Using Edge Contractions and Vertex Deletions to Reduce the Independence Number and the Clique Number

Felicia Lucke[0000−0002−9860−2928] and Felix Mann(✉)[0000−0003−0016−4024]

Université de Fribourg, Boulevard de Pérolles 90, 1700 Fribourg, Switzerland
{felicia.lucke,felix.mann}@unifr.ch

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 also 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 on 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 $d = 1$. Our results answer several open questions stated in [Diner et al., Theoretical Computer Science, 746, p. 49-72 (2012)].

Keywords: blocker problems · edge contraction · vertex deletion · 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 $\pi(G') \leq \pi(G) - d$.

As in the case of regular graph modification problems, we often consider a set of operations consisting 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 [12]), the matching number μ (see [14]), the length of a longest path (see [3,10]), the (total or semi-total) domination number γ (γ_t and γ_{t2}, respectively) (see [6–8]), the clique number ω (see [11]) and the independence number α (see [2]).

In this paper, the set of allowed graph operations will always consist of only one operation, either *vertex deletion* or *edge contraction*. 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. We consider the following two 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 $\pi(G/S) \leq \pi(G) - d$?

When d is not fixed but part of the input, the problems are called DELETION BLOCKER(π) and CONTRACTION BLOCKER(π) , respectively.

When $\pi = \alpha$ or $\pi = \omega$, both problems above are NP-hard on general graphs [5], so it is natural to ask if these problems remain NP-hard when the input is restricted to a special graph class.

Table 1. The table of complexities for some graph classes. 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 [5], where an older version of this table is given.

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

The authors of [5] 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 the problem when d is fixed. We show that Contraction Blocker(α) in bipartite graphs is solvable in polynomial time if d is fixed and that the other problems are NP-hard 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 graphs*, that is, to determine the complexity of $(d\text{-})$Deletion Blocker(π) or $(d\text{-})$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 [5]), Contraction Blocker(γ_{t2}) (for $d = k = 1$, [8]), Contraction Blocker(γ_t) (for $d = k = 1$, [6]) and Contraction Blocker(γ) (for $d = k = 1$, [7]). In [5], 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.

The paper is organised as follows: In Section 2 we explain notation and terminology. In Section 3 we give the proofs of NP-hardness or NP-completeness of the aforementioned problems. Finally, in Section 4 we give a polynomial-time algorithm for d-Contraction Blocker(α) in bipartite graphs.

2 Preliminaries

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

We refer the reader to [4] for any terminology not defined here.

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 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 *closed neighbourhood of v*, denoted by $N_G[v]$, is the set $\{u \in V(G) : \text{dist}_G(u, v) \le 1\}$. Similarly, we define for a set $U \subseteq V(G)$ the closed neighbourhood of U as $N_G[U] = \{u \in V(G) : \exists v \in U, \text{dist}_G(v, u) \le 1\}$. 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 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 $\mathrm{dist}_G(V(A), V(B)) = 1$. 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.

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 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.

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. 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.

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 Figure 1 a *paw*.
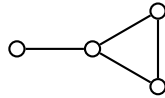

Fig. 1. The paw

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 [4]). *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 [13] it was shown that INDEPENDENT SET is NP-complete in C_3-free graphs. This and Lemma 2 imply the following corollary.

Corollary 1. VERTEX COVER *is NP-complete in C_3-free graphs.*

3 Hardness proofs

We begin by restating VERTEX COVER as a satisfiability problem in order to simplify the notation in the proofs.

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 $f \colon X \to \{\text{true, false}\}$) 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 $C = \{(u \lor w) \colon uw \in E(G)\}$ 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 1), it follows that WEIGHTED POSITIVE 2-SAT is NP-hard, too.

Let G be a graph and $S, S' \subseteq E(G)$ 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.

Corollary 2. *Let G be a graph and $S \subseteq E(G)$ a minimal α-contraction-critical set of edges. Then, $G\big|_S$ is a forest.*

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

Proof. It was shown in [9] 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 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 $G_x = \{v_x\} \cup K_x$, where 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 Figure 1 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. If 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.
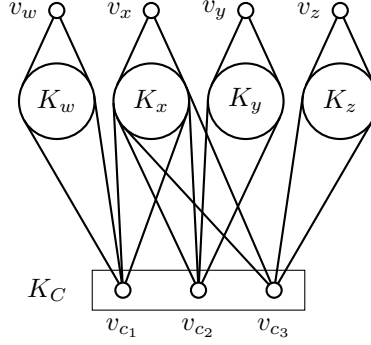

Fig. 2. This is the graph corresponding to the instance of WEIGHTED POSITIVE 2-SAT given by the variables w, x, y, z and the clauses $c_1 = w \vee x, c_2 = x \vee y$ and $c_3 = x \vee z$. The rectangular box corresponds to $G[K_C]$, the vertices contained in it induce a clique. Every set K_i induces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

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 $\{v_x \colon x \in X\} \cup \{v_c\}$ 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_p be the set of positive variables of a satisfying assignment of Φ. For each $x \in X_p$, let e_x be an edge incident to v_x and let $S = \{e_x | x \in X_p\}$. Let $G' = G/S$. We claim that $\alpha(G') < \alpha(G)$. To see this, observe first that for any $x \in X_p$, 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 $G' \simeq G - \{v_x \colon x \in X_p\}$. 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 $x \in X_p$) and $|I \cap G_x| \leq 1$ (for all $x \in X \setminus X_p$), 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_p, it follows that there is $x \in X_p$ 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 2, 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 $E(H) = \{uv \in S : u, v \in K_C\}$.

Suppose for a contradiction that there is a connected component T of H such that for every $x \in X$ with $\mathrm{dist}_G(G_x, V(T)) = 1$ we have $G_x \cap V(S) = \varnothing$. In other words, for every $c = (x \vee y) \in C$ with $v_c \in V(T)$ we have $G_x \cap V(S) = G_y \cap V(S) = \varnothing$. 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 $\mathrm{dist}_G(G_x, V(T)) = 1$, we have that $G_x \cap V(S) = \varnothing$ and thus $\{v_x\}$ is a connected component in $G\big|_S$. Let $X_1 = \{x \in X : \mathrm{dist}_G(u_x, V(T)) = 1\}$ and $X_2 = X \setminus X_1$. Observe that the set $I = T \cup \{\{v_x\} : x \in X_1\} \cup \{\{u_x\} : x \in X_2\}$ is a set of connected components of $G\big|_S$ which correspond to vertices in G/S who are pairwise 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 $\mathrm{dist}_G(G_x, V(T)) = 1$ we have $G_x \cap V(S) = \varnothing$.

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 $G_x \cap V(S) \neq \varnothing$. 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 $G_x \cap V(S) \neq \varnothing$ is at most 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.

Interestingly, 1-Deletion Blocker(α) and 1-Contraction Blocker(α) are equivalent on the instance Φ' constructed in the proof of Theorem 1 and thus the same construction can be used to show NP-hardness of 1-Deletion Blocker(α) in chordal graphs.

Theorem 2. 1-DELETION BLOCKER(α) *is* NP-*complete in chordal graphs.*

Proof. It has been shown in [9] 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 1 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 1. 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_p is the set of positive variables in a satisfying assignment of Φ. We have shown in the proof of Theorem 1 that $\alpha(G - \{v_x \colon x \in X_p\}) < \alpha(G)$, 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 is $u_x \in K_x \setminus W$, since $|W| < |K_x|$. Define a set $Z = \{x \in X \colon v_x \in W\}$ 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. 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 $\{v_c, v_x, v_y\} \cup \{u_z \colon z \in X \setminus \{x, y\}\}$ is an independent set of size $|X| + 1$ in $G - W$, a contradiction to the α-deletion-criticalness of W. Hence the assignment is satisfying and the theorem follows.

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 [5] that 1-DELETION BLOCKER(ω) is NP-hard in perfect graphs the corollary follows.

The last theorem in this section answers a question asked in [5]. Indeed, Theorem 4 settles the missing case of [5, Theorem 24] and completes the complexity dichotomy for H-free graphs, which is as follows.

Theorem 3. *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.*

Theorem 4. *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 1. Let (G, k) be an instance of VERTEX COVER where G is a C_3-free graph. Since VERTEX COVER is trivial to solve on 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'.

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 $\{v_1, \ldots, v_k\} \subseteq V(G)$ be a vertex cover of G. Set $S = \{v_i w : i \in \{1, \ldots, k\}\}$ 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 $\{v_1, \ldots, v_k\}$ is a minimum vertex cover of G, there are no edges in $G^* - w$, meaning 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 2), 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^* and since the 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.

4   Algorithms

In this section we give a polynomial-time algorithm for d-CONTRACTION BLOCKER(α) in bipartite graphs.

Theorem 5. *Let G be a connected, bipartite graph with $|V(G)| \geq 2d + 2$ and $\alpha(G) \geq d + 1$, 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 $|V(G)| \geq 2d + 2$ and $\alpha(G) \geq d + 1$. Let M be a maximum matching of G. Since G is connected, M is non-empty. Consider the following algorithm which constructs a tree T, which is a subgraph of G.

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 $uu' \in E(M)$.
2: Set $V(T) = \{u, u'\}, E(T) = \{uu'\}$.
    while $|E(T)| \leq 2d - 1$ **do**
4:      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**
6:        Let $v \in V(M)$ s.t. $vw \in E(M)$.
        $V(T) = V(T) \cup \{v, w\}, E(T) = E(T) \cup \{w'w, vw\}$
8:      **else** $V(T) = V(T) \cup \{w\}, E(T) = E(T) \cup \{w'w\}$
      end if
10: **end while**
    return T

We claim that the resulting graph T is a tree. Indeed, the initial graph is a single edge and thus a tree. Further, observe that every time there are vertices and edges added to T in lines 7 or 8, the resulting graph remains connected and 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 $\alpha(G^*) \leq \alpha(G') + 1 \leq \alpha(G) - d$.

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
    for every $S \subseteq E(G)$ of size at most k **do**
2:      Let $\beta = 0$.
      Let $G' = G/S$.
4:      Let $U = \big\{ v \in V(G') : v$ *corresponds to a connected component of* $G\big|_S$
                                         which contains at least 2 vertices $\big\}$.
      for every subset $U' \subseteq U$ **do**
6:         **if** U' is independent **then**
            $\beta = \max(\beta, \alpha(G' - (U \cup N_{G'}(U'))) + |U'|)$
8:        **end if**
      end for
10:    **if** $\beta \leq \alpha(G) - d$ **then**
      return YES
12:    **end if**
    end for
14: **return** NO

Theorem 6. *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).

    Since contracting edges in a non-empty graph cannot reduce the number of vertices to zero, it follows that if $\alpha(G) \leq d$ it is not possible to reduce $\alpha(G)$

by d via edge-contractions. Hence, we can assume that $|V(G)| \geq 2d + 2$ and $\alpha(G) \geq d + 1$. By Theorem 5, we know that for $k \geq 2d + 1$, it is always possible to contract at most k edges to reduce the independence number of G by at least d, so we can further assume that $k \leq 2d$.

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 $G' = G/S$ and 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.

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. Bentz, C., Costa, M.C., de Werra, D., Picouleau, C., Ries, B.: Minimum d-transversals of maximum-weight stable sets in trees. Electronic Notes in Discrete Mathematics **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
3. Cerioli, M.R., Lima, P.T.: Intersection of longest paths in graph classes. Discrete Applied Mathematics **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
4. Diestel, R.: Graph Theory, p. 37. Springer-Verlag, Heidelberg (2016)
5. Diner, Ö.Y., Paulusma, D., Picouleau, C., Ries, B.: Contraction and deletion blockers for perfect graphs and h-free graphs. Theoretical Computer Science **746**, 49–72 (2018). https://doi.org/10.1016/j.tcs.2018.06.023
6. Galby, E., Mann, F., Ries, B.: Blocking total dominating sets via edge contractions. Theoretical Computer Science **877**, 18–35 (2021). https://doi.org/10.1016/j.tcs.2021.03.028
7. Galby, E., Mann, F., Ries, B.: Reducing the domination number of (p3+kp2)-free graphs via one edge contraction. Discrete Applied Mathematics **305**, 205–210 (2021). https://doi.org/10.1016/j.dam.2021.09.009
8. 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, https://arxiv.org/abs/2107.03755

9. Gavril, F.: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM Journal on Computing **1**(2), 180–187 (1972). https://doi.org/10.1137/0201013
10. Paik, D., Reddy, S., Sahni, S.: Deleting vertices to bound path length. IEEE Transactions on Computers **43**(9), 1091–1096 (1994). https://doi.org/10.1109/12.312117
11. 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)
12. Picouleau, C., Paulusma, D., Ries, B.: Reducing the chromatic number by vertex or edge deletions. Electronic Notes in Discrete Mathematics **62**, 243–248 (2017). https://doi.org/10.1016/j.endm.2017.10.042, lAGOS'17 – IX Latin and American Algorithms, Graphs and Optimization
13. Poljak, S.: A note on stable sets and colorings of graphs. Commentationes Mathematicae Universitatis Carolinae **15**(2), 307–309 (1974)
14. Zenklusen, R., Ries, B., Picouleau, C., de Werra, D., Costa, M.C., Bentz, C.: Blockers and transversals. Discrete Mathematics **309**(13), 4306–4314 (2009). https://doi.org/10.1016/j.disc.2009.01.006