Submitted to the Symposium on Theoretical Aspects of Computer Science
www.stacs-conf.org

TREEWIDTH REDUCTION FOR CONSTRAINED SEPARATION AND BIPARTIZATION PROBLEMS

DÁNIEL MARX [1] AND BARRY O'SULLIVAN [2] AND IGOR RAZGON [3]

Tel Aviv University
E-mail address: dmarx@cs.bme.hu

Cork Constraint Computation Centre, University College Cork
E-mail address: b.osullivan@cs.ucc.ie

Cork Constraint Computation Centre, University College Cork
E-mail address: 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 for establishing fixed-parameter tractability of constrained separation and bipartization problems. To demonstrate the power of this technique, 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

Finding cuts and separators is a classical topic of combinatorial optimization and in recent years there have been an increase of interest in the fixed-parameter tractability of such problems [19, 11, 15, 28, 16, 13, 5, 20]. Recall that a problem is *fixed-parameter tractable* (or FPT) with parameter k if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function $f(k)$ depending only on k [10, 12, 21]. In typical parameterized separation problems, the parameter k is the size of the separator we are looking for, thus fixed-parameter tractability with respect to this parameter means that the combinatorial explosion is restricted to the size of the separator, but otherwise the running time depends polynomially on the size of the graph.

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. Combining this theorem with the well-known Courcelle's Theorem, we obtain a powerful tool for proving the fixed parameter tractability of constrained separation and bipartization problems. We demonstrate the power of the methodology with the following results.

- We prove that the MINIMUM STABLE $s-t$ CUT problem (Is there an independent set S of size at most k whose removal separates s and t?) is fixed-parameter tractable. This problem received some attention in the community. Our techniques allow to prove various generalizations of this result very easily. First, instead of requiring that S is independent, we can require that it induces a graph that belongs to a hereditary class $\mathcal{G}$; the problem remains FPT. Second, in the MULTICUT problem a list of pairs of terminals are given $(s_1, t_1), \ldots,$ (s_ℓ, t_ℓ) and S is a set of at most k vertices that induces a graph from $\mathcal{G}$ and separates s_i from t_i for every i. We show that this problem is FPT parameterized by k and ℓ, which is a very strong generalization of previous results [19, 28]. Third, the results generalize to the MULTICUT-UNCUT problem, where two sets T_1, T_2 of pairs of terminals are given, and S has to separate every pair of T_1 and *should not* separate any pair of T_2.

- We prove that the EXACT STABLE BIPARTIZATION problem (Is there an independent set of size *exactly* k whose removal makes the graph bipartite?) is fixed-parameter tractable (FPT) answering an open question posed in 2001 by Díaz et al. [9]. We establish this result through proving that the STABLE BIPARTIZATION problem (Is there an independent set of size *at most* k whose removal makes the graph bipartite?) is FPT, answering an open question posed by Fernau [7].
- As a demonstration, we show that the EDGE-INDUCED VERTEX CUT (Are there at most k edges such that removal of their endpoints separates two given terminals s and t?) is FPT, answering an open problem posed in 2007 by Samer [7]. The motivation behind this problem is described in [27]. While the reader might not be particularly interested in this exotic variant of $s-t$ cut, we believe that it nicely demonstrates the message of the paper. Slightly changing the definition of a well-understood cut problem usually makes the problem NP-hard and determining the parameterized complexity of such variants directly is by no means obvious. On the other hand, using our techniques, the fixed-parameter tractability of many such problems can be shown with very little effort. Let us mention (without proofs) three more variants that can be treated in a similar way: (1) separate s and t by the deletion of at most k edges and at most k vertices, (2) in a 2-colored graph, separate s and t by the deletion of at most k black and at most k white vertices, (3) in a k-colored graph, separate s and t by the deletion of one vertex from each color class.

Thus our method leads to the solution of several independent problems; it seems that the same combinatorial difficulty lies at the heart of these problems. Our technique manages to overcome this difficulty and it is expected to be of use for further problems of similar flavor. Note that while designing FPT-time algorithms for bounded-treewidth graphs and in particular the use of Courcelle's Theorem is a fairly standard technique, we use this technique for problems where there is no bound on the treewidth in the input.

(Multiterminal) cut problems [19, 16, 13, 5] play a mysterious, not yet fully understood role in the fixed-parameter tractability of certain problems. Proving that BIPARTIZATION [25], DIRECTED FEEDBACK VERTEX SET [6], and ALMOST 2-SAT [23] are FPT answered longstanding open questions, and in each case the algorithm relies on a nonobvious use of separators. Furthermore, EDGE MULTICUT has been observed to be equivalent to FUZZY CLUSTER EDITING, a correlation clustering problem [3, 8, 1]. Thus aiming for a better understanding of separators in a parameterized setting seems to be a fruitful direction of research. Our results extend our understanding of separators by showing that various additional constraints can be accommodated. It is important to point out that our algorithm is very different from previous parameterized algorithm for separation problems [19, 16, 13, 5]. Those algorithms in the literature exploited certain nice properties of separators, and hence it seems impossible to generalize them for the problems we consider here. On the other hand, our approach is very robust and, as demonstrated by our examples, it is able to handle many variants.

The paper assumes the knowledge of the definition of treewidth and its algorithmic use, including Courcelle's Theorem (see the surveys [2, 14]).

2. Treewidth reduction

The main combinatorial result of the paper is presented in this section. We start with some preliminary definitions. Two slightly different notion of separation will be used in the paper:

Definition 2.1. We say that a set S of vertices *separates* sets of vertices 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 2.2 ([24], [12, 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 2.2 says that if every W has a balanced k-separator, then the treewidth is at most $3k$. This observation, and the following simple extension, will be convenient tools for showing that a certain graph has low treewidth.

Lemma 2.3. *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 $W_i := (W \cap C_i) \setminus (\bigcup_{j=1}^{i-1} C_j)$; 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. ∎

If we are interested in separators of graph G fully contained in a subset C of vertices, then each component of $G \setminus C$ (or the neighborhood of each component in C) can be replaced by a clique, since there is no way to disconnect these components with separators in C. The notion of torso and Proposition 2.5 formalize this concept.

Definition 2.4. Let G be a graph and $C \subseteq V(G)$. The graph $\text{torso}(G, C)$ has vertex set C and 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 2.5. *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 $\text{torso}(G, C_1)$ if and only if S separates these vertices in $\text{torso}(G, C_2)$. In particular, by setting $C_2 = V(G)$, we get that $S \subseteq C_1$ separates a and b in $\text{torso}(G, C_1)$ if and only if it separates them in G.*

Proof. 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 $\text{torso}(\text{torso}(G, C_2), C_1) = \text{torso}(G, C_1)$. ■

Analogously to Lemma 2.3, we can show that if we have a treewidth bound on $\text{torso}(G, C_i)$ for every i, then these bounds add up for the union of the C_i's.

Lemma 2.6. *Let G be a graph and $C_1, \dots, C_r$ be subsets of $V(G)$ such that for every $1 \le i \le r$, the treewidth of* $\text{torso}(G, C_i)$ *is at most w. Then the treewidth of* $\text{torso}(G, C)$ *for $C := \bigcup_{i=1}^{r} C_i$ is at most $3r(w + 1)$.*

Proof. 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(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 2.5, 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 2.3 hold, and W has a balanced separator $S \subseteq C$ of size at most $r(w+1)$ in G. Again by Proposition 2.5, the set S is a balanced separator of W in $\text{torso}(G, C)$ as well. By Lemma 2.2(1), it follows that $\text{torso}(G, C)$ has treewidth 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 ∞. The aim of this section is to show that, for every k, we can construct a set C' covering all the $s - t$ separators of size at most k such that $\text{torso}(G, C')$ has treewidth bounded by a function of k. Equivalently, we can require that C' covers every $s - t$ separator of excess at most $e := k - \ell$, where ℓ is the minimum size of an $s - t$ separator.

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. The following result is folklore; it can be proved by a simple application of the uncrossing technique (see the Appendix) and it can be deduced also from the observations of [22] on the strongly connected components of the residual graph after solving a flow problem.

Lemma 2.7. *Let s, t be two vertices in graph G such that the minimum size of an $s - t$ separator is ℓ. Then there is a collection $\mathcal{X} = \{X_1, \dots X_q\}$ 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) $|\delta(X_i)| = \ell$ for every $1 \le i \le q$, and
(3) every $s - t$ separator of size ℓ is fully contained in $\bigcup_{i=1}^{q} \delta(X_i)$.

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

Lemma 2.7 shows that the union C of all minimum $s - t$ separators can be covered by a chain of minimum $s - t$ separators. It is not difficult to see that this chain can be used to define a tree decomposition (in fact, a path decomposition) of $\text{torso}(G, C)$. This observation solves the problem for $e = 0$. For the general case, we use induction on e.

Lemma 2.8. *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 $k = \ell + e$). 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* $\text{torso}(G, C')$ *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 2.7 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 2.9. If a vertex $v \in L_i$ is in C, then there are 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 $G_{i,A,B}$ of size at most k (recall that $k = \ell + e$) and excess at most $e - 1$.

Proof. By definition of C, 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 $G \setminus K$. Let us 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 $a - b$ 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 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 from s to t 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. This completes the proof of Claim 2.9. ■

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 an $s-t$ 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 2.5, it will imply that in torso(G, C'), W has a balanced separator with the same upper bound. By Lemma 2.2(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(2) and Proposition 2.5, graph 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 2.3. By definition of C', $W \cap L_i \subseteq C^*$, hence the existence of the desired separator S follows.

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 2.7. 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. ■

Remark 2.10. The recursion $g(\ell, e) := 3 \cdot (2\ell + 3^{2\ell} \cdot g(\ell, e-1))$ implies that $g(\ell, e)$ is $2^{O(e\ell)}$, i.e., the treewidth bound is exponential in ℓ and e. It is an obvious question whether it is possible to improve this dependence to polynomial. However, a simple example (graph G is the n-dimensional hypercube, $k = (n-1)n$, s and t are opposite vertices) shows that the function $g(\ell, e)$ has to be exponential. The size of the minimum $s-t$ separator is $\ell := n$. We claim that every vertex v of the hypercube (other than s and t) is part of a minimal $s-t$ separator of size at most $n(n-1)$. To see this, let P be a shortest path connecting s and v. Let $P' = P - v$ be the subpath of P connecting s with a neighbor v' of v. Let S be the neighborhood of P'; clearly S is an $s-t$ separator and $v \in S$. However, $S \setminus v$ is not an $s-t$ separator: the path P is not blocked by $S \setminus v$ as $S \setminus v$ does not contain any vertex farther from s than v. Since P' has at most $n-1$ vertices and every vertex has degree n, we have $|S| \leq n(n-1)$. Thus v (and every other vertex) is part of a minimal separator of size at most $n(n-1)$. Hence if we set $\ell := n$ and $e := n(n-1)$, then C contains every vertex of the hypercube. The treewidth of an n-dimensional hypercube is $\Omega(2^n/\sqrt{n})$ [4], which is also a lower bound on $g(\ell, e)$.

The following theorem states our main combinatorial tool in a form that will be very convenient to use:

Theorem 2.11. [The treewidth reduction theorem] *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 of size at most k 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 $s, t \in S$ that can be separated by the removal of at most k vertices, the algorithm of Lemma 2.8 computes a set $C'_{s,t}$ containing all the minimal $s-t$ separators of size at most k. By Lemma 2.6, if C' is the union of these at most $\binom{|S|}{2}$ sets, then $G' = \text{torso}(G, C')$ 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 2.4). 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 $w_1, \ldots, w_{k+1}$ and replace $\{u, v\}$ by the set of edges $\{\{u, w_1\} \ldots \{u, w_{k+1}\}, \{w_1, v\}, \ldots, \{w_{k+1}, v\}\}$. Let G^* be the resulting graph. It is not hard to check that G^* satisfies all the properties of the present theorem. ∎

Remark 2.12. The treewidth of G^* may be larger than the treewidth of G. We use the phrase 'treewidth reduction' in the sense that the treewidth of G^* is bounded by a function of k and $|S|$, while the treewidth of G is unbounded.

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 if G has a $s-t$ separator C of size at most k such that $G[C] \in \mathcal{G}$. The following theorem is the central result of this section.

Theorem 3.1. *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 2.11 for $S = \{s,t\}$. According to Theorem 2.11, 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 $|K| \leq k$ 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 2.11), K separates s from t in G^* and $G^*[K] \in \mathcal{G}$. The opposite direction can be proved similarly.

Thus we have established an 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 [14], Remarks 3.19[1] and 3.20), it will follow that the $\mathcal{G}$-MINCUT problem is FPT. The part of φ describing the separation of s and t is based on the ideas from [13].

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 [13] that looks as follows:

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

[1]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 [26]

where 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 [13]

To construct 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 Induces$_{\mathcal{G}}(C) =$ Induces$_1(C) \vee \dots \vee$ Induces$_r(C)$, where 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 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

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

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 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 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-hand part of the main implication of Separates(S), hence Separates(S) is true. To verify that 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 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 AtMost$_k(C)$, Separates(C), and Induces$_{\mathcal{G}}(C)$ are all true. Consequently, $|C| \leq k$. Indeed otherwise, we can select $k + 1$ *distinct* elements of C that falsify at 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 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 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 Induces$_{\mathcal{G}}(C)$ that 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. ∎

In particular, let $\mathcal{G}^0$ be the class of all graphs without edges. Then $\mathcal{G}^0$-MINCUT is the MINIMUM STABLE $s - t$ CUT problem whose fixed-parameter tractability has been posed as an open question by Kanj [17]. Clearly, $\mathcal{G}^0$ is hereditary and hence the $\mathcal{G}^0$-MINCUT is FPT.

Theorem 3.1 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 show (with a very easy argument) that some of these problems actually become hard if the size is required to be exactly k. Let graph G' be obtained from graph G by introducing two isolated vertices s and t. Now there is an independent set of size exactly k that is an $s - t$ separator in G' if and only if there is an independent set of size k in G, implying that finding such a separator is W[1]-hard.

Theorem 3.2. *It is* W[1]*-hard to decide if G has an $s-t$ separator that is an independent set of size exactly k.*

Samer and Szeider [27] 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 [27] whether this problem is FPT. Samer reposted this problem as an open question in [7]. Using Theorem 3.1, we answer this question positively. For this purpose, we introduce $\mathcal{G}_k$, the class of graphs where the number of vertices minus the size of the maximum matching is at most k, observe that this class is hereditary, and show that (G, s, t, k) is a 'YES'-instance of the *edge-induced vertex-cut* problem if and only if $(G, s, t, 2k)$ is a 'YES' instance of the $\mathcal{G}_k$-mincut problem. Then we apply Theorem 3.1 to get the following corollary (the proof is in the Appendix).

Corollary 3.3. *The* EDGE-INDUCED VERTEX-CUT *problem is* FPT.

MULTICUT is the generalization of MINCUT where, instead of s and t, the input contains a set $(s_1, t_1), \dots, (s_\ell, t_\ell)$ 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 [19, 28] 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 3.1 for $\mathcal{G}$-MULTICUT: all we need to do is to change the construction of φ such that it requires the separation of each pair (s_i, t_i). We state this here in an even more general form. In the $\mathcal{G}$-MULTICUT-UNCUT problem the input contains an additional integer $\ell' \le \ell$, 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 3.4. *If $\mathcal{G}$ is* decidable *and* hereditary, *then $\mathcal{G}$-*MULTICUT-UNCUT *is* FPT *parameterized by k and ℓ.*

Theorem 3.4 helps clarifying a theoretical issue. In Section 2, we defined C as the set of all vertices appearing in minimal $s-t$ separators of size at most k. There is no obvious way of finding this set in FPT-time and Lemma 2.6 produces only a superset C' of C. However, Theorem 3.4 can be used to find C: a vertex v is in C if and only if there is a set S of size at most $k-1$ and two neighbors v_1, v_2 of v such that S separates s and t in $G \setminus v$, but S does not separate s from v_1 and t from v_2 in $G \setminus v$ (including the possibility that $v_1 = s$ or $v_2 = t$).

4. Constrained Bipartization Problems

Reed et al. [25] 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 (see also [18] for a somewhat simpler presentation of the algorithm). 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 4.1. *Let G be a 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 $Y := (B \cap W') \cup (W \cap B')$.*

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

Proof. Using the algorithm of [25], 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 $R = \{v_1, \dots, v_r\}$ 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 4.1, 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 4.1, 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 3.1. ∎

Theorem 4.2 immediately implies that the STABLE BIPARTIZATION problem is FPT: just set $\mathcal{G}$ to be the class of all graphs without edges. This answers an open question of Fernau [7]. Next, we show that the EXACT STABLE BIPARTIZATION problem is FPT, answering a question posed by Díaz et al. [9]. This result may seem surprising because the corresponding exact separation problem is W[1]-hard by Theorem 3.2 and hence the approach of Theorem 4.2 is unlikely to work. Instead, we argue that under appropriate conditions, any solution of size at most k can be extended to an independent set of size exactly k.

Theorem 4.3. *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 has to be 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 can be done in polynomial time). It is not difficult to see that the minimality of C implies that either C is a triangle or C is chordless. Moreover, in the latter case, 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 $1 \le |V(C) \cap D| \le 3k + 1$, 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 4.2 to find an independent set $S \subseteq D$ of size at most k whose removal makes the graph bipartite, and then argue that S can be extended to an independent set of size exactly k. 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 the minimality of C, 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.

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 check if $G[D]$ has k independent vertices, which can be done in a polynomial time. ■

References

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[2] H. L. Bodlaender. Treewidth: Characterizations, applications, and computations. In *WG*, pages 1–14, 2006.

[3] H. L. Bodlaender, M. R. Fellows, P. Heggernes, F. Mancini, C. Papadopoulos, and F. A. Rosamond. Clustering with partial information. In *MFCS*, pages 144–155, 2008.

[4] L. S. Chandran and T. Kavitha. The treewidth and pathwidth of hypercubes. *Discrete Math.*, 306(3):359–365, 2006.

[5] J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. In *WADS*, pages 495–506, 2007.

[6] 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.

[7] 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.

[8] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2-3):172–187, 2006.

[9] J. Díaz, M. Serna, and D. M. Thilikos. (H, C, K)-coloring: fast, easy, and hard cases. In *MFCS 2001*, pages 304–315, 2001.

[10] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Monographs in Computer Science. Springer, New York, 1999.

[11] U. Feige and M. Mahdian. Finding small balanced separators. In *STOC 2006*, pages 375–384. ACM, 2006.

[12] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[13] G. Gottlob and S. T. Lee. A logical approach to multicut problems. *Inform. Process. Lett.*, 103(4):136–141, 2007.

[14] M. Grohe. Logic, graphs, and algorithms. In J. Flum, E. Grädel, and T. Wilke, editors, *Logic and Automata- History and Perspectives*. Amsterdam University Press, 2007.

[15] S. Guillemot. FPT algorithms for path-transversals and cycle-transversals problems in graphs. In *IWPEC*, pages 129–140, 2008.

[16] 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.

[17] I. Kanj. Open problem session of Dagstuhl seminar 08431, 2008.

[18] D. Lokshtanov, S. Saurabh, and S. Sikdar. Simpler parameterized algorithm for OCT. To appear in IWOCA'09.

[19] D. Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394–406, 2006.

[20] D. Marx and I. Razgon. Constant ratio fixed-parameter approximation of the edge multicut problem, 2009. Accepted for ESA 2009.

[21] R. Niedermeier. *Invitation to fixed-parameter algorithms*, volume 31 of *Oxford Lecture Series in Mathematics and its Applications*. Oxford University Press, Oxford, 2006.

[22] J.-C. Picard and M. Queyranne. On the structure of all minimum cuts in a network and applications. *Math. Programming Stud.*, (13):8–16, 1980. Combinatorial optimization, II (Proc. Conf., Univ. East Anglia, Norwich, 1979).

[23] I. Razgon and B. O'Sullivan. Almost 2-SAT is fixed-parameter tractable. In *ICALP (1)*, pages 551–562, 2008.

[24] 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.

[25] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

[26] N. Robertson and P. Seymour. Graph minors. X. obstructions to tree-decomposition. *J. Comb. Theory, Ser. B*, 52(2):153–190, 1991.

[27] M. Samer and S. Szeider. Complexity and applications of edge-induced vertex-cuts. *CoRR*, abs/cs/0607109, 2006.

[28] M. Xiao. Algorithms for multiterminal cuts. In *CSR*, pages 314–325, 2008.

Appendix A. Proofs Section 2

Proof of Lemma 2.7. 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.,

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

Both $\delta(X_i \cap X_j)$ and $\delta(X_i \cup X_j)$ are $s-t$ separators (because both $X_i \cap X_j$ and $X_i \cup X_j$ contain s) and hence have size at least k. The left hand side is 2ℓ, hence there is equality and $|\delta(X_i \cap X_j)| = |\delta(X_i \cup X_j)| = \ell$ 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

$$|\delta(X_i \cap X_j) \cup \delta(X_i \cup X_j)| = 2\ell - |\delta(X_i \cap X_j) \cap \delta(X_i \cup X_j)| \geq 2\ell - |\delta(X_i) \cap \delta(X_j)| = |\delta(X_i) \cup \delta(X_j)|,$$

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.

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. ∎

Appendix B. Proofs Section 3.1

Proof of Corollary 3.3. 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 3.1 that $\mathcal{G}_k$-MINCUT is FPT.

We will 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.

Assume that $(G, s, t, 2k)$ is a 'YES' instance of the $\mathcal{G}_k$-MINCUT problem and let S be a $s-t$ separator of size at most $2k$ such that $G[S] \in \mathcal{G}_k$. Since $\mathcal{G}_k$ is hereditary, we may assume that S is minimal. Let M be a maximum matching of $G[S]$. Then, by definition of $\mathcal{G}_k$, $|M| + (|V(G[S])| - 2|M|) \leq k$ or, in other words, $(|V(G[S])| - 2|M|) \leq k - |M|$. 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 (due to the minimality of S, it does not contain isolated vertices). Thus s and t may be separated by removal a set extendible 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 3.2. Let G be a graph and let G' be a graph obtained from G by adding two isolated vertices s and t. Clearly, G has an independent set of size exactly k if and only if G' has an independent set of size exactly k *separating* s and t. Since it is W[1]-hard to check existence of an independent set of size exactly k, it follows that it is also W[1]-hard to check existence of an independent set of size exactly k separating s and t.

The hardness to check existence of a separator of size exactly k that is a clique or a dominating set can be proven similarly. ∎

Proof of Theorem 3.4. It is convenient to represent the input of the $\mathcal{G}$-MULTICUT-UNCUT problem in the form (G, T_1, T_2, k), where T_1 is the set of pairs of terminals to be separated, T_2 is a set of pairs forbidden to be separated. Now apply the transformation described in the proof of Theorem 2.11 with respect to G and S, where S is the set of all terminals participating in the elements of T_1 and T_2. Let G^* be the resulting graph. Since the treewidth of G^* depends on k and $|S|$, it in fact depends on k and $|T_1| + |T_2|$. Observe that apart from the properties stated in Theorem 2.11, the graph also possesses the following one. For any $C' \subseteq V(G) \cap V(G^*)$, the pairs of terminals of T_2 are not separated in $G \setminus C'$ if and only if they are not separated in $G^* \setminus C'$. This allows us to derive that (G, T_1, T_2, k) is a 'YES' instance of the $\mathcal{G}$-MULTICUT-UNCUT problem if and only if (G^*, T_1, T_2, k) is a 'YES' instance of the $\mathcal{G}$-MULTICUT-UNCUT problem. Indeed, let C be a set of at most k non-terminal vertices of $V(G)$ such that in $G \setminus C$ all pairs terminals of T_1 are separated, no pair of T_2 is separated, and $G[C] \in \mathcal{G}$. Let C^* be a minimal subset of C subject to separation of the pairs of T_1. Clearly, in $G \setminus C^*$ no pair of T_2 and, since $\mathcal{G}$ is a hereditary class, $G[C'] \in \mathcal{G}$. By construction of G^*, C^* separates all pairs of T_1 in G^*, does not separate any pair of T_2 and $G^*[C^*] \in \mathcal{G}$. Thus, if (G, T_1, T_2, k) is a 'YES'-instance of the $\mathcal{G}$-MULTICUT-UNCUT problem, (G^*, T_1, T_2, k) is a 'YES'-instance of this problem as well. The opposite direction can be proven similarly.

Now, let $H = (G^*, \{s_1, t_1\}, \dots \{s_\ell, t_\ell\})$ be a labeled graph where the labels are elements of $T_1 \cup T_2$. We fix a number ℓ' and assume that the first ℓ' pairs are elements of T_1 while the rest are elements of T_2. We construct an MSO formula F of size depending on k and l such that $H \models F$ if and only if (G^*, T_1, T_2, k) is a 'YES' instance of the $\mathcal{G}$-MULTICUT-UNCUT problem. This will imply the present theorem. Denote $\{s_i, t_i\}$ by R_i. Then

$$F = \exists C(\text{AtMost}_k(C) \wedge \text{Induces}_{\mathcal{G}}(C) \wedge \bigwedge_{i=1}^{\ell'} \text{Separates}_{R_i}(C) \wedge \bigwedge_{i=\ell'+1}^{\ell} \neg(\text{Separates}_{R_i}(C))),$$

where $\text{AtMost}_k(C)$ and $\text{Induces}_{\mathcal{G}}(C)$ are as in the proof of Theorem 3.1, $\text{Separates}_{R_i}(C)$ is obtained from $\text{Separates}(C)$ in the proof of Theorem 3.1 by replacing ST by R_i. The verification that F

indeed has the desired property can be done similarly to the verification of the properties of φ done in the proof of Theorem 3.1. ■

Appendix C. Proofs Section 4

Proof of Lemma 4.1. In a 2-coloring of $G \setminus S$, each vertex either has the same color as in (B', W') (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. ■

Proof of Theorem 4.3 . 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 either C is a triangle or C is chordless. Moreover, in the latter case, every vertex v not in C is adjacent to at most 2 vertices of the cycle. To see this, note first that if the length of C is more than 3, then the minimality of C implies that v cannot be adjacent with two adjacent vertices of C (as they would form a triangle). Thus if v has at least 3 (nonadjacent) neighbors in C, then the length of C is at least 7 and v has two neighbors x and y whose distance in C is at least 3. Vertices x and y split C into a path of odd length and a path of even length. Replacing the even-length path (whose length is at least 4) with the path xvy of length 2 gives a shorter odd cycle, contradicting the minimality of C.

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 4.2 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. ■