> \tfrac{n}{2} - d - \mu(T) + \tfrac{n}{2} \\ &= \alpha(T) - d. \end{aligned}$$

So at least $2(d + \mu(T)) - n$ edge contractions are necessary to decrease the independence number by d. It remains to check if k is sufficiently high for us to allow this number of edge contractions.

As we can find a maximum matching of tree T (and thus compute $\mu(T)$) in $O(n)$ time by using the algorithm of Savage [17], our algorithm runs in $O(n)$ time. $\square$

Remark 1. By König's Theorem, we have that $\alpha(G) + \mu(G) = |V(G)|$ for any bipartite graph G, but we can only use the proof of Theorem 2 to obtain a result for trees for the following reason: trees form the largest subclass of (connected) bipartite graphs that are closed under edge contraction, and this property plays a crucial role in our proof.

4 Triangle-Free Graphs

In this section we show that Deletion Blocker(α) is co-NP-hard for triangle-free graphs even if $d = k = 1$. We call a vertex *forced* if it is in every maximum independent set of a graph [5]. Recall that the set of all forced vertices is called the *core* of a graph and that Boros, Golumbic and Levit [4] proved that computing whether the core of a graph has size at least k is co-NP-hard for every fixed $k \geq 1$. As a special case of their result, the problem of testing the existence of a forced vertex is co-NP-hard. In this section we prove that the latter problem, or equivalently, Deletion Blocker(α) with $d = k = 1$, stays co-NP-hard even for triangle-free graphs.

We need some terminology and a well-known observation that follows from a construction of Poljak [15]. We say that we 2-*subdivide* an edge e of a graph G if we apply two consecutive edge subdivisions on e. It is readily seen that a graph G with m edges has an independent set of size k if and only if the graph obtained by 2-subdividing each edge of G has an independent set of size $k + m$ (see also [15]). Let $\mathcal{G}$ be a graph class. Then we let $\mathcal{G}^2$ be the graph class obtained from $\mathcal{G}$ after 2-subdividing each edge in every graph in $\mathcal{G}$.

Lemma 2. *([15]). If* Independent Set *is* NP*-complete for a graph class $\mathcal{G}$, then it is also* NP*-complete for $\mathcal{G}^2$.*

Two vertices in a graph G are *true twins* if they are adjacent to each other and apart from this have the same neighbours in G. The graph G^* obtained from a graph G by adding a new vertex u' for each vertex u of G that is a true twin of u is called the *twin graph* of G; see Fig. 1 for an example. We call u' the *copy* of u. Let $\mathcal{G}^*$ be the graph class obtained from a graph class $\mathcal{G}$ by replacing each graph in $\mathcal{G}$ by its twin graph. Note that $\alpha(G^*) = \alpha(G)$ for every graph G. Hence the following lemma holds.

Lemma 3. *If* Independent Set *is* NP*-complete for a graph class $\mathcal{G}$, then it is also* NP*-complete for $\mathcal{G}^*$.*

Theorem 3. Deletion Blocker*(α) is co-*NP*-hard for triangle-free graphs even if $d = k = 1$.*

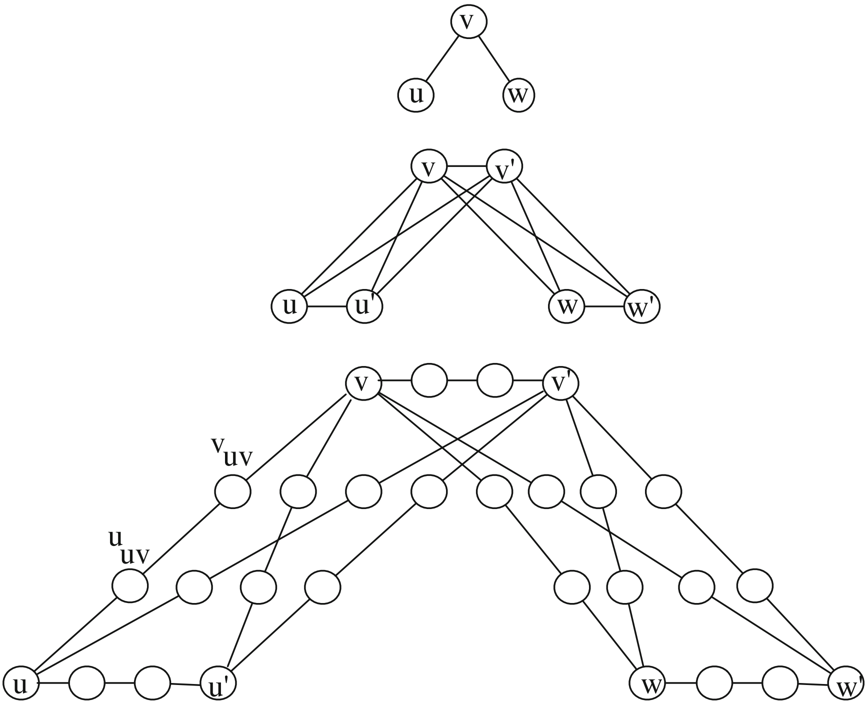

Fig. 1. An example of a graph G' constructed from a graph G via the graph G^*.

Proof. We prove that the equivalent problem of testing whether a triangle-free graph has a forced vertex is co-NP-hard via a reduction from INDEPENDENT SET. Let G be a graph with at least two vertices. From G we construct its twin graph G^*. We now subdivide each edge of G^* twice. We call the resulting graph G'. For an edge $e = uv$ in G^* (where $v = u'$ is possible), we call the two newly introduced vertices u_e and v_e, where u_e is the vertex adjacent to u and v_e the one adjacent to v. See Fig. 1 for an example of a graph G'.

We now show the following claim.

Claim. G' has no forced vertices.

We prove this claim as follows. For contradiction, suppose x is a forced vertex of G', that is, x belongs to every maximum independent set of G'. First suppose $x = u$ or $x = u'$ for some vertex u of G, say $x = u$. Then, by symmetry, its copy u' is also a forced vertex of G'. Let I be a maximum independent set of G'. Since u, u' are forced, we have $u, u' \in I$ and therefore $u_{uu'}, u'_{uu'} \notin I$. But then $(I \setminus \{u\}) \cup \{u_{uu'}\}$ is another maximum independent set of G' not containing u, a contradiction.

Now suppose $x = u_{uu'}$ for some vertex u of G. Then, by symmetry, $u'_{uu'}$ is a forced vertex as well. This is a contradiction, since $u_{uu'}$ and $u'_{uu'}$ are adjacent.

Finally suppose $x = u_{uv}$ for some vertices u, v of G^* with $v \neq u'$. Let I be a maximum independent set of G'. Since u_{uv} is a forced vertex, we have $u_{uv} \in I$

and therefore $v_{uv} \notin I$. From the above we know that v cannot be forced. Hence, we may assume without loss of generality that I is chosen in such a way that $v \notin I$. But then $(I \setminus \{u_{uv}\}) \cup \{v_{uv}\}$ is another maximum independent set of G' not containing u_{uv}, a contradiction. This completes the proof of the claim.

We continue as follows. By Lemmas 2 and 3, Independent Set is NP-complete even for the class of graphs G' constructed above. Let ℓ be an integer that together with G' forms an instance of Independent Set. In particular note that G' is triangle-free. Let m be the number of edges of G^*. Then we may assume without loss of generality that $\ell \geq m$ (as a trivial lower bound for the size of a maximum independent set in G' is m: we can construct an independent set of size m by taking for each edge uv of G^*, one of the two vertices u_{uv}, v_{uv}).

We construct a graph F from G' by taking an independent set J on $\ell+1-m$ vertices and by making each vertex of J adjacent to every vertex u of G and to its copy u'. Note that we do not make vertices of J adjacent to any vertices in G' obtained from 2-subdividing the edges of G^*. Hence, as G' is triangle-free, J is independent, and no vertex u of G is adjacent to its copy u' in G', we find that F is triangle-free.

In order to complete our proof we are left to show that $\alpha(G') \leq \ell$ if and only if F contains a forced vertex y, or equivalently, $\alpha(F-y) \leq \alpha(F)-1$.

First suppose that $\alpha(G') \leq \ell$. We claim that every vertex in J is forced. In order to see this let $y \in J$. First note that $\alpha(F) \geq \ell+1$, as the set of vertices that consists of all vertices of J and, for each edge uv in G^*, exactly one of the two vertices u_{uv}, v_{uv} is an independent set of size $\ell+1-m+m=\ell+1$. Now let I be a maximum independent set of $F-y$. If I contains a vertex y' of J, then I must have size ℓ (since I cannot contain a vertex u of G or its copy u', as y' is adjacent to such vertices). If I does not contain a vertex of J, then I must be an independent set of G'. Then I has size at most $\alpha(G') \leq \ell$ by our assumption on $\alpha(G')$. In fact, as ℓ is a lower bound on the size of I (recall that $\alpha(F) \geq \ell+1$), we have that I has size ℓ in this case as well. Hence, in both cases we find that $\alpha(F-y)=\ell \leq \alpha(F)-1$ implying that y is a forced vertex of F.

Now suppose that F contains a forced vertex y, so $\alpha(F-y) \leq \alpha(F)-1$. In fact we must have $\alpha(F-y)=\alpha(F)-1$. We distinguish three cases.

First assume that y belongs to J. Let I be a maximum independent set of F. Then y must be in I, as y is forced. This means that I must have size $\ell+1$, thus $\alpha(F)=\ell+1$, as I cannot contain a vertex u of G or its copy u' (because $y \in I$) and I can contain, besides all vertices of J, exactly one of u_{uv}, v_{uv} for every edge uv of G^*. As y is forced, this implies that $\alpha(F-y)=\ell$. As G' is an induced subgraph of $F-y$, this means that $\alpha(G') \leq \ell$.

Now assume that $y=u$ or $y=u'$ for some u in G. Let I be a maximum independent set of F. As y is forced, y belongs to I. As y is adjacent to every vertex in J, we find that no vertex of J belongs to I. Then I is a maximum independent set of G'. However, in that case we can replace I by another maximum independent set of G', and thus of F, that does not contain y (by the above Claim). So we conclude that y is not a forced vertex of F, which is a contradiction.

Finally assume that $y = u_{uv}$ for some edge uv of G^* (where $v = u'$ is possible). If I shares no vertices with J, then we repeat the arguments of the previous case. Suppose I intersects with J. Then I does not contain v. Hence we may replace y by v_{uv} to get a maximum independent set of F that does not contain y. This implies that y is not forced, a contradiction. This completes the proof of Theorem 3. $\square$

5 A Sufficient Condition for Hardness

In this section we give a sufficient condition for computational hardness of Deletion Blocker(α). Let $\mathcal{G}$ be a graph class with the following property: if $G \in \mathcal{G}$, then so are $G \otimes G$ and $G \otimes sP_1$ for any integer $s \geq 1$. We call such a graph class *stable-proof*. We show that determining the existence of a forced vertex is co-NP-hard on any stable-proof graph class, for which Independent Set is NP-complete (note that we can only show co-NP-hardness for reasons discussed before).

Theorem 4. *If* Independent Set *is* NP*-complete for a stable-proof graph class $\mathcal{G}$, then* Deletion Blocker*(α) is co-*NP*-hard for $\mathcal{G}$, even if $d = k = 1$.*

Proof. Let $\mathcal{G}$ be a graph class that is stable-proof. From a given graph $G \in \mathcal{G}$ and integer $\ell \geq 1$ we construct the graph $G' = G \otimes G \otimes (\ell + 1)P_1$. Note that $G' \in \mathcal{G}$ by definition and that $\alpha(G') = \max\{\alpha(G), \ell+1\}$. We claim that $\alpha(G) \leq \ell$ if and only if G' can be 1-vertex-deleted into a graph G^* with $\alpha(G^*) \leq \alpha(G') - 1$.

First suppose that $\alpha(G) \leq \ell$. Then $\alpha(G') = \ell + 1$. In G' we delete a vertex v of the $(\ell+1)P_1$. This yields the graph $G^* = G \otimes G \otimes \ell P_1$. We have that $\alpha(G^*) = \max\{\alpha(G), \ell\} = \ell$. As $\alpha(G') = \ell + 1$, this means that $\alpha(G^*) \leq \alpha(G') - 1$.

Now suppose that G' can be 1-vertex-deleted into a graph G^* with $\alpha(G^*) \leq \alpha(G') - 1$. As deleting a vertex in one of the two copies of G does not lower the independence number of G', the deleted vertex must belong to the $(\ell + 1)P_1$. This means that $G^* = G \otimes G \otimes \ell P_1$. As $\alpha(G^*) = \max\{\alpha(G), \ell\} \leq \alpha(G') - 1 = \max\{\alpha(G), \ell + 1\} - 1$, we conclude that $\alpha(G) \leq \ell$. $\square$

Remark 2. We cannot apply Theorem 4 on triangle-free graphs, as the class of triangle-free graphs is not stable-proof.

6 The Two Classifications

In this section we combine Theorems 3 and 4 with a number of known results for obtaining dichotomy results for our two blocker problems restricted to H-free graphs. Before we present these dichotomies we first state some known results that we need for their proofs.

Lemma 4. (*[15]*). Independent Set *is* NP*-complete for C_5-free graphs.*

Lemma 5. (*[15]*). Vertex Cover *is* NP*-complete for* C_3*-free graphs.*

We also need two of our previous results.

Lemma 6. (*[14]*). *Let* G *be a triangle-free graph containing at least one edge and let* $k \geq 1$ *be an integer. Then* (G, k) *is a yes-instance of* 1-Deletion Blocker*(*ω*) if and only if* (G, k) *is a yes-instance of* Vertex Cover*.*

Lemma 7. (*[7]*). *The problems* Contraction Blocker*(*α*) and* Deletion Blocker*(*α*) are polynomial-time solvable for cographs but* NP*-complete on split graphs.*

We also use the following observation.

Lemma 8. *If* H *is a* $(3P_1, 2P_2)$*-free forest, then* $H \subseteq_i P_4$*.*

Proof. As H is $3P_1$-free, H contains at most two connected components. Suppose H contains exactly two connected components. Then, as H is $2P_2$-free, at least one of these components must be a P_1. As H is $3P_1$-free, this means that H is an induced subgraph of $P_2 + P_1$, so $H \subseteq_i P_4$. Suppose H is connected. As H is $3P_1$-free, H contains no claw and no path on more than five vertices. Hence, $H \subseteq_i P_4$. □

We are now ready to present our first classification.

Theorem 5. *Let* H *be a graph. If* $H \subseteq_i P_4$*, then* Contraction Blocker*(*α*) is polynomial-time solvable for* H*-free graphs, otherwise it is* NP*-hard for* H*-free graphs.*

Proof. Let H be a graph. Recall that a cograph is a P_4-free graph. Hence, if H is an induced subgraph of P_4, then we use Lemma 7 to obtain polynomial-time solvability.

Now suppose that H is not an induced subgraph of P_4. If H contains an induced cycle that is odd, then we use Theorem 1 to obtain NP-hardness. If H contains an induced cycle that is even, then H either contains an induced C_4 or, if the even cycle has at least six vertices, an induced $2P_2$. This means that we can use Lemma 7 to obtain NP-hardness after recalling that split graphs are $(2P_2, C_4)$-free. Assume H contains no cycle. Then H is a forest. If H contains an induced $3P_1$, then we use Lemma 1 to obtain NP-hardness, after observing that cobipartite graphs are $3P_1$-free. Assume H is $3P_1$-free. Then $2P_2 \subseteq_i H$ by Lemma 8, which means we can use Lemma 7 again to obtain NP-hardness. □

Remark 3. In some cases of Theorem 5, such as when $H = C_5$, we could have applied Theorem 4 to obtain co-NP-hardness even if $d = k = 1$.

We now consider the vertex deletion variant and present our second classification.

Theorem 6. *Let* H *be a graph. If* $H \subseteq_i P_4$*, then* Deletion Blocker*(*α*) is polynomial-time solvable for* H*-free graphs, otherwise it is* NP*-hard or co-*NP*-hard for* H*-free graphs.*

Proof. Let H be a graph. If $H \subseteq_i P_4$, then we use Lemma 7 to obtain polynomial-time solvability. Suppose H is not an induced subgraph of P_4. First suppose H contains an induced cycle C_r. If $r = 3$, then we use Theorem 3 to find that the problem is co-NP-hard even if $d = k = 1$. If $r = 4$, then we use Lemma 7 (after recalling that split graphs are C_4-free) to find that the problem is NP-hard. If $r = 5$, then we combine Lemma 4 with Theorem 4 after observing that the class of C_5-free graphs is stable-proof. We then find that the problem is co-NP-hard even if $d = k = 1$. Note that we could have applied Lemma 7 to obtain NP-hardness, as split graphs are C_5-free, If $r \geq 6$, then H contains an induced $2P_2$ and we apply Lemma 7 (as split graphs are $2P_2$-free) to find that the problem is NP-hard.

Now assume that H is forest. By Lemma 8, either $2P_2 \subseteq_i H$ or $3P_1 \subseteq_i H$. If $2P_2 \subseteq_i H$, then we apply Lemma 7 again to obtain NP-hardness. If $3P_1 \subseteq_i H$, then we apply Lemmas 5 and 6 to obtain NP-hardness after observing that a graph is a yes-instance for 1-Deletion Blocker(α) if and only if its complement is a yes-instance for 1-Deletion Blocker(ω). ☐

We are left to state our result for Deletion Blocker(ω), which follows immediately from Theorem 6 after making two observations. First, Deletion Blocker(ω) for H-free graphs is equivalent to Deletion Blocker(α) for $\overline{H}$-free graphs. Second, the graph P_4 is self-complementary, that is, $\overline{P_4} = P_4$.

Theorem 7. *Let H be a graph. If $H \subseteq_i P_4$, then* Deletion Blocker*(ω) is polynomial-time solvable for H-free graphs; otherwise it is co-*NP*-hard or* NP*-hard for H-free graphs.*

7 Conclusions

For every graph H we determined the computational complexities of Contraction Blocker(α) and Deletion Blocker(π) ($\pi \in \{\alpha, \omega\}$) restricted to H-free graphs, and it would be interesting to generalize these results to families of more than one forbidden induced subgraph. In our previous paper [14] we obtained dichotomies for $\pi \in \{\omega, \chi\}$ but for three of the four classifications we needed to assume that H is connected. For comparing our new results with previous results we therefore need to restrict ourselves to connected graphs H. This leads to the following summary:
For a connected graph H, the following holds:

(i) *If $H \subseteq_i P_4$* or *$H \subseteq_i \overline{P_1 + P_3}$* then Contraction Blocker(ω) *is polynomial time solvable for H-free graphs; otherwise it is co-*NP*-hard for H-free graphs.*
(ii) *For $\pi \in \{\alpha, \chi\}$, if $H \subseteq_i P_4$ then* Contraction Blocker(π) *is polynomial time solvable for H-free graphs; otherwise it is co-*NP*-hard for H-free graphs.*
(iii) *For $\pi \in \{\alpha, \omega, \chi\}$, if $H \subseteq_i P_4$then* Deletion Blocker(π) *is polynomial time solvable for H-free graphs; otherwise it is co-*NP*-hard for H-free graphs.*

It is an open problem to generalize the results of the above summary from connected graphs H to arbitrary graphs H. For part (i) we need to settle one remaining case, namely $H = C_3 + P_1$ [14]. Part (ii) has been generalized to arbitrary graphs already; see [14] for the case when $\pi = \chi$ and see Sect. 6 for the case when $\pi \in \{\alpha, \omega\}$. Part (iii) has been settled for all graphs H already for $\pi \in \{\alpha, \omega\}$ (Sect. 6), whereas the situation for $\pi = \chi$ is less clear with a number of cases still being open; in particular polynomial-time results for disconnected graphs H exist incomparable to the case when $H \subseteq_i P_4$, e.g., if $H = 3P_1$ [14].

It is possible to construct graph classes for which a blocker problem is tractable, but the original problem is NP-complete. Take for instance the class of graphs G' from the proof of Theorem 3. The Independent Set problem is NP-complete for this graph class, but its members are all no-instances of Contraction Blocker(α) when $d = k = 1$. However, this class is not a hereditary graph class, that is, it is not closed under vertex deletion. In fact we do not know of such examples of hereditary graph classes. Hence, it would be interesting to prove for $\pi \in \{\alpha, \omega, \chi\}$ whether Contraction Blocker(π) and Deletion Blocker(π) are computationally hard on every hereditary graph class $\mathcal{G}$, for which Independent Set, Clique or Coloring, respectively, is NP-complete.

Finally, we have shown that Contraction Blocker(α) is NP-hard for bipartite graphs. We pose the question of determining the computational complexity of d-Contraction Blocker(α) ($d \geq 1$) restricted to bipartite graphs as an open problem.

References

1. Bazgan, C., Bentz, C., Picouleau, C., Ries, B.: Blockers for the stability number and the chromatic number. Graphs Comb. **31**, 73–90 (2015)
2. Bazgan, C., Toubaline, S., Tuza, Z.: The most vital nodes with respect to independent set and vertex cover. Discrete Appl. Math. **159**, 1933–1946 (2011)
3. Bentz, C., Costa, M.-C., de Werra, D., Picouleau, C., Ries, B.: Weighted transversals and blockers for some optimization problems in graphs. In: Progress in Combinatorial Optimization, Wiley-ISTE (2012)
4. Boros, E., Golumbic, M.C., Levit, V.E.: On the number of vertices belonging to all maximum stable sets of a graph. Discrete Appl. Math. **124**, 17–25 (2002)
5. Costa, M.-C., de Werra, D., Picouleau, C.: Minimum d-blockers and d-transversals in graphs. J. Comb. Optim. **22**, 857–872 (2011)
6. Diestel, R.: Graph Theory. Springer, Heidelberg (2005)
7. Diner, Ö.Y., Paulusma, D., Picouleau, C., Ries, B.: Contraction blockers for graphs with forbidden induced paths. In: Paschos, V.T., Widmayer, P. (eds.) CIAC 2015. LNCS, vol. 9079, pp. 194–207. Springer, Cham (2015). doi:10.1007/978-3-319-18173-8_14
8. Földes, S., Hammer, P.L.: Split graphs. In: 8th South-Eastern Conference on Combinatorics, Graph Theory and Computing, Congressus Numerantium, vol. 19, pp. 311–315 (1977)
9. Golovach, P.A., Heggernes, P., Hof, P.V., Paul, C.: Hadwiger number of graphs with small chordality. In: Kratsch, D., Todinca, I. (eds.) WG 2014. LNCS, vol. 8747, pp. 201–213. Springer, Cham (2014). doi:10.1007/978-3-319-12340-0_17

10. Hammer, P.L., Hansen, P., Simeone, B.: Vertices belonging to all or to no maximum stable sets of a graph. SIAM J. Algebraic Discrete Methods **3**, 511–522 (1982)
11. Levit, V.E., Mandrescu, E.: Combinatorial properties of the family of maximum stable sets of a graph. Discrete Appl. Math. **117**, 149–161 (2002)
12. Levit, V.E., Mandrescu, E.: Vertices belonging to all critical sets of a graph. SIAM J. Discrete Math. **26**, 399–403 (2012)
13. Pajouh, F.M., Boginski, V., Pasiliao, E.L.: Minimum vertex blocker clique problem. Networks **64**, 48–64 (2014)
14. 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.) ISCO 2016. LNCS, vol. 9849, pp. 38–49. Springer, Cham (2016). doi:10.1007/978-3-319-45587-7_4
15. Poljak, S.: A note on the stable sets and coloring of graphs. Comment. Math. Univ. Carol. **15**, 307–309 (1974)
16. Ries, B., Bentz, C., Picouleau, C., de Werra, D., Costa, M.-C., Zenklusen, R.: Blockers and transversals in some subclasses of bipartite graphs: when caterpillars are dancing on a grid. Discrete Math. **310**, 132–146 (2010)
17. Savage, C.: Maximum matchings and trees. Inf. Process. Lett. **10**, 202–205 (1980)
18. Toubaline, S.: Détermination des éléments les plus vitaux pour des problèmes de graphes, Ph. D thesis, Université Paris-Dauphine (2010)