Treewidth reduction for constrained separation and bipartization problems

Dániel Marx[1], Barry O'Sullivan[2], and Igor Razgon[2]

[1] Department of Computer Science and Information Theory, Budapest University of Technology and Economics, dmarx@cs.bme.hu
[2] Cork Constraint Computation Centre, University College Cork, {b.osullivan,i.razgon}@cs.ucc.ie

Abstract. We present a method for reducing the treewidth of a graph while preserving all the minimal $s-t$ separators. This technique turns out to be very useful in the design of parameterized algorithms. We prove the fixed-parameter tractability of the $s-t$ Cut, Multicut, and Bipartization problems (parameterized by the maximal number k of vertices being removed) with various additional restrictions (e.g., the vertices being removed from the graph form an independent set). These results answer a number of open questions in the area of parameterized complexity.

1 Introduction

The main technical contribution of the present paper is a theorem stating that given a graph G, two terminal vertices s, t, and a parameter k, we can compute in a FPT-time a graph G^* having the treewidth bounded by a function of k while (roughly speaking) preserving all the minimal $s-t$ separators of size at most k (recall that an FPT-time algorithm has running time $f(k) \cdot n^{O(1)}$ for some function f depending only on k). Combining this theorem with the well-known Courcelle's Theorem, we prove the fixed-parameter tractability of a wide variety of *constrained* separation and bipartization problems, answering a number of open questions in the area of parameterized complexity.

In particular, we consider 'meta-problems' that we call $\mathcal{G}$-MINCUT and $\mathcal{G}$-BIPARTIZATION. The task of the former is, given a graph G and parameter k, to check whether there is a set $C \subseteq V(G)$ of size at most k that separates given terminals s, t and induces a graph belonging to class $\mathcal{G}$. The task of the latter is to find out whether there is set $C \subseteq V(G)$ with $|C| \leq k$, $G[C] \in \mathcal{G}$ such that the removal of C makes G bipartite. We prove that both problems are FPT provided that $\mathcal{G}$ is hereditary (i.e. whenever a graph belongs to $\mathcal{G}$, all its induced subgraphs do) and decidable. Setting $\mathcal{G}$ to be the class of all graphs without edges immediately implies that the STABLE SEPARATION problem (are there at most k independent vertices whose removal separates s and t?) as well as STABLE BIPARTIZATION problem (are there at most k independent vertices whose removal makes G bipartite?) are both FPT, answering the open questions posed by Kanj [11] and Fernau [4]. More elaborated arguments show that it is FPT to check whether there are at most k edges such that removal of their endpoints separates s and t, answering the open question posed by Samer and Szeider in [17] and by Samer in [4] and that it is FPT to check the existence of *exactly* k independent vertices whose removal makes the graph bipartite, answering an open question posed by Díaz et al. [5].

Finally, we analyze the constrained bipartization problems in a more general environment of $(H, C, \leq K)$-coloring [5], where the parameter is the maximum number of vertices mapped to C in the homomorphism and prove that the problem is FPT provided that $H - C$ consists of two adjacent vertices without loops.

The proposed results are related to two directions of investigation in the area of parameterized complexity. The first direction is understanding the fixed-parameter tractability of graph separation problems, mainly various versions of the MULTICUT problem, e.g. [12, 9, 7, 2]. The second direction is applying the ideas from the area of graph separation to design parameterized algorithms for problems from other areas, e.g. [15, 3, 13]. The technique of reduction from bipartization to a graph separation problem proposed in [15] serves in the present paper as a bridge between the results related to the above two directions.

The paper assumes the knowledge of the definition of treewidth and its algorithmic use, including Courcelle's Theorem (see the surveys [1, 8]).

2 Treewidth reduction

We present the main combinatorial result of the paper in this section. Two slightly different notion of separation will be used:

Definition 1. *We say that a set S of vertices* separates *sets A and B if no component of $G \setminus S$ contains vertices from both $A \setminus S$ and $B \setminus S$. If s and t are two distinct vertices of G, then an $s - t$* separator *is a set S of vertices disjoint from $\{s, t\}$ such that s and t are in different components of $G \setminus S$.*

In particular, if S separates A and B, then $A \cap B \subseteq S$. Furthermore, given a set W of vertices, we say that a set S of vertices is a *balanced separator* of W if $|W \cap C| \leq |W|/2$ for every connected component C of $G \setminus S$. A *k-separator* is a separator S with $|S| = k$. The treewidth of a graph is closely connected with the existence of balanced separators:

Lemma 2 ([14], [6, Section 11.2]).

1. *If $G(V, E)$ has treewidth greater than $3k$, then there is a set $W \subseteq V$ of size $2k + 1$ having no balanced k-separator.*
2. *If $G(V, E)$ has treewidth at most k, then every $W \subseteq V$ has a balanced $(k+1)$-separator.*

Note that the contrapositive of (1) in Lemma 2 says that if every W has a balanced k-separator, then the treewidth is at most $3k$.

Lemma 3. *Let G be a graph, $C_1, \ldots, C_r$ subsets of vertices, and let $C := \bigcup_{i=1}^{r} C_i$. Suppose that every $W_i \subseteq C_i$ has a balanced separator $S_i \subseteq C_i$ of size at most w. Then every $W \subseteq C$ has a balanced separator $S \subseteq C$ of size wr.*

Proof. For a given $W \subseteq C$, let us define $W_i := (W \cap C_i) \setminus (\bigcup_{j=1}^{i-1} C_i)$; it is clear that the W_i's form a partition of W. Let S_i be the separator corresponding to W_i. Let $S := \bigcup_{i=1}^{r} S_i$. Each component of $G \setminus S$ contains at most $|W_i|/2$ vertices of W_i, thus each component contains at most $|W|/2$ vertices of W. □

When we are reducing a problem to an 'important' subset C, we have to introduce additional edges to account for connections via vertices not in C:

Definition 4. *Let G be a graph and $C \subseteq V(G)$. The graph* torso(G, C) *has vertex set C and two vertices $a, b \in C$ are connected by an edge if $\{a, b\} \in E(G)$ or there is a path P in G connecting a and b whose internal vertices are not in C.*

Proposition 5. *Let $C_1 \subseteq C_2$ be two subsets of vertices in G and let $a, b \in C_1$ two vertices. A set $S \subseteq C_1$ separates a and b in* torso(G, C_1) *if and only if S separates these vertices in* torso(G, C_2)*. (By setting $C_2 = V(G)$, we obtain a special case where* torso(G, C_2) *is replaced by G.)*

Analogously to Lemma 3, we can show that if we have a bound on torso(G, C_i) for every i, then these bounds add up for the union of the C_i's.

Lemma 6. *Let G be a graph and $C_1, \ldots, C_r$ be subsets of $V(G)$ such that for every $1 \le i \le r$, the treewidth of* torso(G, C_i) *is at most w. Then the treewidth of* torso(G, C) *for $C := \bigcup_{i=1}^{r} C_i$ is at most $3r(w + 1)$.*

If the minimum size of an $s - t$ separator is ℓ, then the *excess* of an $s - t$ separator S is $|S| - \ell$ (which is always nonnegative). Note that if s and t are adjacent, then no $s-t$ separator exists, and in this case we say that the minimum size of an $s - t$ separator is ∞. If X is a set of vertices, we denote by $\delta(X)$ the set of those vertices in $V(G) \setminus X$ that are adjacent to at least one vertex of X.

Lemma 7. *Let s, t be two vertices in graph G such that the minimum size of an $s - t$ separator is k. Then there is a collection $\mathcal{X} = \{X_1, \ldots X_q\}$ of sets where $\{s\} \subseteq X_i \subseteq V(G) \setminus (\{t\} \cup \delta(\{t\}))$ $(1 \le i \le q)$, such that*

1. *$X_1 \subset X_2 \subset \cdots \subset X_q$,*
2. *$|\delta(X_i)| = k$ for every $1 \le i \le q$, and*
3. *every $s - t$ separator of size k is fully contained in $\bigcup_{i=1}^{q} \delta(X_i)$.*

Furthermore, such a collection $\mathcal{X}$ can be found in polynomial time.

Proof. Let $\mathcal{X} = \{X_1, \ldots, X_q\}$ be a collection of sets such that (2) and (3) holds. Let us choose the collection such that q is minimum possible, and among such collections, $\sum_{i=1}^{q} |X_i|^2$ is maximum possible. We show that for every i, j, either $X_i \subset X_j$ or $X_j \subset X_i$ holds, thus the sets can be ordered such that (1) holds.

Suppose that neither $X_i \subset X_j$ nor $X_j \subset X_i$ holds for some i and j. We show that after replacing X_i and X_j in $\mathcal{X}$ with the two sets $X_i \cap X_j$ and $X_i \cup X_j$, properties (2) and (3) still hold, and the resulting collection $\mathcal{X}'$ contradicts the optimal choice of $\mathcal{X}$. The function δ is well-known to be submodular, i.e.,

$$|\delta(X_i)| + |\delta(X_j)| \ge |\delta(X_i \cap X_j)| + |\delta(X_i \cup X_j)|.$$

Both $\delta(X_i \cap X_j)$ and $\delta(X_i \cup X_j)$ are $s-t$ separators and hence have size at least k. The left hand side is $2k$, hence there is equality and $|\delta(X_i \cap X_j)| = |\delta(X_i \cup X_j)| = k$ follows. This means that property (2) holds after the replacement. Observe that $\delta(X_i \cap X_j) \cup \delta(X_i \cup X_j) \subseteq \delta(X_i) \cup \delta(X_j)$: any edge that leaves $X_i \cap X_j$ or $X_i \cup X_j$ leaves either X_i or X_j. We show that there is equality here, implying that property (3) remains true after the replacement. It is easy to see that $\delta(X_i \cap X_j) \cap \delta(X_i \cup X_j) \subseteq \delta(X_i) \cap \delta(X_j)$, hence we have

$$\begin{aligned}|\delta(X_i \cap X_j) \cup \delta(X_i \cup X_j)| &= 2k - |\delta(X_i \cap X_j) \cap \delta(X_i \cup X_j)| \\ &\geq 2k - |\delta(X_i) \cap \delta(X_j)| = |\delta(X_i) \cup \delta(X_j)|,\end{aligned}$$

showing the required equality.

If $X_i \cap X_j$ or $X_i \cup X_j$ was already present in $\mathcal{X}$, then the replacement decreases the size of the collection, contradicting the choice of $\mathcal{X}$. Otherwise, we have that $|X_i|^2 + |X_j|^2 < |X_i \cap X_j|^2 + |X_i \cup X_j|^2$ (to verify this, simply represent $|X_i|$ as $|X_i \cap X_j| + |X_i \setminus X_j|$, $|X_j|$ as $|X_i \cap X_j| + |X_j \setminus X_i|$, $|X_i \cup X_j|$ as $|X_i \cap X_j| + |X_i \setminus X_j| + |X_j \setminus X_i|$ and do direct calculation having in mind that both $|X_i \setminus X_j|$ and $|X_j \setminus X_i|$ are greater than 0), again contradicting the choice of $\mathcal{X}$. Thus an optimal collection $\mathcal{X}$ satisfies (1) as well. The polynomial time algorithm for computing $\mathcal{X}$ is described in the Appendix. □

Lemma 8. *Let s,t be two vertices of graph G and let ℓ be the minimum size of an $s-t$ separator. For some $e \geq 0$, let C be the union of all minimal $s-t$ separators having* excess *at most e (i.e. of size at most $\ell + e$). Then there is an $O(f(\ell,e) \cdot |V(G)|^d)$ time algorithm that returns a set $C' \supseteq C \cup \{s,t\}$ such that the treewidth of* torso(G, C') *is at most $g(\ell, e)$, for some constant d and functions f and g depending only on ℓ and e.*

Proof. We prove the lemma by induction on e. Consider the collection $\mathcal{X}$ of Lemma 7 and define $S_i := \delta(X_i)$ for $1 \leq i \leq q$. For the sake of uniformity, we define $X_0 := \emptyset$, $X_{q+1} := V(G) \setminus \{t\}$, $S_0 := \{s\}$, $S_{q+1} := \{t\}$. For $1 \leq i \leq q+1$, let $L_i := X_i \setminus (X_{i-1} \cup S_{i-1})$. Also, for $1 \leq i \leq q+1$ and two disjoint *non-empty* subsets A, B of $S_i \cup S_{i-1}$, we define $G_{i,A,B}$ to be the graph obtained from $G[L_i \cup A \cup B]$ by contracting the set A to a vertex a and the set B to a vertex b. Taking into account that if C includes a vertex of some L_i then $e > 0$, we prove the key observation that makes it possible to use induction.

Claim. If a vertex $v \in L_i$ is in C, then there are two disjoint non-empty subsets A, B of $S_i \cup S_{i-1}$ such that v is part of a minimal $a-b$ separator K_2 in $G_{i,A,B}$ having size at most k and excess at most $e-1$.

Proof. Suppose that there is a minimal $s-t$ separator K of size at most k that contains v. Let $K_1 := K \setminus L_i$ and $K_2 := K \cap L_i$. Partition $(S_i \cup S_{i-1}) \setminus K$ into the set A of vertices reachable from s in $G \setminus K$ and the set B of vertices non-reachable from s in $G \setminus K$. Observe that both A and B are non-empty. Indeed, due to the minimality of K, G has a path P from s to t such $V(P) \cap K = \{v\}$. By selection of v, S_{i-1} separates v from s and S_i separates v from t. Therefore, at least one vertex u of S_{i-1} occurs in P before v and at least one vertex w of S_i occurs in P after v. The prefix of P ending at u and suffix of P starting at w are both subpaths in $G \setminus K$. It follows that u is reachable from s in $G \setminus K$, i.e. belongs to A and that w is reachable from t in $G \setminus K$, hence non-reachable from s and thus belongs to B.

To see that K_2 is an $a - b$ separator in $G_{i,A,B}$, suppose that there is a path P connecting a and b in $G_{i,A,B}$ avoiding K_2. Then there is a corresponding path P' in G connecting a vertex of A and a vertex of B. Path P' is disjoint from K_1 (since it contains vertices of L_i and $(S_i \cup S_{i-1}) \setminus K$ only) and from K_2 (by construction). Thus a vertex of B is reachable from s in $G \setminus K$, a contradiction.

To see that K_2 is a minimal separator, suppose that there is a vertex $u \in K_2$ such that $K_2 \setminus \{u\}$ is also an $a - b$ separator in $G_{i,A,B}$. Since K is minimal, there is an $s - t$ path P in $G \setminus (K \setminus u)$, which has to pass through u. Arguing as when we proved that A and B are non-empty, we observe that P includes vertices of both A and B, hence we can consider a minimal subpath P' of P between a vertex $a' \in A$ and a vertex $b' \in B$. We claim that all the internal vertices of P' belong to L_i. Indeed, due to the minimality of P', an internal vertex of P' can belong either to L_i or to $V(G) \setminus (K_1 \cup L_i \cup S_{i-1} \cup S_i)$. If all the internal vertices of P' are from the latter set then there is a path from a' to b' in $G \setminus (K_1 \cup L_i)$ and hence in $G \setminus (K_1 \cup K_2)$ in contradiction to $b' \in B$. If P' contains internal vertices of both sets then G has an edge $\{u, w\}$ where $u \in L_i$ while $w \in V(G) \setminus (K_1 \cup L_i \cup S_{i-1} \cup S_i)$. But this is impossible since $S_{i-1} \cup S_i$ separates L_i from the rest of the graph. Thus it follows that indeed all the internal vertices of P' belong to L_i. Consequently, P' corresponds to a path in $G_{i,A,B}$ from a to b that avoids $K_2 \setminus u$, a contradiction that proves the minimality of K_2.

Finally, we have to show that K_2 has excess at most $e - 1$. Let K_2' be a minimum $a - b$ separator in $G_{i,A,B}$. Observe that $K_1 \cup K_2'$ is an $s - t$ separator in G. Indeed, consider a path P in $G \setminus (K_1 \cup K_2')$. It necessarily contains a vertex $u \in K_2$, hence arguing as in the previous paragraph we notice that P includes vertices of both A and B. Considering a minimal subpath P' of P between a vertex $a' \in A$ and $b' \in B$ we observe, analogously to the previous paragraph that all the internal vertices of this path belong to L_i. Hence this path correspond to a path between a and b in $G_{i,A,B}$. It follows that P', and hence P, includes a vertex of K_2', a contradiction showing that $K_1 \cup K_2'$ is indeed an $s - t$ separator in G. Due to the minimality of K_2, $K_2' \neq \emptyset$. Thus $K_1 \cup K_2'$ contains at least one vertex from L_i, implying that $K_1 \cup K_2'$ is not a minimum $s - t$ separator in G. Thus $|K_2| - |K_2'| = (|K_1| + |K_2|) - (|K_1| + |K_2'|) < k - \ell = e$, as required. □

Now we define C'. Let $C_0 := \bigcup_{i=0}^{q+1} S_i$. For $e = 0$, $C' = C_0$. Assume that $e > 0$. For $1 \leq i \leq q + 1$ and disjoint non-empty subsets A, B of $S_i \cup S_{i-1}$, let $C_{i,A,B}$ be the union of all minimal $a - b$ separators of size at most k and excess at most $e - 1$ in $G_{i,A,B}$. We define C' as the union of C_0 and all sets $C_{i,A,B}$ as above. Observe that C' is defined correctly in the sense that any vertex v participating in a $s - t$ minimal separator of size at most k indeed belongs to C'. For $e = 0$, the correctness of C' follows from definition of sets S_i. For $e > 0$, the correctness follows from the above Claim if we take into account that since $\bigcup_{i=1}^{q+1} L_i \cup C_0 = V(G)$, v belongs to some L_i.

We shall show that the treewidth of torso(G, C') is at most $g(\ell, e)$, a function recursively defined as follows: $g(\ell, 0) := 6\ell$ and $g(\ell, e) := 3 \cdot (2\ell + 3^{2\ell} \cdot (g(\ell, e-1)+1))$ for $e > 0$. We do this by showing that in graph G, every set $W \subseteq C'$ has a balanced separator of size at most 2ℓ (for $e = 0$) and at most $2\ell + 3^{2\ell} \cdot (g(\ell, e-1)+1)$ (for $e > 0$). By Proposition 5, it will imply that in torso(G, C'), W has a balanced separator with the same upper bound. By Lemma 2(1), the desired upper bound on the treewidth will immediately follow.

Let $W \subseteq C'$ be an arbitrary set. Let $1 \leq i \leq q+1$ be the smallest value such that $|W \cap X_i| \geq |W|/2$. Consider the separator $S_i \cup S_{i-1}$ (whose size is at most 2ℓ). In $G \setminus (S_i \cup S_{i-1})$, the sets X_{i-1}, L_i, and $V(G) \setminus (S_i \cup S_{i-1} \cup X_{i-1} \cup L_i)$ are pairwise separated from each other. By selection of i, the first and the third sets do not contain more than half of W. If $e = 0$, then C' is disjoint with L_i, hence the treewidth upper bound follows for $e = 0$. We assume that $e > 0$ and, using the induction assumption, will show that $W \cap L_i$ has a balanced separator S of size at most $3^{2\ell} \cdot (g(\ell, e-1)+1)$. This will immediately imply that $S \cup S_i \cup S_{i-1}$ is a balanced separator of W of size at most $2\ell + 3^{2\ell} \cdot (g(\ell, e-1)+1)$, which, in turn, will imply the desired upper bound on the treewidth of torso(G, C').

By the induction assumption, the treewidth of torso$(G_{i,A,B}, C_{i,A,B})$ is at most $g(\ell, e-1)$ for any pair of disjoint subsets A, B of $S_i \cup S_{i-1}$ such that $G_{i,A,B}$ has an $a-b$ separator of size at most k. By the combination of Lemma 2(2) and Proposition 5 G, has a balanced separator of size at most $(g(\ell, e-1)+1)$ for any set $W_{i,A,B} \subseteq C_{i,A,B}$. Let C^* be the union of $C_{i,A,B}$ for all such A and B Taking into account that the number of choices of A and B is at most $3^{2\ell}$, for any $W^* \subseteq C^*$, G has a balanced separator of size at most $3^{2\ell} \cdot (g(\ell, e-1)+1)$ according to Lemma 3. By definition of C', $W \cap L_i \subseteq C^*$, hence the existence of the desired separator S follows. The running time analysis can be found in the appendix. ◻

Theorem 9. *Let G be a graph, $S \subseteq V(G)$, and let k be an integer. Let C be the set of all vertices of G participating in a minimal $s-t$ cut for some $s, t \in S$. Then there is an* FPT *algorithm, parameterized by k and $|S|$, that computes a graph G^* having the following properties:*

1. *$C \cup S \subseteq V(G^*)$*
2. *For every $s, t \in S$, a set $K \subseteq V(G^*)$ with $|K| \leq k$ is a minimal $s-t$ separator of G^* if and only if $K \subseteq C \cup S$ and K is a minimal $s-t$ separator of G.*
3. *The treewidth of G^* is at most $h(k, |S|)$ for some function h.*
4. *For any $K \subseteq C$, $G^*[K]$ is isomorphic to $G[K]$.*

Proof. For every $s, t \in S$, the algorithm of Lemma 8 computes a set $C'_{s,t}$ containing all the minimal $s-t$ separators of size at most k. By Lemma 6, if C' is the union of these $\binom{|S|}{2}$ sets, then $G' = $ torso(G, C') has treewidth bounded by a function of k and $|S|$. Note that G' satisfies all the requirements of the theorem except the last one: two vertices of C' non-adjacent in G may become adjacent in G' (see Definition 4). To fix this problem we subdivide each edge $\{u, v\}$ of G'

such that $\{u,v\} \notin E(G)$ into two edges add a vertex between them, and, to avoid selection of this vertex into a cut, we split it into $k+1$ copies. In other words, for each edge $\{u,v\} \in E(G')\setminus E(G)$ we introduce $k+1$ new vertices $w_1,\dots,w_{k+1}$ and replace $\{u,v\}$ by the set of edges $\{\{u,w_1\}\dots\{u,w_{k+1}\},\{w_1,v\},\dots,\{w_{k+1},v\}\}$. Let G^* be the resulting graph. It is not hard to check that G^* satisfies all the properties of the present theorem. □

3 Constrained separation problems

Let $\mathcal{G}$ be a class of graphs. Given a graph G, vertices s, t, and parameter k, the $\mathcal{G}$-MINCUT problem asks whether G has a $s-t$ separator of size at most k such that $G[C] \in \mathcal{G}$. The following theorem is the central result of this section.

Theorem 10. *Assume that* $\mathcal{G}$ *is* decidable *and* hereditary *(i.e. whenever* $G \in \mathcal{G}$ *then for any* $V' \subseteq V$, $G[V'] \in \mathcal{G}$*). Then the* $\mathcal{G}$-MINCUT *problem is* FPT.

Proof. Let G^* be a graph satisfying the requirements of Theorem 9 for $S=\{s,t\}$. According to Theorem 9, G^* can be computed in a FPT time. We claim that (G,s,t,k) is a 'YES' instance of the $\mathcal{G}$-MINCUT problem if and only if (G^*,s,t,k) is a 'YES' instance of this problem. Indeed, let K be an $s-t$ separator in G such that $|K| \le k$ and $G(K) \in \mathcal{G}$. Since $\mathcal{G}$ is hereditary, we may assume that K is minimal (otherwise we may consider a minimal subset of K separating s from t). By the second and fourth properties of G^* (see Theorem 9), K separates s from t in G^* and $G^*[K] \in \mathcal{G}$. The opposite direction can be proved similarly.

Thus we have established a FPT-time reduction from an instance of the $\mathcal{G}$-MINCUT problem to another instance of this problem where the treewidth is bounded by a function of parameter k. Now, let $G_1 = (V(G^*), E(G^*), ST)$ be a labeled graph where $ST = \{s,t\}$. We present an algorithm for construction of a monadic second-order (MSO) formula φ whose atomic predicates (besides equality) are $E(x_1,x_2)$ (showing that x_1 and x_2 are adjacent in G^*) and predicates of the form $X(v)$ (showing that v is contained in $X \subseteq V$), whose size is bounded by a function of k, and $G_1 \models \varphi$ if and only if (G^*,s,t,k) is a 'YES' instance of the $\mathcal{G}$-MINCUT problem. According to a restricted version of the well-known Courcelle's Theorem (see the survey article of Grohe [8], Remarks 3.19[3] and 3.20), it will follow that the $\mathcal{G}$-MINCUT problem is FPT. The detailed construction is postponed to the Appendix. □

Theorem 10 allows to answer two open questions in the area of parameterized complexity. In particular, let $\mathcal{G}^0$ be the class of all graphs without edges. Then $\mathcal{G}^0$-MINCUT is the Minimum Stable Cut problem whose fixed-parameter tractability has been posed as an open question by Kanj [11]. Clearly $\mathcal{G}^0$ is hereditary and hence the $\mathcal{G}^0$-MINCUT is FPT.

[3] Although the branchwidth of G_1 appears in the parameter, it can be replaced by the treewidth of G_1 since the former is bounded by a function of k if and only if the latter is [16]

Samer and Szeider [17] introduced the notion of *edge-induced vertex-cut* and the corresponding computational problem: given a graph G and two vertices s and t, the task is to find out if there are k edges such that deleting the *endpoints* of these edges separates s and t. It remained an open question in [17] whether this problem is FPT. Samer reposted this problem as an open question in [4]. We answer this question positively.

Corollary 11. *The* EDGE-INDUCED VERTEX-CUT *problem is* FPT.

Proof. (Sketch) Let $\mathcal{G}_k$ contain those graphs where the number of vertices minus the size of the maximum matching is at most k. It is not hard to observe that $\mathcal{G}_k$ is hereditary by noticing that for any $H \in \mathcal{G}_k$ and $v \in V(H)$ the difference between the number of vertices and the size of maximum matching does not increase by removal of v. It follows from Theorem 10 that $\mathcal{G}_k$-MINCUT is FPT.

We may assume w.l.o.g. that G does not have isolated vertices (if there are, they can be safely removed before the run of our algorithm). Then we show that the $\mathcal{G}_k$-MINCUT with parameter $2k$ is equivalent to the problem of finding out whether s can be separated from t by removal of a set S that *can be extended to the union of at most k edges*. Taking into account that the latter problem is an equivalent reformulation of the EDGE-INDUCED VERTEX-CUT problem, this will complete the present proof. □

MULTICUT is the generalization of MINCUT where, instead of s and t, the input contains a set $(s_1, t_1), \ldots, (s_\ell, t_\ell)$ of terminal pairs. The task is to find a set S of at most k nonterminal vertices that separate s_i and t_i for every $1 \le i \le \ell$. MULTICUT is known to be FPT [12, 18] parameterized by k and ℓ. In the $\mathcal{G}$-MULTICUT problem, we additionally require that S induces a graph from $\mathcal{G}$. It is not difficult to generalize Theorem 10 for $\mathcal{G}$-MULTICUT.

Theorem 12. *Assume that $\mathcal{G}$ is* decidable *and* hereditary*. Then $\mathcal{G}$-*MULTICUT *is* FPT *parameterized by k and ℓ.*

We generalize Theorem 12 one more step further. In the $\mathcal{G}$-MULTICUT-UNCUT problem the input contains an additional integer ℓ', and we change the problem by requiring for every $\ell' \le i \le \ell$ that S *does not* separate s_i and t_i.

Theorem 13. *Assume that $\mathcal{G}$ is* decidable *and* hereditary*. Then $\mathcal{G}$-*MULTICUT-UNCUT *is FPT parameterized by k and ℓ.*

We close this section with a very simple hardness result. Theorem 10 can be used to decide if there is an $s - t$ separator of size *at most* k having a certain property, but cannot be used if we are looking for $s - t$ separators of size *exactly* k. We argue that some of these problems actually become hard if the size has to be exactly k. Let graph G' be obtained from graph G by introducing two isolated vertices s and t. Now there is a k-clique separating s and t in G' if and only if there is a k-clique in G, implying that finding such a separator is W[1]-hard. The same argument (with minor modifications) can be applied for other properties as well.

Theorem 14. *It is W[1]-hard (parameterized by k) to decide if there is a clique (or independent set, dominating set) of size exactly k separating s and t in G.*

4 Constrained Bipartization Problems and (H, C, K)-coloring

Reed et al. [15] solved a longstanding open question by proving the fixed-parameter tractability of the BIPARTIZATION problem: given a graph G and an integer k, find a set S of at most k vertices such that $G \setminus S$ is bipartite. In fact, they showed that the BIPARTIZATION problem can be solved by at most 3^k applications of a procedure solving MINCUT. The key result that allows to transform BIPARTIZATION to a separation problem is the following lemma.

Lemma 15. *Let G be bipartite graph and let (B', W') be a 2-coloring of the vertices. Let B and W be two subsets of $V(G)$. Then for any S, $G \setminus S$ has a 2-coloring where $B \setminus S$ is black and $W \setminus S$ is white if and only if S separates $X := (B \cap B') \cup (W \cap W')$ and $Y := (B \cap W') \cup (W \cap B')$.*

Proof. In a 2-coloring of $G \setminus S$, each vertex either has the same color (call it an unchanged vertex) or the opposite color as in (B', W') (call it a changed vertex). Observe that a changed and an unchanged vertex cannot be adjacent: they have the same color either under (B', W') or under the considered coloring of $G \setminus S$. Consequently, a changed and an unchanged vertex cannot belong to the same connected component of $G \setminus S$, because this would imply existence of an edge between a changed and an unchanged vertex. If B is black and W is white in a 2-coloring of $G \setminus S$, then clearly $X \setminus S$ is unchanged and $Y \setminus S$ is changed. Thus S has to separate X and Y in G.

For the other direction, suppose that $X \setminus S$ is separated from $Y \setminus S$ in $G \setminus S$. We modify the coloring (B', W') by changing the color of every vertex that is in the same connected component of $G \setminus S$ as some vertex of Y. Since all the vertices of the same component are either all change their colors or all remain colored in the same color as in (B', W'), the resulting coloring is a proper 2-coloring of $G \setminus S$. By construction, all vertices of Y have the desired color. Since S separates X and Y, the vertices of $X \setminus S$ are unchanged and hence have the required colors as well. □

In this section we consider the $\mathcal{G}$-BIPARTIZATION problem: a generalization of the BIPARTIZATION problem where, in addition to $G \setminus S$ being bipartite, it is also required that S induces a graph belonging to a class $\mathcal{G}$.

Theorem 16. *$\mathcal{G}$-BIPARTIZATION is FPT if $\mathcal{G}$ is hereditary and decidable.*

Proof. Using the algorithm of [15], we first try to find a set S_0 of size at most k such that $G \setminus S_0$ is bipartite. If no such set exists, then clearly there is no set S satisfying the requirements. Otherwise, we branch into $3^{|S_0|}$ directions: each vertex of S_0 is removed or colored black or white. For a particular branch, let $R = \{v_1, \ldots, v_r\}$ be the vertices of S_0 to be removed and let B_0 (resp., W_0) be the vertices of S_0 having color black (resp., white) in a 2-coloring of the resulting bipartite graph. Let us call a set S such that $S \cap S_0 = R$, and $G \setminus S$ is bipartite and having a 2-coloring where B_0 and W_0 are colored black and white, respectively, a set *compatible* with (R, B_0, W_0). Clearly, (G, k) is a 'YES' instance of the $\mathcal{G}$-BIPARTIZATION problem if and only if for at least one branch corresponding to partition (R, B_0, W_0) of S_0, there is a set compatible with (R, B_0, W_0) having size at most k and such that $G[S] \in \mathcal{G}$. Clearly, we need to check only those branches where $G[B_0]$ and $G[W_0]$ are both independent sets.

We transform finding a set compatible with (R, B_0, W_0) into a separation problem. Let (B', W') be a 2-coloring of $G \setminus S_0$. Let $B = N(W_0) \setminus S_0$ and $W = N(B_0) \setminus S_0$. Let us define X and Y as in Lemma 15, i.e., $X := (B \cap B') \cup (W \cap W')$, and $Y := (B \cap W') \cup (W \cap B')$. We construct a graph G' that is obtained from G by deleting the set $B_0 \cup W_0$, adding a new vertex s adjacent with $X \cup R$, and adding a new vertex t adjacent with $Y \cup R$. Note that every $s-t$ separator in G' contains R. By Lemma 15, a set S is compatible with (R, B_0, W_0) if and only if S is an $s-t$ separator in G'. Thus what we have to decide is whether there is an $s-t$ separator S of size at most k such that $G'[S] = G[S]$ is in $\mathcal{G}$. That is, we have to solve the $\mathcal{G}$-MINCUT instance (G', s, t, k). The fixed-parameter tractability of the $\mathcal{G}$-BIPARTIZATION problem now immediately follows from Theorem 10. □

In particular, deleting an independent set of size at most k to make the graph bipartite is FPT, answering a question of Fernau [4]. Next, to answer an open question appearing in [5], we consider the related problem of deleting an indpendent set of size *exactly* k to make the graph bipartite. An obvious approach would be to find appropriate separators of size exactly k (instead of size at most k) in the algorithm of Theorem 16. However, by Theorem 14, this approach is unlikely to work. Instead, we argue that under appropriate conditions, any solution of size at most k can be extended to a independent set of size exactly k.

Theorem 17. *Given a graph G and an integer k, deciding whether G can be made bipartite by the deletion of an independent set of size exactly k is fixed-parameter tractable.*

Proof. (Sketch) It is more convenient to consider an annotated version of the problem where the independent set being deleted is a subset of a set $D \subseteq V(G)$ given as part of the input. Without the annotation, D is initially set to $V(G)$. If G is not bipartite, then the algorithm starts by finding an odd cycle C of minimum length (which is known to be doable in polynomial time). It is not difficult to see that the minimality of C implies that C is a triangle or C is chordless or every vertex not in C is adjacent to at most 2 vertices of the cycle.

If $|V(C) \cap D| = 0$, then clearly no subset of D is a solution. If $1 \leq |V(C) \cap D| \leq 3k+1$, then we branch on selection of each vertex $v \in V(C) \cap D$ into the set S of vertices being removed and apply the algorithm recursively with the parameter k being decreased by 1 and the set D being updated by removal of v and $N(v) \cap D$. If $|V(C) \cap D| > 3k + 1$, then we apply the approach of Theorem 16 to find an independent set S of size at most k whose removal makes the graph bipartite. To ensure that $S \subseteq D$, we may, for example split all vertices $v \in V(G) \setminus D$ into $k + 1$ independent copies with the same neighborhood as v. If $|S| = k$, we are done. Otherwise, $|S| = k' < k$. In this case we observe that by construction each

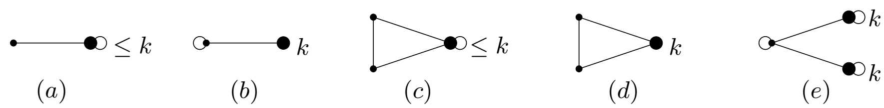

Fig. 1. (H, C, K)- (or $(H, C, \leq K)$-) coloring with these graphs is equivalent to finding (a) a vertex cover of size at most k, (b) an indepenent set of size k, (c) a bipartization set of size at most k, (d) an independent bipartization set of size exactly k, (e) a bipartite independent set of size $k + k$.

vertex of S (either in C or outside C) forbids the selection of at most 3 vertices of $V(C) \cap D$ including itself. Thus the number of vertices of $V(C) \cap D$ allowed for selection is at least $3k + 1 - 3k' = 3(k - k') + 1$. Since the cycle is chordless, we can select $k - k'$ independent vertices among them and thus complement S to being of size exactly k.

The above algorithm has a number of stopping conditions, the only non-trivial of them occurs if G is bipartite but $k > 0$. In this case we simply check if $G[D]$ has k independent vertices, which can be done in a polynomial time. □

Constrained bipartization can be also considered in terms of (H, C, K)-*coloring*. H-coloring (cf. [10]) is a generalization of ordinary vertex coloring: given graphs G and H, an H-*coloring* of G is a homomorphism $\theta : V(G) \to V(H)$, that is, if $u, v \in V(G)$ are adjacent in G, then $\theta(u)$ and $\theta(v)$ are adjacent in H (including the possibility that $\theta(u) = \theta(v)$ is a vertex with a loop). It is easy to see that a graph is k-colorable if and only if it has a K_k-coloring.

Díaz et al. [5] introduced a generalization of H-coloring where, for certain vertices $v \in V(H)$, we have a restriction on how many vertices of G can map to v. Formally, let $C \subseteq V(H)$ and let K be a mapping from C to $\mathbb{Z}^+$. An (H, C, K)-*coloring* of G is an H-coloring with the additional restriction that $|\theta^{-1}(v)| = K(v)$ for every $v \in C$. $(H, C, \leq K)$-*coloring* is the variant of the problem where we require $|\theta^{-1}(v)| \leq K(v)$, i.e., vertex v can be used *at most* $K(v)$ times. As show in Fig. 1 and discussed in [5], these colorings can express a wide range of fundamental problems such as k-independent set, k-vertex cover, bipartization, and bipartite independent set.

Following [5], we consider the parameterized version of (H, C, K)-coloring, where the parameter is $k := \sum_{v \in C} K(c)$, the number of times the cardinality constrained vertices can be used. Díaz et al. [5] started the program of characterizing the easy and hard cases of (H, C, K)- and $(H, C, \leq K)$-coloring. We make progress in this direction by showing that $(H, C, \leq K)$-coloring is FPT whenever $H - C$ consists of two adjacent vertices without loops. As this case includes Figure 1(c), it generalizes the Bipartization problem. We prove fixed-parameter tractability for an even more general problem: in *list* $(H, C, \leq K)$-*coloring* the input contains a list $L(v) \subseteq V(H)$ for each vertex $v \in V(G)$ and θ has to satisfy the additional requirement that $\theta(v) \in L(v)$ for every $v \in V(G)$.

Theorem 18. *For every fixed H, list $(H, C, \leq K)$-coloring is FPT if $H - C$ consists of two adjacent vertices without loops.*

To understand the power of Theorem 18, observe that unlike $\mathcal{G}$-bipartization, $(H, C, \leq K)$-coloring allows to handle the cases of constrained bipartization where constraints are imposed on the adjacency relation of the removed vertices with the rest of the graph: these constraints can be specified by an appropriate setting of edges between C and $H \setminus C$.

References

1. H. L. Bodlaender. Treewidth: Characterizations, applications, and computations. In *WG*, pages 1–14, 2006.
2. J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. In *WADS*, pages 495–506, 2007.
3. J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.
4. E. Demaine, G. Z. Gutin, D. Marx, and U. Stege. Seminar 07281 open problems. In *Structure Theory and FPT Algorithmics for Graphs, Digraphs and Hypergraphs*, 2007.
5. J. Díaz, M. Serna, and D. M. Thilikos. (H, C, K)-coloring: fast, easy, and hard cases. In *MFCS 2001*, pages 304–315, 2001.
6. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
7. G. Gottlob and S. T. Lee. A logical approach to multicut problems. *Inform. Process. Lett.*, 103(4):136–141, 2007.
8. M. Grohe. Logic, graphs, and algorithms (available online in http://www2.informatik.hu-berlin.de/~grohe/pub/meta-survey.pdf). In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata- History and Perspectives*. Amsterdam University Press, 2007.
9. J. Guo, F. Hüffner, E. Kenar, R. Niedermeier, and J. Uhlmann. Complexity and exact algorithms for vertex multicut in interval and bounded treewidth graphs. *European Journal of Operational Research*, 186(2):542–553, 2008.
10. P. Hell. From graph colouring to constraint satisfaction: there and back again. In *Topics in discrete mathematics*, volume 26 of *Algorithms Combin.*, pages 407–432. Springer, Berlin, 2006.
11. I. Kanj. Open problem session of Dagstuhl seminar 08431, 2008.
12. D. Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394–406, 2006.
13. I. Razgon and B. O'Sullivan. Almost 2-sat is fixed-parameter tractable (extended abstract). In *ICALP (1)*, pages 551–562, 2008.
14. B. Reed. Tree width and tangles: A new connectivity measure and some applications. In R. Bailey, editor, *Surveys in Combinatorics*, volume 241 of *LMS Lecture Note Series*, pages 87–162. Cambridge University Press, 1997.
15. B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.
16. N. Robertson and P. Seymour. Graph minors. X. obstructions to tree-decomposition. *J. Comb. Theory, Ser. B*, 52(2):153–190, 1991.
17. M. Samer and S. Szeider. Complexity and applications of edge-induced vertex-cuts. *CoRR*, abs/cs/0607109, 2006.
18. M. Xiao. Algorithms for multiterminal cuts. In *CSR*, pages 314–325, 2008.

A Proofs omitted from the main body of the paper

In this section we provide proofs and parts of proofs of all statements except Theorem 18 omitted from the main body of the paper due to space constraints. The proof of Theorem 18 is given in the next section of the Appendix.

Proof (Proposition 5). Assume first that $C_2 = V(G)$, that is $\text{torso}(G, C_2) = G$. Let P be a path connecting a and b in G and suppose that P is disjoint from a set S. The path P contains vertices from C_1 and from $V(G) \setminus C_1$. If $u, v \in C_1$ are two vertices such that every vertex of P between u and v is from $V(G) \setminus C_1$, then by definition there is an edge uv in $\text{torso}(G, C_1)$. Using these edges, we can modify P to obtain a path P' that connects a and b in $\text{torso}(G, C_1)$ and avoids S.

Conversely, suppose that P is a path connecting a and b in $\text{torso}(G, C_1)$ and it avoids $S \subseteq C_1$. If P uses an edge uv that is not present in G, then this means that there is a path connecting u and v whose internal vertices are not in C_1. Using these paths, we can modify P to obtain a path P' that uses only the edges of G. Since $S \subseteq C_1$, the new vertices on the path are not in S, i.e., P' avoids S as well.

For the general statement observe that it follows from the previous paragraph that $S \subseteq C_1$ separates a and b in $\text{torso}(\text{torso}(G, C_2), C_1)$ if and only if it separates a and b in $\text{torso}(G, C_2)$. Now the statement of the proposition immediately follows from an easy observation that $\text{torso}(\text{torso}(G, C_2), C_1) = \text{torso}(G, C_1)$. □

Proof (of Lemma 6). Let $C := \bigcup_{i=1}^{r} C_i$ and let W be an arbitrary subset of C. Since $\text{torso}(G, C_i)$ has treewidth at most w, Lemma 2(2) implies that, for every set $W_i \subseteq C_i$, $\text{torso}(G, C_i)$ has a balanced separator $S_i \subseteq C_i$ of size at most $w + 1$. By Proposition 5, it follows that S_i is balanced separator of W_i in G as well (otherwise, there are two vertices that are separated by S_i in $\text{torso}(G, C_i)$ but not separated in G). Thus the conditions of Lemma 3 hold, and W has a balanced separator $S \subseteq C$ of size at most $r(w+1)$ in G. Again by Proposition 5, the set S is a balanced separator of W in $\text{torso}(G, C)$ as well. By Lemma 2(1), it follows that $\text{torso}(G, C)$ has treewidth at most $3r(w + 1)$. □

Proof (of Lemma 7 continued). To construct $\mathcal{X}$ in polynomial time, we proceed as follows. It is easy to check in polynomial time whether a vertex v is in a minimum $s - t$ separator, and if so to produce such a separator S_v. Let X_v be the set of vertices reachable from s in $G \setminus S_v$. It is clear that X_v satisfies (2) and if we take the collection $\mathcal{X}$ of all such X_v's, then together they satisfy (3). If (1) is not satisfied, then we start doing the replacements as above. Each replacement either decreases the size of the collection or increases $\sum_{i=1}^{t} |X_i|^2$ (without increasing the collection size), thus the procedure terminates after a polynomial number of steps. □

Proof (of Lemma 8 continued). We conclude the proof by showing that the above set C' can be constructed in time $O(f(\ell, e) \cdot |V(G)|^d)$. In particular, we present an algorithm whose running time is $O(f(\ell, e) \cdot (|V(G)|-2)^d)$ (we assume that G has more than 2 vertices), where $f(\ell, e)$ is recursively defined as follows: $f(\ell, 0) = 1$ and $f(\ell, e) = f(\ell, e-1) \cdot 3^{2\ell} + 1$ for $e > 0$.

The sets X_i can be computed as shown in the proof of Lemma 7. Then the sets S_i can be obtained in the first paragraph of the proof of the present lemma. Their union results in C_0 which is C' for $e = 0$. Thus for $e = 0$, C' can be computed in time $O(|V(G)| - 2)^d)$ (instead of considering s and t, we may consider their sets of neighbors). Since the computation involves computing a minimum cut, we may assume that $d > 1$. Now assume that $e > 0$. For each i such that $1 \leq i \leq q+1$ and $|L_i| > 0$, we explore all possible disjoint subsets A and B of $S_i \cup S_{i-1}$. For the given choice, we check if the size of a minimum $a-b$ separator of $G_{i,A,B}$ is at most k (observe that it can be done in $O(|L_i|^d))$ and if yes, compute the set $C_{i,A,B}$. By the induction assumption, the computation takes $O(f(\ell, e-1) \cdot |L_i|^d)$. So, exploring all possible choices of A and B takes $O(f(\ell, e-1) \cdot 3^{2\ell} \cdot |L_i|^d)$. The overall complexity of computing C' is

$$O((|V(G)| - 2)^d + f(\ell, e-1) \cdot 3^{2\ell} \cdot \sum_{i=1}^{q+1} |L_i|^d).$$

Since all L_i are disjoint and $\bigcup_{i=1}^{q+1} L_i \subseteq V(G) \setminus \{s, t\}$, $\sum_{i=1}^{q+1} |L_i| \leq |V(G)| - 2$, hence $\sum_{i=1}^{q+1} (|L_i|)^d \leq (|V(G)|-2)^d$. Taking into account the recursive expression for $f(\ell, e)$, the desired runtime follows. □

Proof (of Theorem 10 continued). We construct the formula φ as

$$\varphi = \exists C(\text{AtMost}_k(C) \wedge \text{Separates}(C) \wedge \text{Induces}_{\mathcal{G}}(C)),$$

where $\text{AtMost}_k(C)$ is true if and only if $|C| \leq k$, $\text{Separates}(C)$ is true if and only if C separates the vertices of ST in G^*, $\text{Induces}_{\mathcal{G}}(C)$ is true if and only C induces a graph of $\mathcal{G}$.

In particular, $\text{AtMost}_k(C)$ states that C does not have $k+1$ mutually non-equal elements: this can be implemented as

$$\forall c_1, \ldots, \forall c_{k+1} \bigvee_{1 \leq i,j \leq k+1} (c_i = c_j).$$

Formula $\text{Separates}(C)$ is a slightly modified formula $\text{uvmc}(X)$ from [7] that looks as follows:

$$\forall s \forall t \big((ST(s) \wedge ST(t) \wedge \neg(s = t))\big) \\ \rightarrow \big(\neg C(s) \wedge \neg C(t) \wedge \forall Z(\text{Connects}(Z, s, t) \rightarrow \exists v(C(v) \wedge Z(v)))\big),$$

where $\text{Connects}(Z, s, t)$ is true if and only if in the modeling graph there is a path from s and t all vertices of which belong to Z. For definition of the predicate Connects, see Definition 3.1. in [7]

To construct $\text{Induces}_{\mathcal{G}}(C)$, we explore all possible graphs having at most k vertices and for each of these graphs we check whether it belongs to $\mathcal{G}$. Since the number of graphs being explored depends on k and $\mathcal{G}$ is a decidable class, in a FPT time we can compile the set $\{G'_1, \dots, G'_r\}$ of all graphs of at most k vertices that belong to $\mathcal{G}$. Let $k_1, \dots k_r$ be the respective numbers of vertices of $G'_1, \dots G'_r$. Then $\text{Induces}_{\mathcal{G}}(C) = \text{Induces}_1(C) \vee \dots \vee \text{Induces}_r(C)$, where $\text{Induces}_i(C)$ states that C induces G'_i. To define Induces_i, let $v_1, \dots v_{k_i}$ be the set of vertices of G'_i and define $\text{Adjacency}(c_1, \dots, c_{k_i})$ as the conjunction of all $E(c_x, c_y)$ such that v_x and v_y are adjacent in G'_i. Then

$$\begin{aligned}&\text{Induces}_i(C) = \text{AtMost}_{k_i}(C) \wedge \\ &\quad \exists c_1 \dots \exists c_{k_i} \Big(\bigwedge_{1 \le j \le k_i} C(c_j) \wedge \bigwedge_{1 \le x,y \le k_i} c_x \neq c_y \wedge \text{Adjacency}(c_1, \dots, c_{k_i}) \Big).\end{aligned}$$

Let us now verify that indeed $G_1 \models \varphi$ if and only if (G^*, s, t, k) is a 'YES' instance of the $\mathcal{G}$-MINCUT problem. Assume first the latter and let S be an $s-t$ separator of size at most k such that $G^*[S] \in \mathcal{G}$. Let us observe that all the three main conjuncts of φ quantified by C are satisfied when S is substituted instead C. That $\text{AtMost}_k(S)$ is true immediately follows from the pigeonhole principle: if we take $k+1$ elements out of a set of at most k elements, at least 2 of them must be equal. To show that $\text{Separates}(S)$ is true w.r.t. G_1, we draw the following line of implications. Set S separates s and t in G^*, hence the set of vertices of every path from s to t intersects with S, hence every set Z including as a subset a set of vertices of a path from s to t intersects with S. Formally written, the last statement can be expressed as follows $\forall Z(\text{Connects}(Z, s, t) \rightarrow \exists v(S(v) \wedge Z(v)))$, but this (together with the fact that S is disjoint with $\{s, t\}$) is the right part of the main implication of $\text{Separates}(S)$, hence $\text{Separates}(S)$ is true. To verify that $\text{Induces}_{\mathcal{G}}(S)$ is true w.r.t. G_1, let $G'_i \in \mathcal{G}$ be the graph isomorphic to $G^*[S]$ and observe that $\text{Induces}_i(S)$ is true by construction.

For the opposite direction assume that $G_1 \models \varphi$. It follows that there is a set of vertices C such that $\text{AtMost}_k(C)$, $\text{Separates}(C)$, and $\text{Induces}_{\mathcal{G}}(C)$ are all true. Consequently, $|C| \le k$. Indeed otherwise, we can select $k+1$ *distinct* elements of C that falsify at $\text{AtMost}_k(C)$. It also follows that C is disjoint with $\{s, t\}$ and separates s from t in G^*. Indeed s and t satisfy the left part of the main implication of $\text{Separates}(C)$, hence the right part of it must be satisfied as well. It immediately implies that C is disjoint with s and t. If we assume that C does not separate s and t then there is a P path from s to t avoiding C. Let $Z = V(P)$. Then $\text{Connects}(V(P), s, t)$ is true while $\exists v(C(v) \wedge Z(v)$ is false falsifying last conjunct of the right part of the main implication, a contradiction. Finally, it follows from $\text{Induces}_{\mathcal{G}}(C)$ that $\text{Induces}_i(C)$ is true for some i. By construction, this means that $G^*[C]$ is isomorphic to $G'_i \in \mathcal{G}$. Thus (G^*, s, t, k) is a 'YES' instance of the $\mathcal{G}$-MINCUT problem. □

Proof (of Corollary 11 continued). Assume that $(G, s, t, 2k)$ is a 'YES' instance of the $\mathcal{G}_k$-multicut problem. Let S be a $s-t$ separator such that $G[S] \in \mathcal{G}_k$. Let M be a maximum matching of $G[S]$. Then, by definition of $\mathcal{G}_k$, $|M| + (|V(G[S])| - 2|M|) \le k$ or, in other words, $(|V(G[S])| - 2|M|) \le k - |M|$. The $2|M|$ vertices of $G[S]$ (incident to the matching) are covered by $|M|$ edges. The remaining at most $k - |M|$ vertices can be covered by selecting an edge of G incident to each of them (that is possible due to our assumption about the absence of isolated vertices). Thus s and t may be separated by removal a set extendable to the union of at most k edges. Conversely, assume that s and t can be separated by removal of set S of vertices that can be extended to the union of at most k edges of G. Clearly $|S| \leq 2k$. It is not hard to observe that the size of the smallest set of edges covering S equals the size of the maximum matching $|M|$ of $G[S]$ plus $|V(G[S])| - 2|M|$ edges for the vertices not covered by the matching. By definition of S, $|M| + |V(G[S])| - 2|M| \leq k$. It follows that $G[S] \in \mathcal{G}_k$. Thus, $(G, s, t, 2k)$ is a 'YES' instance of the $\mathcal{G}_k$-multicut problem. □

Proof (of Theorem 17). It is more convenient to consider an annotated version of the problem where the independent set being deleted is a subset of a set $D \subseteq V(G)$ given as part of the input. Without the annotation, D is initially set to $V(G)$. The algorithm has the following 4 stopping conditions.

- If $k = 0$ and G is bipartite then return 'YES'.
- If $k = 0$, but G is not bipartite then return 'NO'.
- If $k > 0$, but G is bipartite then decide in a polynomial time whether $G[D]$ has an independent set of size exactly k.
- If $k > 0$ and $G \setminus D$ is not bipartite then return 'NO'.

Assume that no one of the above conditions is satisfied. Then the algorithm starts by finding an odd cycle C of minimum length (which is known to be doable in polynomial time, see for example Section 2 of `http://www.lancs.ac.uk/staff/letchfoa/articles/odd_circuit.pdf`). It is not difficult to see that the minimality of C implies that C is a triangle or C is chordless or every vertex not in C is adjacent to at most 2 vertices of the cycle.

Since no one of the stopping conditions holds, $|V(C) \cap D| > 0$. If $1 \leq |V(C) \cap D| \leq 3k + 1$, then we branch on selection of each vertex $v \in V(C) \cap D$ into the set S of vertices being removed and apply the algorithm recursively with the parameter k being decreased by 1 and the set D being updated by removal of v and $N(v) \cap D$. If $|V(C) \cap D| > 3k + 1$, then we apply the approach of Theorem 16 to find an independent set S of size at most k whose removal makes the graph bipartite. To ensure that $S \subseteq D$ we may, for example split all vertices $v \in V(G) \setminus D$ into $k + 1$ independent copies with the same neighborhood as v. If $|S| = k$, we are done. Otherwise, $|S| = k' < k$. In this case we observe that by construction each vertex of S (either in C or outside C) forbids the selection of at most 3 vertices of $V(C) \cap D$ including itself. Thus the number of vertices of $V(C) \cap D$ allowed for selection is at least $3k + 1 - 3k' = 3(k - k') + 1$. Since the cycle is chordless, we can select $k - k'$ independent vertices among them and thus complement S to being of size exactly k. Thus if the algorithm succeeds to find an independent set S of size at most k whose removal makes the graph bipartite, it may safely return 'YES'. It is clear that otherwise 'NO' is returned. □

B Proof of Theorem 18

We start with the introduction of new terminology. Given G, (H, C, K) as in the statement of the theorem and $L : V(G) \to 2^{V(H)}$ associating each vertex of G with the set of allowed vertices of H, we say that θ is a $(H, C, \leq K)$-coloring of (G, L) if θ is a $(H, C, \leq K)$-coloring of G such that for each $v \in V(G)$, $\theta(v) \in L(v)$. The *exceptional set* of θ is the set S of all vertices of G that are mapped to C by θ. Since $H \setminus C$ consists of two vertices without loops, $G \setminus S$ is bipartite. Moreover, the size of S is bounded by the parameter $k := \sum_{v \in C} K(v)$. Thus the considered problem is in fact a problem of constrained bipartization. However S, is not necessarily a minimal set whose removal makes the graph bipartite and hence we cannot straightforwardly use the approach of Theorem 16. Nevertheless, we *do* use the treewidth reduction approach based on the following definition.

Definition 19. *An $(H, C, \leq K)$-coloring θ of (G, L) is* minimal *if there is no $(H, C, \leq K)$-coloring θ' of (G, L) such that the exceptional set of θ' is a subset of the exceptional set of θ.*

Observe that if there is $(H, C, \leq K)$-coloring of (G, L), then there is a minimal $(H, C, \leq K)$-coloring as well. We prove that there is an FPT-computable graph G^* that preserves exceptional sets of all minimal $(H, C, \leq K)$-colorings of (G, L) and whose treewidth is bounded by a function of k (recall that $k = \sum_{v \in C} K(v)$). Similarly to the cases of $\mathcal{G}$-MINCUT and $\mathcal{G}$-BIPARTIZATION, we use this result to transform the given instance of the $(H, C, \leq K)$-coloring problem to an instance with bounded treewidth and then apply Courcelle's Theorem.

We need some technical results. First, we restate Lemma 8 in terms of separating two sets X and Y (instead of $s - t$ separators).

Lemma 20. *Let X, Y be two sets of vertices of graph G. For some $k \geq 0$, let C be the union of all minimal sets S of size at most k separating X and Y. Then for some constant d there is an $O(f(k) \cdot |V(G)|^d)$ time algorithm that returns a set $C' \supseteq C$ such that the treewidth of* $\text{torso}(G, C')$ *is at most $g(k)$, for some functions f and g depending only on k.*

Proof. Let G' be the graph obtained from G by introducing two new vertices s, t and connecting s (resp., t) to every vertex of X (resp., Y). It is clear that a set $S \subseteq V(G)$ is an $s - t$ separator in G' if and only if S separates X and Y in G. Let us use the algorithm of Lemma 8 to obtain a set C' (containing s, t) that fully contains all the minimal $s - t$ separators. It follows that $C' \setminus \{s, t\}$ fully contains all the minimal sets that separate X and Y in G. Furthermore, we observe that the treewidth of $\text{torso}(G, C' \setminus \{s, t\})$ is not larger that the treewidth of $\text{torso}(G', C')$. In fact, the former graph is a subgraph of the latter: if two vertices $a, b \in C' \setminus \{s, t\}$ are adjacent in $\text{torso}(G, C' \setminus \{s, t\})$, then they are adjacent in $\text{torso}(G', C')$ as well. □

The following lemma will be used for the inductive proof of the treewidth reduction result.

Lemma 21. *Let $C' \subseteq V(G)$ such that* $\text{torso}(G, C')$ *has treewidth at most w_1. Let $R_1, \ldots, R_r$ be components of $G \setminus C'$, and for every $1 \leq i \leq r$, let $C'_i \subseteq R_i$ be such that* $\text{torso}(G[R_i], C'_i)$ *has treewidth at most w_2. Then* $\text{torso}(G, C'')$ *has treewidth at most $w_1 + w_2 + 1$ for $C'' := C' \cup \bigcup_{i=1}^{r} C'_i$.*

Proof. Let T be a tree decomposition of $\text{torso}(G, C')$ with width at most w_1, and let T_i be a tree decomposition of $\text{torso}(G[R_i], C'_i)$ with width at most w_2. Let $N_i \subseteq C'$ be the neighborhood of R_i in G. Since N_i induces a clique in $\text{torso}(G, C')$, we have $|N_i| \leq w_1 + 1$ and there is a bag B_i of T containing N_i. Let us modify T_i by including N_i in every bag and then let us join T and T_i by connecting an arbitrary bag of T_i with B_i. By performing this step for every $1 \leq i \leq r$, we get a tree decomposition with width at most $w_1 + w_2 + 1$. To show that it is indeed a tree decomposition of $\text{torso}(G, C'')$, it is sufficient to observe that the set of edges of $\text{torso}(G, C')$ is a superset of the set of edges of the graph induced by C' in $\text{torso}(G, C'')$, and that $\text{torso}(G[R_i], C'_i)$ is exactly the same as the graph induced by C'_i in $\text{torso}(G, C'')$. ◻

Now we are ready to formulate the treewidth reduction result.

Lemma 22. *Assume that G is bipartite. Then there is an* FPT *algorithm parameterized by $k = \sum_{v \in V(C)} K(v)$ that finds a set C'' such that the treewidth of* $\text{torso}(G, C'')$ *is at most $f(k, |V(H)|)$ for some function f and the exceptional set of every minimal $(H, C, \leq K)$-coloring of (G, L) is a subset of C''.*

Proof. The proof is by induction on k. For $k = 0$, we can set $C'' = \emptyset$, hence $\text{torso}(G, C'')$ is the empty graph whose treewidth is 0. Assume now that $k > 0$. Denote the vertices of $H \setminus C$ by b and w. Let B be the set of all vertices $v \in V(G)$ such that $w \notin L(v)$. Analogously, let W be the set of all vertices $v \in V(G)$ such that $w \notin L(v)$. Let (B', W') be a 2-coloring of G and set $X := (B \cap B') \cup (W \cap W')$ and $Y := (B \cap W') \cup (W \cap B')$ as in Lemma 15. If X and Y are not connected then, by Lemma 15, there is a 2-coloring of G where B and W are colored in black and white respectively. In other words, there is a $(H, C, \leq K)$-coloring of (G, L) where each vertex of G is mapped to b and w. Consequently, all minimal $(H, C, \leq K)$-colorings of (G, L) have exceptional sets of size 0 and hence $C'' = \emptyset$ as in the case with $k = 0$.

If X and Y are connected, then let us use Lemma 20 to compute in FPT time a set C' such that every minimal set separating X and Y in G is a subset of C' and $\text{torso}(G, C')$ is bounded by a function of k.

Let P be a connected component of $G \setminus C'$ and let N be the subset of C' that consists of all vertices adjacent to the vertices of P. Let θ be an $(H, C, \leq K)$-coloring of $(G[N], L[N])$ where $L[N]$ is the restriction of L to the vertices of N. Let L_θ be the function on $V(P)$ obtained from $L[V(P)]$ by the following operation: for each $v \in V(P)$, remove $u \in L(v)$ from the list of v whenever there is a neighbor x of v in G such that $x \in N$ and $\theta(x)$ is not adjacent to u in H. In other words, L_θ updates the list of allowed colors of $V(P)$ so that they are compatible with the mapping of θ on N. Furthermore, let K' be a function associating the vertices of H with integers so that $\sum_{v \in C} K'(v) \leq k - 1$. By the induction assumption there is an FPT algorithm parameterized by $k-1$ that returns a set $C_{P,\theta,K'} \subseteq V(P)$ such that torso$(P, C_{P,\theta,K'})$ has the treewidth bounded by a $f(k-1, |V(H)|)$ and the exceptional set of any minimal $(H, C \leq K')$-coloring of (P, L_θ) is a subset of $C_{P,\theta,K'}$. Let C_P be the union of all possible sets $C_{P,\theta,K'}$. Observe that the number of possible mappings θ is bounded by a function of k and $|V(H)|$: the vertices of N create a clique in torso(G, C') hence $|N|$ is bounded by a function of k. As well, the number of possible mappings K' is bounded by a function of $k-1$ and $|V(H)|$. Therefore by Lemma 6, the treewidth of torso(P, C_P) is bounded by a function of k and $|V(H)|$. Let C'' be the union of C' and the sets C_P for all the connected components P of $G \setminus C'$. According to Lemma 21, the treewidth of torso(G, C'') is bounded by $f(k, |V(H)|)$ for an appropriately selected function f. (Such function can be defined similarly to function g in the proof of Lemma 8). Also, arguing similarly to Lemma 8, we can observe that C'' can be computed in an FPT time parameterized by k.

It remains to be shown that the exceptional set S of every minimal $(H, C, \leq K)$-coloring θ of (G, L) is a subset of C''. Since in $G \setminus S$ vertices of $B \setminus S$ are colored in black (i.e., mapped to b by θ) and the vertices of $W \setminus S$ are colored in white (i.e., mapped to w by θ), S separates X and Y according to Lemma 15. Therefore, S contains at least one element of C'. Consequently, for any connected component P of $G \setminus C'$, $|S \cap V(P)| \leq k-1$. Let θ_P be the restriction of θ to the vertices of P and for each vertex v of C define $K'(v)$ as the number of vertices of P mapped to v by θ_P. Let θ' be the restriction of θ to the vertices of C' adjacent to $V(P)$. It is not hard to observe that θ_P is a minimal $(H, C, \leq K')$-coloring of $(P, L_{\theta'})$. In other words, $S \cap V(P) \subseteq C_{P,\theta',K'} \subseteq C_P$. Since each vertex v belongs either to C' or to some $V(P)$, the present lemma follows. □

Lemma 23. *For every fixed H, $(H, C, \leq K)$-coloring can be solved in* FPT *time parameterized by k, $|V(H)|$, and w, where w is the treewidth of G.*

Proof. The problem can be solved by a straightforward application of Courcelle's Theorem; we only sketch the proof. Let (G, L) be an instance of the $(H, C, \leq K)$-coloring. For each $x \in V(H)$, let L_x be the subset of $V(G)$ consisting of all vertices v such that $x \in L(v)$. Denote the vertices of H by $x_1, \dots, x_r$ and let $G_1 = (V(G), E(G), L_{x_1}, \dots, L_{x_r})$ be a labeled graph. We construct a formula φ such that $G_1 \models \varphi$ if and only if there is a $(H, C, \leq K)$-coloring of (G, L).

The formula φ is defined as

$$\exists V_1 \exists V_2 \dots \exists V_r \Big(\bigwedge_{\substack{1 \leq i \leq r \\ x_i \in C}} \text{AtMost}_{K(c)}(V_c) \wedge \text{partition}(V_1, \dots, V_r) \wedge \bigwedge_{\substack{x_i, x_j \in V(H) \\ x_i x_j \notin E(H)}} \forall v, u((V_i(v) \wedge V_i(u)) \rightarrow \neg E(v, u)) \Big),$$

where

$$\text{partition}(V_1, \dots, V_r) := \Big(\bigwedge_{1 \leq i < j \leq r} \text{disjoint}(V_i, V_i) \Big) \wedge \Big(\forall v \bigvee_{1 \leq i \leq r} V_i(v) \Big)$$

expresses that $(V_1, \ldots, V_r)$ is a partition of $V(G)$. It is not hard to see that $G_1 \models \varphi$ if and only if for some choice of $V_1, \ldots, V_r$ there is an $(H, C, \leq K)$-coloring θ of (G, L) defined by $\theta(v) = x_i$ if and only if $v \in V_i$. Note furthermore that the length of φ depends only on k and $|V(H)|$. □

Proof (of Theorem 18). First we show that it can be assumed that G is bipartite. Otherwise, we use the FPT algorithm of [15] to find a set S' of at most k vertices whose deletion make G bipartite. We branch on the $|V(H)|^{|S'|}$ possible ways of defining θ on S'. For each of these ways we appropriately update the values of $K(v)$ for all $v \in C$. Also, if a vertex $v \in V(G) \setminus S'$ has a neighbor $u \in S'$, then we modify the list of v such that it contains only vertices adjacent to $\theta(u)$ in H. It is clear that the original instance has a solution if and only if at least one of the resulting instances has a solution.

We may also assume that for each $v \in C$, $K(v) > 0$. Otherwise, simply remove from H the vertices with $K(v) = 0$.

As G is bipartite, we can use Lemma 22 to obtain the set C''. We transform G the following way. Let P_i be a connected component of $G \setminus C''$ having more than one vertex. Let (X_i, Y_i) be the bipartition of P_i (unique due to the connectedness of P_i). We replace P_i by two adjacent vertices x_i and y_i such that x_i (resp., y_i) is adjacent with the neighborhood of X_i (resp., Y_i) in C''. We define $L(x_i) = \{b, w\} \cap \bigcap_{v \in X_i} L(v)$, and $L(y_i)$ is defined analogously. Let G' be the graph obtained after performing this operation for every component P, and let L' be the resulting list assignment on G'.

We claim that (G, L) has a $(H, C, \leq K)$-coloring if and only if (G', L') has a $(H, C, \leq K)$-coloring. Suppose first that θ is a minimal $(H, C, \leq K)$-coloring of (G, L). We know that the exceptional set of θ is contained in C'', thus vertices of a component P of $G \setminus C''$ are mapped to b and w. Thus if (X_i, Y_i) is the bipartition of P_i, then due to P_i being a connected graph, every vertex of X_i is mapped to the same vertex of $H \setminus C$ and similarly for Y_i. Thus by mapping x_i and y_i to these vertices in G', we can obtain a $(H, C, \leq K)$-coloring of (G', L').

Suppose now that (G', L') has a $(H, C, \leq K)$-coloring θ'. Again, let P_i be a component of $G \setminus C''$ with bipartition (X_i, Y_i). We can obtain a $(H, C, \leq K)$-coloring θ of (G, L) by mapping every vertex of X to $\theta'(x_i) \in \{b, w\}$ and every vertex of Y_i to $\theta'(y_i) \in \{b, w\}$.

We show that the treewidth of G' is bounded by a function of k. Indeed, according to Lemma 22, the treewidth of $\mathrm{torso}(G, C'')$ is bounded by a function of k and $|V(H)|$. But since we assumed that for each $v \in C$, $K(v) > 0$, $|V(H)| \leq k + 2$. It follows that the treewidth of $\mathrm{torso}(G, C'')$ is bounded by a function of k only. Furthermore the treewidth of G' is greater than the treewidth of $\mathrm{torso}(G, C')$ by at most 2. Indeed, $G'[C'']$ is a subgraph $\mathrm{torso}(G, C'')$: they have the same set of vertices but the set of edges of the former may be a subset of the set of edges of the latter. Therefore, any tree decomposition of $\mathrm{torso}(G, C'')$ is a tree decomposition of $G'[C'']$. Such a tree decomposition has the property that that for each component P of $G \setminus C'$, there is a bag B including the set N of neighbors of this component in C'. We create a new bag containing N and the two vertices corresponding to P in G' and connect this new bag to B. It is not hard to observe that doing so for each connected component of $G \setminus C'$ we obtain a tree decomposition of G' of width greater than the initial tree decomposition of $\mathrm{torso}(G', C'')$ by at most 2. This completes the proof of the bound on the treewidth of G'.

Thus, the argumentation above implies that under assumption that $H \setminus C$ is a graph of two adjacent vertices without loops, the instance (G, L) of the $(H, C, {\leq}K)$-coloring problem can be transformed in FPT time parameterized by k into an instance (G', L') of the $(H, C, {\leq}K)$-coloring problem and the treewidth of G' bounded by a function of k. Taking again into account that $|V(H)| \leq k+2$, the present theorem now follows from Lemma 23. $\square$