

Parameterized Complexity Dichotomy for Steiner Multicut*

Karl Bringmann¹, Danny Hermelin², Matthias Mnich³, and Erik Jan van Leeuwen⁴

- 1 Institute of Theoretical Computer Science, ETH Zurich, Zurich, Switzerland
karlb@inf.ethz.ch
- 2 Ben Gurion University of the Negev, Israel
hermelin@bgu.ac.il
- 3 Universität Bonn, Bonn, Germany
mmnich@uni-bonn.de
- 4 Max Planck Institut für Informatik, Saarbrücken, Germany
erikjan@mpi-inf.mpg.de

Abstract

We consider the STEINER MULTICUT problem, which asks, given an undirected graph G , a collection $\mathcal{T} = \{T_1, \dots, T_t\}$, $T_i \subseteq V(G)$, of terminal sets of size at most p , and an integer k , whether there is a set S of at most k edges or nodes such that of each set T_i at least one pair of terminals is in different connected components of $G \setminus S$. This problem generalizes several well-studied graph cut problems, in particular the MULTICUT problem, which corresponds to the case $p = 2$. The MULTICUT problem was recently shown to be fixed-parameter tractable for parameter k [Marx and Razgon, Bousquet et al., STOC 2011]. The question whether this result generalizes to STEINER MULTICUT motivates the present work.

We answer the question that motivated this work, and in fact provide a dichotomy of the parameterized complexity of STEINER MULTICUT on general graphs. That is, for any combination of k , t , p , and the treewidth $\text{tw}(G)$ as constant, parameter, or unbounded, and for all versions of the problem (edge deletion and node deletion with and without deletable terminals), we prove either that the problem is fixed-parameter tractable or that the problem is hard (W[1]-hard or even (para-)NP-complete). Among the many results in the paper, we highlight that:

- The edge deletion version of STEINER MULTICUT is fixed-parameter tractable for parameter $k + t$ on general graphs (but has no polynomial kernel, even on trees).
- In contrast, both node deletion versions of STEINER MULTICUT are W[1]-hard for the parameter $k + t$ on general graphs.
- All versions of STEINER MULTICUT are W[1]-hard for the parameter k , even when $p = 3$ and the graph is a tree plus one node.

Since we allow k , t , p , and $\text{tw}(G)$ to be any constants, our characterization includes a dichotomy for STEINER MULTICUT on trees (for $\text{tw}(G) = 1$) as well as a polynomial time versus NP-hardness dichotomy (by restricting $k, t, p, \text{tw}(G)$ to constant or unbounded).

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases graph cut problems, Steiner cut, fixed-parameter tractability

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.157

* K. Bringmann is supported by the ETH Zurich Postdoctoral Fellowship Program. D. Hermelin has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement number 631163.11, and by the ISRAEL SCIENCE FOUNDATION (grant No. 551145/14).



© Karl Bringmann, Danny Hermelin,
Matthias Mnich, and Erik Jan van Leeuwen;
licensed under Creative Commons License CC-BY

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).

Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 157–170

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



1 Introduction

Graph cut problems are among the most fundamental problems in algorithmic research. The classic result in this area is the polynomial-time algorithm for the s - t cut problem of Ford and Fulkerson [19] (independently proven by Elias et al. [17] and Dantzig and Fulkerson [13]). This result inspired a research program to discover the computational complexity of this problem and of more general graph cut problems. One well-studied generalization of the s - t cut problem is the MULTICUT problem, in which we want to disconnect t pairs of nodes instead of just one pair. In a recent major advance of the research program on graph cut problems, Bousquet et al. [3] and Marx and Razgon [32] showed that MULTICUT is fixed-parameter tractable in the size k of the cut only, meaning that it has an algorithm running in time $f(k) \cdot \text{poly}(|V(G)|)$ for some function f , resolving a longstanding problem in parameterized complexity (with many papers [30, 33, 21, 31] building up to this result).

In this paper, we continue the research program on generalized graph cut problems, and consider the STEINER MULTICUT problem. This problem was proposed by Klein et al. [25], and appears in several versions, depending on whether we want to delete edges or nodes, and whether we are allowed to delete terminal nodes. Formally, these versions of the STEINER MULTICUT problem are defined as follows:

{EDGE, NODE, RESTR. NODE} STEINER MULTICUT

Input: An undirected graph G with terminal sets $T_1, \dots, T_t \subseteq V(G)$, and integer $k \in \mathbb{N}$.

Task: Find a set S of k {edges, nodes, non-terminal nodes} such that for $i = 1, \dots, t$ and at least one pair $u, v \in T_i$ there is no $u - v$ path in $G \setminus S$.

Observe that MULTICUT is the special case of STEINER MULTICUT in which each terminal set has size two. In general, the terminal sets of STEINER MULTICUT can have arbitrary size, and we use p to denote $\max_i |T_i|$.

The complexity of STEINER MULTICUT has been investigated extensively, but so far only from the perspective of approximability. This line of work was initiated by Klein et al. [25], who gave an LP-based $O(\log^3(kp))$ -approximation algorithm. The approximability of several variations of the problem has also been considered [35, 20, 2]; in particular, Garg et al. [20] give an $O(\log t)$ -approximation algorithm for MULTICUT. On the hardness side, even MULTICUT is APX-hard [12, 4] and cannot be approximated within any constant factor assuming the Unique Games Conjecture [6]. We also remark that Steiner cuts (the case when $t = 1$) are of interest: they are an ingredient in several LP-based approximation algorithms (for example for STEINER FOREST [1, 26]) and there is a connection to the number of edge-disjoint Steiner trees that each connect all terminals [29]. To the best of our knowledge, however, STEINER MULTICUT in its general form has not yet been considered from the perspective of parameterized complexity.

1.1 Our Contributions

In this paper, we fully chart the (parameterized) complexity landscape of STEINER MULTICUT according to k , t , p (defined as above), and the treewidth $\text{tw}(G)$. For all three versions of STEINER MULTICUT, for each possible combination of k , t , p , and $\text{tw}(G)$, where each may be either chosen as a constant, a parameter, or unbounded, we consider the complexity of STEINER MULTICUT. We show a complete dichotomy: either we provide a fixed-parameter algorithm with respect to the chosen parameters, or we prove a W[1]-hardness or (para-)NP-completeness result that rules out a fixed-parameter algorithm (unless many canonical NP-complete problems have subexponential- or polynomial-time algorithms respectively).

■ **Table 1** Summary of known and new complexity results for STEINER MULTICUT, where new results are marked with *; the other entries are either known or follow easily from known results in the literature. Only maximal FPT results and minimal W[-] or NP-hardness results are listed; empty cells are dominated by other results. E.g. EDGE STEINER MULTICUT with parameter t is hard, since it is already NP-hard for $t = 3, p = 2$. For NODE STEINER MULTICUT, one also has to apply the rule that $k < t$ (see Section 2) to generate a full characterization of all cases. Tree diagrams of this table are offered in the full version of this paper.

constants	params	EDGE STEINER MC	NODE STEINER MC	RESTR. NODE STEINER MC
k	—	poly (Sect. 2)	poly (Sect. 2)	poly (Sect. 2)
$t \leq 2$	—	poly (Sect. 2)		poly (Sect. 2)
$t = 3, p = 2$	—	NP-h [12]		NP-h [12]
—	k, t	*FPT (Thm. 1)	*W[1]-h (Thm. 2)	*W[1]-h (Thm. 2)
—	k, p, t		FPT (Sect. 2)	FPT (Sect. 2)
t	k			FPT (Sect. 2)
$p = 2$	k	FPT [3, 32]	FPT [3, 32]	FPT [3, 32]
$p = 3, \text{tw} = 2$	k	*W[1]-h (Thm. 3)	*W[1]-h (Thm. 3)	*W[1]-h (Thm. 3)
—	t, tw	*FPT (Thm. 13)	*FPT (Thm. 13)	*FPT (Thm. 13)
$\text{tw} = 1$	—		*poly (Thm. 15)	
$\text{tw} = 1$	k	*W[2]-h (Thm. 16)		*W[2]-h (Thm. 16)
$\text{tw} = 1$	k, p	*FPT (Thm. 18)		*FPT (Thm. 18)
$\text{tw} = 1, p = 2$	—	NP-h [4]		NP-h [4]
$\text{tw} = 2, p = 2$	—		NP-h [4]	

The dichotomy is composed of three main results, along with many smaller ones (see Table 1). These three main results are stated in the three theorems below:

► **Theorem 1.** EDGE STEINER MULTICUT is fixed-parameter tractable for parameter $k + t$.

► **Theorem 2.** NODE STEINER MULTICUT and RESTR. NODE STEINER MULTICUT are W[1]-hard for the parameter $k + t$.

► **Theorem 3.** NODE STEINER MULTICUT, EDGE STEINER MULTICUT, and RESTR. NODE STEINER MULTICUT are W[1]-hard for the parameter k , even if $p = 3$ and $\text{tw}(G) = 2$.

Observe the sharp gap described by Theorem 1 and Theorem 2 between the parameterized complexity of the edge deletion version versus the node deletion version; this gap does not exist for the MULTICUT problem. Also note that Theorem 3 implies that the fixed-parameter algorithms for MULTICUT for parameter k [3, 32] do not generalize to STEINER MULTICUT.

To obtain the fixed-parameter algorithm of Theorem 1, we have to avoid the brute-force choice of a pair of separated terminals of each terminal set: Although one can trivially reduce every instance of the EDGE STEINER MULTICUT problem to at most $\binom{p}{2}^t$ instances of MULTICUT parameterized by k , this only yields an $f(k) \cdot n^{O(t)}$ -time algorithm (for unbounded p). Our contribution in Theorem 1 is that we improve on this simple algorithm and obtain a runtime of $f(k, t) \cdot n^{O(1)}$. We give two independent proofs of Theorem 1:

- Our first proof uses a variation of the recent technique of Chitnis et al. [8] known as randomized contractions (even though the technique actually yields deterministic algorithms). The rough idea of the algorithm is to first determine a large subgraph G'

of the input graph G , such that G' has no small cut and only has a small interface (i.e. a small number of vertices that connect the subgraph to the rest of the graph). We can then branch on the behavior of a solution on the interface to determine a set $U \subseteq E(G')$ of ‘useless’ edges, in the sense that when U is contracted in G a smallest solution (of size at most k) persists in the remaining graph. By iterating this procedure, we can reduce the size of the graph until it is small enough to be handled by exhaustive enumeration. Our algorithm is similar to the one for EDGE MULTIWAY CUT-UNCUT in the paper by Chitnis et al. [8]; however, in contrast to that problem, there seems to be no straightforward projection of the instance onto G' in our case, and therefore more involved arguments are needed to determine the set U .

- Our second proof (presented only in the full version) is based on several novel structural lemmas that show that a minimal edge Steiner cut can be decomposed into important separators and minimal s - t cuts. Using a branching strategy, we ascertain the topology of the decomposition that is promised by the structural lemmas. Since there are only few important separators of bounded size [30, 7, 32] and all relevant minimal s - t cuts lie in a graph of bounded treewidth (following the treewidth reduction techniques of Marx et al. [31]), we can then optimize over important separators and minimal s - t cuts.

The advantage of the first algorithm over the second is that it runs in single-exponential time, instead of double-exponential time. However, the second algorithm is slightly faster in terms of n . Moreover, as part of the correctness proof of the second algorithm, we present some structural lemmas that give additional insight into the properties of the cut, which may be of independent interest. Therefore, we present both algorithms in the full version, but only show the first algorithm in this extended abstract.

The $W[1]$ -hardness results of Theorem 2 and 3 all rely on reductions from the MULTI-COLORED CLIQUE problem [18]. For the proof of Theorem 3, we introduce a novel intermediate problem, NAE-INTEGER-3-SAT, which is an integer variant of the better known NOT-ALL-EQUAL-3-SAT problem. We show that NAE-INTEGER-3-SAT is $W[1]$ -hard parameterized by the number of variables. This is a powerful starting point for parameterized hardness reductions and should turn out to be useful to prove the hardness of other problems.

To complete our dichotomy, we chart the full (parameterized) complexity of STEINER MULTICUT on trees, that is, for graphs G with $\text{tw}(G) = 1$ (these results are mostly deferred to the full version of this paper). In fact, some of the hardness results that we prove for STEINER MULTICUT on general graphs even hold for trees. We also show that many of the results for trees do not carry over to graphs of bounded treewidth, the only exception being a fixed-parameter algorithm for parameters $\text{tw}(G) + t$.

We remark that our characterization induces a polynomial time vs. NP -hardness dichotomy for STEINER MULTICUT, i.e., for any choice of $k, p, t, \text{tw}(G)$ as any constants or unbounded (and all three problem variants), we either prove that STEINER MULTICUT is in P or that it is NP -hard. This characterization can be obtained from Table 1 by considering all its polynomial time and NP -hardness results as well as using the rule that any fixed-parameter algorithm induces a polynomial-time algorithm by setting all parameters to $O(1)$.

1.2 Related Work

We already mentioned several results on the special case of STEINER MULTICUT when $p = 2$ (MULTICUT) [30, 33, 21, 3, 32, 31]. MULTICUT is itself a generalization of MULTIWAY CUT, also known as MULTITERMINAL CUT, where the goal is to delete k edges or nodes to separate all terminals from each other. This problem is NP -complete even for three terminals [12]

and has been extensively studied from a parameterized point of view (see, e.g., the work of Cao et al. [5] or Cygan et al. [11]). The parameterized complexity of many different other graph cut problems has also been considered in recent years [16, 8, 24, 30]. On trees, we only mention here that EDGE MULTICUT and RESTR. NODE MULTICUT remain NP-hard [4], but are fixed-parameter tractable [23, 22]. In contrast, NODE MULTICUT has a polynomial-time algorithm on trees [4].

Organization. We begin our exposition in Section 2 by giving easy results for certain parameter combinations of STEINER MULTICUT. Thereafter, we present our fixed-parameter algorithm for EDGE STEINER MULTICUT (Theorem 1) in Section 3. Following this, in the two subsequent sections, we present our $W[1]$ -hardness proofs: the proof of Theorem 3 in Section 4, and of Theorem 2 in Section 5. Section 6 then focuses on trees to complete our dichotomy. We conclude with some discussion and open problems in Section 7. The full version of this paper is included as an appendix; there we also define basic notions of parameterized complexity.

2 Easy and Known Results

In this section, we collect easy and known results about the STEINER MULTICUT problem. Some of these results are scattered throughout the literature, while others are new. First, observe that whenever the cut size k is constant, we can solve the problem in polynomial time by simply guessing the desired set S of at most k edges or nodes.

Furthermore, NODE STEINER MULTICUT is trivially solvable when $t \leq k$, as in this case we may simply delete an arbitrary terminal node from each set T_i , resulting in a solution of size at most k ; thus, any instance is always a “yes”-instance in this case.

We may reduce STEINER MULTICUT to $\binom{p}{2}^t$ instances of MULTICUT by branching for each terminal set over its separated terminals. Since MULTICUT is in FPT for parameter k , we obtain a fixed-parameter algorithm for STEINER MULTICUT for parameter $k + t + p$. Also, since $\binom{p}{2}^t \leq n^{O(t)}$, STEINER MULTICUT is in FPT for parameter k and any constant t .

Now, MULTICUT on instances with $t = 1$ (i.e. instances that have only one terminal pair $|T_1| = \{s, t\}$) is polynomial-time solvable by running an $s - t$ cut algorithm. For $t = 2$ a result by Yannakakis et al. [34, Lemma 1] also yields a polynomial time algorithm for MULTICUT. Again by branching over the separated terminals in both terminal sets, we obtain a polynomial time algorithm for STEINER MULTICUT for $t \leq 2$.

When there are three or more terminal sets, then STEINER MULTICUT generalizes MULTIWAY CUT and thus is NP-complete [12] even when $p = 2$.

We next show that RESTR. NODE STEINER MULTICUT is as least as hard as NODE STEINER MULTICUT. Therefore, whenever NODE STEINER MULTICUT is $W[1]$ -hard (or NP-hard) for a certain combination of parameters, then so is RESTR. NODE STEINER MULTICUT.

► **Lemma 4.** *Any instance of NODE STEINER MULTICUT can be reduced in polynomial time to an instance of RESTR. NODE STEINER MULTICUT with the same parameter values k , t , p , and $\text{tw}(G)$.*

Proof. Take an instance (G, \mathcal{T}, k) , $\mathcal{T} = \{T_1, \dots, T_t\}$, of NODE STEINER MULTICUT and transform it to an instance of RESTR. NODE STEINER MULTICUT by adding for each terminal node $v \in T_1 \cup \dots \cup T_t$ a new pendant node v' . Then replace v by v' in every terminal set T_i . It is easy to see that the original instance admits a node cut of size k if and only if the new instance admits a node cut of size k that does not use any terminal nodes. ◀

3 Tractability for Edge Deletion and Parameter $k + t$

In this section, we prove Theorem 1, namely that EDGE STEINER MULTICUT parameterized by $k + t$ is fixed-parameter tractable. The proof uses the technique of randomized contractions pioneered by Chitnis et al. [8]; a second, independent proof is deferred to the full version. Later we will see that Theorem 1 is “maximal”, in the sense that EDGE STEINER MULTICUT is $W[1]$ -hard parameterized by k or t alone (this follows from Theorem 16 and the fact that even EDGE MULTICUT is NP-hard when $t = 3$ [12] respectively), that the corresponding node deletion problem is $W[1]$ -hard parameterized by $k + t$ (Theorem 2), and that there exists no polynomial kernel for EDGE STEINER MULTICUT parameterized by $k + t$ (Theorem 17).

We first state some notions and supporting lemmas from the paper of Chitnis et al. [8], which are needed to make our proof work. The *identification* of two vertices $v, w \in V(G)$ results in a graph G' by removing v, w , adding a new vertex vw , and if v or w is an endpoint of an edge, then we replace this endpoint by vw . Note that the identification of two vertices does not remove any edges, and generally results in a multigraph (with parallel edges and self-loops). Without confusion, we may sometimes refer to vw by its old names v or w .

The *contraction* of an edge $(v, w) \in E(G)$ results in a graph G' by removing all edges between v and w , and then identifying v and w . This is also known as contraction without removing parallel edges. Again, the result of a contraction is generally a multigraph. Given a set $F \subseteq E(G)$ of edges that induce a connected subgraph of G with $a + 1$ vertices of which v is one, after contracting all edges of F , we say that a vertices were contracted *onto* v .

► **Definition 5** ([8]). Given two integers a, b , an (a, b) -good edge separation of a connected graph G is a partition (V_1, V_2) of $V(G)$ such that $|V_1|, |V_2| > a$, $G[V_1]$ and $G[V_2]$ are connected, and the number of edges between V_1 and V_2 is at most b .

We now define several notions and prove a few lemmas that are implicit in the work of Chitnis et al. [8].

► **Definition 6.** A b -bordered subgraph of G is a connected induced subgraph G' of G such that in G at most b vertices of $V(G')$ have an edge to a vertex of $V(G) \setminus V(G')$. We call the vertices of G' that have an edge in G to a vertex of $V(G) \setminus V(G')$ the *border vertices* of G' .

► **Lemma 7.** Given a connected graph G and two integers a, b (b even), one can find in time $2^{O(\min\{a, b\} \log(a+b))} |V(G)|^4 \log |V(G)|$ a b -bordered subgraph of G that does not admit an $(a, b/2)$ -good edge separation.

Let G be a connected graph and let a be an integer. Given a set $F \subseteq E(G)$, let G_F denote the graph obtained from G by contracting all edges of F , and then identifying into a single vertex (which we denote by h_F) all vertices onto which at least a vertices were contracted. Observe that G_F is potentially a multigraph, and that h_F might not exist.

A subset Y of the edges of a connected graph G is a *separator* if $G \setminus Y$ has more than one connected component. The set Y is a *minimal separator* if there is no $Y' \subset Y$ such that $G \setminus Y'$ has the same connected components as $G \setminus Y$.

► **Lemma 8** ([8]). Let G be a connected graph, let a, b be two integers (b even), and let $F \subseteq E(G)$ with $|F| \leq b/2$. If G admits no $(a, b/2)$ -good edge separation, then $G \setminus F$ has at most $(b/2) + 1$ connected components, of which at most one has more than a vertices.

► **Lemma 9.** Let G be a connected graph, let a, b be any two integers (b even) such that G does not admit an $(a, b/2)$ -good edge separation and such that $|V(G)| > a(b/2 + 1)$, and let $Y \subseteq E(G)$ with $|Y| \leq b/2$ be a minimal separator. In time $2^{O(b \log(a+b))} |E(G)| \log |E(G)|$

one can find a family \mathcal{F} of $2^{O(b \log(a+b))} \log |E(G)|$ subsets of $E(G)$ that contains a set $F_0 \subseteq E(G)$ with the following properties: (1) $F_0 \cap Y = \emptyset$, (2) h_{F_0} exists in G_{F_0} , (3) h_{F_0} is the identification of a subset of the vertices of a connected component of $G \setminus Y$, and (4) for each connected component C of $G_{F_0} \setminus \{h_{F_0}\}$, Y either contains all edges of $G_{F_0}[C \cup \{h_{F_0}\}]$ or none of these edges.

It is important to observe that the construction of the family \mathcal{F} does not require knowledge of Y itself, beyond that it has size at most $b/2$. Moreover, note that all edges of Y are present in G_{F_0} , as $F_0 \cap Y = \emptyset$.

We are now ready to describe the algorithm for EDGE STEINER MULTICUT for the parameter $k + t$. The basic intuition is to find a part of the graph that does not have a (q, k) -good edge separation for some q , but that only has a small number of border vertices. In this part of the graph we find and contract a set of edges that are provably not part of some smallest edge Steiner multicut. We repeat this procedure until the graph is small enough to be handled by an exhaustive enumeration algorithm.

Consider an instance (G, \mathcal{T}, k) with $\mathcal{T} = \{T_1, \dots, T_t\}$ of EDGE STEINER MULTICUT. We may assume that G is connected. Let q be an integer determined later (q will depend on k and t only). We assume that $|E(G)| > q$, or we can use exhaustive enumeration to solve the problem in $tq^{O(k)}$ time.

We apply the algorithm of Lemma 7 to find a $2k$ -bordered subgraph G' of G that does not admit a (q, k) -good edge separation. Let B denote the set of border vertices of G' . Note that possibly $G' = G$, in which case $B = \emptyset$. The idea is now to determine a set of edges of G' that is not used by some optimal solution.

Let S be a smallest edge Steiner multicut of (G, \mathcal{T}) . Let G'' denote the graph $(B \cup (V(G) \setminus V(G')), E(G) \setminus E(G'))$. Observe that $E(G')$ and $E(G'')$ partition $E(G)$. Let $S' = S \cap E(G')$ and let $S'' = S \cap E(G'')$. We call a terminal set *active* if it is not separated in $G \setminus S'$. We call border vertices $b, b' \in B$ *paired* if there is a path between b and b' in $G'' \setminus S''$. Note that this defines an equivalence relation on B . We call an equivalence class B' of this relation *i -active* if the terminal set T_i is active and the component of $G'' \setminus S''$ that contains B' contains a terminal of T_i . Intuitively, this information suffices to compute a set $Z \subseteq E(G')$ such that $Z \cup S''$ is a smallest edge Steiner multicut of (G, \mathcal{T}) . Then we could contract (in G) all other edges of G' to get a smaller instance, and repeat this until the instance is small enough to be solved by exhaustive enumeration.

Of course, we do not know S , and thus we do not know this equivalence relation on B nor which classes are i -active for each $i = 1, \dots, t$. However, we can branch over all possibilities. In each branch, we find a small set of edges, which we mark. At the end, we contract (in G) the set of edges of G' that were not marked, and thus reduce the size of the instance. We will prove that in one of the branches, we mark a smallest set Z of edges (of size at most k) such that $(Z \cup S'')$ is a smallest edge Steiner multicut (of size at most k) of (G, \mathcal{T}) . Therefore, after contraction, a smallest edge Steiner multicut (of size at most k) of (G, \mathcal{T}) persists (if such a cut existed in the first place). In particular, we argue that we mark Z in the branch with the active classes \mathcal{T}^S , the equivalence relation \mathcal{B}^S , and i -active classes \mathcal{B}_i^S of \mathcal{B}^S for $i = 1, \dots, |\mathcal{T}^S|$ that are induced by S .

The algorithm proceeds by branching over all possibilities. Let $\mathcal{T}' = \{T'_1, \dots, T'_{t'}\}$ be an arbitrary subset of \mathcal{T} , let \mathcal{B} be an arbitrary equivalence relation on B , and let \mathcal{B}_i denote an arbitrary subset of \mathcal{B} for $i = 1, \dots, t'$. We say that we made the *right choice* if $\mathcal{T}' = \mathcal{T}^S$, $\mathcal{B} = \mathcal{B}^S$, and $\mathcal{B}_i = \mathcal{B}_i^S$ for $i = 1, \dots, t'$. The algorithm considers two cases.

Case 1: If $|V(G')| \leq q(k + 1)$, then we can essentially use exhaustive enumeration. Let \tilde{G} be the graph obtained from G' by identifying two border vertices if they are in the same

equivalence class of \mathcal{B} . This also makes each \mathcal{B}_i a set of vertices, which by abuse of notation, we denote by \mathcal{B}_i as well. For $i = 1, \dots, t'$, let \tilde{T}_i be equal to $\mathcal{B}_i \cup (T'_i \cap (V(G') \setminus B))$. Then $\tilde{\mathcal{T}} = \{\tilde{T}_1, \dots, \tilde{T}_{t'}\}$. We verify that no terminal set in $\tilde{\mathcal{T}}$ is a singleton; otherwise, we can continue with the next branch.

► **Lemma 10.** *Assume we made the right choice. Then S' is an edge Steiner multicut of $(\tilde{G}, \tilde{\mathcal{T}})$. Moreover, for any edge Steiner multicut X of $(\tilde{G}, \tilde{\mathcal{T}})$, $X \cup S''$ is an edge Steiner multicut of (G, \mathcal{T}) .*

Now the algorithm uses exhaustive enumeration to find a smallest edge Steiner multicut of $(\tilde{G}, \tilde{\mathcal{T}})$ of size at most k (if one exists) in $t(qk)^{O(k)}$ time. Mark this set of edges in G' .

By Lemma 10, if we made the right choice, we mark a set Z such that $Z \cup S''$ is a smallest edge Steiner multicut of (G, \mathcal{T}) .

Case 2: If $|V(G')| > q(k+1)$, then a more complicated approach is needed, because we cannot just use exhaustive enumeration. In fact, we cannot work with $(\tilde{G}, \tilde{\mathcal{T}})$ directly here, as \tilde{G} might have a (q, k) -good edge separation, even though G' does not.

We proceed as follows. Apply Lemma 9 with $a = q$ and $b = 2k$ to G' with respect to $Y := S'$, and let \mathcal{F} be the resulting family. Note that $Y = S'$ is a minimal separator, $|V(G')| > q(k+1) = a(b/2 + 1)$, and G' has no $(a, b/2)$ -good edge separation, and thus the lemma indeed applies. Consider an arbitrary $F \in \mathcal{F}$. We augment our definition of the *right choice* by adding the condition that $F = F_0$, where F_0 is the family that Lemma 9 promises exists in \mathcal{F} . Now find G'_F . If h_F does not exist in G'_F , then we proceed to the next set F , as Lemma 9 promises that h_F exists if we made the right choice.

We call a set $X \subseteq E(G'_F)$ an *all-or-nothing cut* if for each connected component C' of $G'_F \setminus \{h_F\}$, X either contains all edges of $G'_F[C' \cup \{h_F\}]$ or none of these edges. Note that Lemma 9 promises that S' is an all-or-nothing cut in G'_F for $F = F_0 \in \mathcal{F}$.

Let \mathcal{C} denote the set of connected components of $G'_F \setminus \{h_F\}$ that contain a vertex onto which a border vertex was contracted. Let $Y \subseteq \bigcup_{C \in \mathcal{C}} E(G'_F[C \cup \{h_F\}])$ be an arbitrary set of edges that contains for each $C \in \mathcal{C}$ either all edges of $G'_F[C \cup \{h_F\}]$ or none of these edges. Note that Y is basically an all-or-nothing cut restricted to the edges induced by \mathcal{C} . The algorithm will consider all possible choices of Y . We augment our definition of the *right choice* again, by adding the condition that $Y = Y_0$, where $Y_0 := S' \cap (\bigcup_{C \in \mathcal{C}} E(G'_F[C \cup \{h_F\}]))$.

Now the algorithm deletes all edges in Y and contracts any edges in $(\bigcup_{C \in \mathcal{C}} E(G'_F[C \cup \{h_F\}])) \setminus Y$. Denote the resulting graph by H_F . Let \hat{G} be the graph obtained from H_F by identifying two border vertices if they are in the same equivalence class of \mathcal{B} . This also compresses each \mathcal{B}_i into a set of vertices, which by abuse of notation, we denote by \mathcal{B}_i as well. For $i = 1, \dots, t'$, let \hat{T}_i be equal to $\mathcal{B}_i \cup (T'_i \cap (V(\hat{G}) \setminus B))$. Then $\hat{\mathcal{T}}$ consists of all \hat{T}_i that are not already separated in H_F . We verify that no terminal set in $\hat{\mathcal{T}}$ is a singleton; otherwise, we can continue with the next branch.

► **Lemma 11.** *Assume that we made the right choice. Then $S' \setminus Y$ is an edge Steiner multicut of $(\hat{G}, \hat{\mathcal{T}})$ that is an all-or-nothing cut. Moreover, for any edge Steiner multicut X of $(\hat{G}, \hat{\mathcal{T}})$, $Y \cup X \cup S''$ is an edge Steiner multicut of (G, \mathcal{T}) .*

We now aim to find a smallest edge Steiner multicut X of $(\hat{G}, \hat{\mathcal{T}})$ that is an all-or-nothing cut. Let $\{C'_1, \dots, C'_u\}$ be the set of connected components of $\hat{G} \setminus \{h_F\}$. Let $\hat{\mathcal{T}}|_i$ denote the set of terminal sets in $\hat{\mathcal{T}}$ that are separated if one removes all edges of $E(\hat{G}[\{h_F\} \cup C'_i])$ from G'_F . Define $z[\mathcal{U}, i]$, where $\mathcal{U} \subseteq \hat{\mathcal{T}}$ and $1 \leq i \leq u$, as the size of the smallest all-or-nothing cut

of the terminal sets in \mathcal{U} using only edges in or going out of C'_1, \dots, C'_i . Then for any $\mathcal{U} \subseteq \hat{\mathcal{T}}$,

$$z[\mathcal{U}, 1] = \begin{cases} \infty & \text{if } \mathcal{U} \not\subseteq \hat{\mathcal{T}}|_1 \\ |E(G'_F[\{h_F\} \cup C'_1])| & \text{otherwise (i.e. if } \mathcal{U} \subseteq \hat{\mathcal{T}}|_1) \end{cases}$$

and for $i > 1$, $z[\mathcal{U}, i] = \min \left\{ z[\mathcal{U}, i-1], |E(G'_F[\{h_F\} \cup C'_i])| + z[\mathcal{U} \setminus \hat{\mathcal{T}}|_i, i-1] \right\}$. Note that $z[\hat{\mathcal{T}}, u]$ holds the size of the smallest edge Steiner multicut of $(\hat{G}, \hat{\mathcal{T}})$ that is an all-or-nothing cut (if one exists). Finding the set achieving this smallest size is straightforward from the dynamic-programming table. Finally, over all choices of F and all choices of Y , mark in G' the smallest set of edges that was found if it has size at most k .

By Lemma 11, if we made the right choice, we mark a set Z such that $Z \cup S''$ is a smallest edge Steiner multicut of (G, \mathcal{T}) .

In both cases, let M be the set of marked edges. Now we contract all unmarked edges $E(G') \setminus M$ in G . Let \tilde{G} denote the resulting graph; note that in general \tilde{G} is a multigraph. Each time we contract an edge between two vertices u and v , we replace u and v by uv in all terminal sets in \mathcal{T} . Let $\tilde{\mathcal{T}}$ denote the resulting set of terminal sets. Observe that if a terminal set \tilde{T}_i in $\tilde{\mathcal{T}}$ is a singleton set and T_i was not a singleton set in \mathcal{T} , then we can answer “no”. Now it remains to prove that (G, \mathcal{T}, k) is a “yes”-instance if and only if $(\tilde{G}, \tilde{\mathcal{T}}, k)$ is. For this, it suffices to note that an edge Steiner multicut of $(\tilde{G}, \tilde{\mathcal{T}})$ corresponds directly to an edge Steiner multicut of (G, \mathcal{T}) , and that for any smallest edge Steiner multicut S of (G, \mathcal{T}) there is a smallest edge Steiner multicut $Z \cup (S \setminus E(G'))$ of (G, \mathcal{T}) such that $Z \subseteq M$.

Since there are at most $2k$ border vertices and t terminal sets, there are $r = 2^{O(kt \log k)}$ different branches that we consider for \mathcal{T}' , \mathcal{B} , and \mathcal{B}_i , and in each we mark at most k edges. Choose $q = rk + 1$. Now note that $|E(G')| \geq q$. If $G = G'$, then this follows from the assumption that $|E(G)| > q$. If $G \neq G'$, then G' was obtained after considering multiple (q, k) -good separations. Hence, $|V(G')| > q$, and since G' is connected, $|E(G')| \geq q$. Since $q = rk + 1$, at least one edge of G' was not marked and thus contracted. Therefore, $|V(G)|$ decreases by at least one, and the entire procedure finishes after at most $|V(G)|$ iterations.

To analyze the running time, note that the dynamic-programming algorithm requires time $2^{O(t)k} + O(t|E(G)|)$. The family \mathcal{F} contains $2^{O(k^2 t \log k)} \log |E(G)|$ sets and can be constructed in $2^{O(k^2 t \log k)} |E(G)| \log |E(G)|$ time. Hence, Case 2 runs in $2^{O(k^2 t \log k)} |E(G)| \log |E(G)|$ time. Case 1 runs in $2^{O(k^2 t \log k)}$ time. Since there are $r = 2^{O(kt \log k)}$ different branches for \mathcal{T}' , \mathcal{B} , and \mathcal{B}_i that we consider, and it takes $2^{O(k^2 t \log k)} |V(G)|^4 \log |V(G)|$ time to find a $2k$ -bordered subgraph that does not admit a (q, k) -good separation, each iteration takes $2^{O(k^2 t \log k)} |V(G)|^4 \log |V(G)|$ time. Since there are at most $|V(G)|$ iterations, the total running time is $2^{O(k^2 t \log k)} |V(G)|^5 \log |V(G)|$. Actually, using the recurrence outlined by Chitnis et al. [8], one can show a bound on the running time of $2^{O(k^2 t \log k)} |V(G)|^4 \log |V(G)|$.

4 Steiner Multicuts for Graphs of Bounded Treewidth

In this section, we consider STEINER MULTICUT on graphs of bounded treewidth. To start the exposition, we note that EDGE STEINER MULTICUT and RESTR. NODE STEINER MULTICUT are NP-complete for trees and NODE STEINER MULTICUT is NP-complete on series-parallel graphs [4], which are graphs of treewidth two. This means that any efficient algorithm for STEINER MULTICUT on graphs of bounded treewidth needs an additional parameter.

We first show Theorem 3, namely that all variants of STEINER MULTICUT for the parameter k are W[1]-hard, even if $p = 3$ and $\text{tw}(G) = 2$ (but t is unbounded). The graph G is in fact a tree plus one node. We then contrast this result by showing that the problem is fixed-parameter tractable on bounded treewidth graphs when t is a parameter.

The reduction for Theorem 3 is from an intermediate problem, (MONOTONE) NAE-INTEGGER-3-SAT. In this problem, we are given variables x_1, \dots, x_k that each take a value in $\{1, \dots, n\}$ and clauses C_1, \dots, C_m of the form $\text{NAE}(x_{i_1} \leq a_1, x_{i_2} \leq a_2, x_{i_3} \leq a_3)$, $a_1, a_2, a_3 \in \{1, \dots, n\}$, which is satisfied if not all three inequalities are true and not all are false (i.e., they are “not all equal”). The goal is to find an assignment of the variables that satisfies all given clauses. We remark that NAE-INTEGGER-3-SAT generalizes MONOTONE NAE-3-SAT (by restriction to $n = 2$), and that NAE-INTEGGER-3-SAT can be solved in time $O(m \cdot n^k)$, by enumerating all assignments.

► **Lemma 12.** *NAE-INTEGGER-3-SAT is W[1]-hard for parameter k .*

Proof. Let (G, k) be an instance of MULTICOLORED CLIQUE [18]. We use V_i to denote the set of vertices of color i , $n_i = |V_i|$, and $E_{i,j}$ to denote the set of edges with one endpoint in V_i and the other in V_j . We create an instance of NAE-INTEGGER-3-SAT on variables x_i , one for each color $1 \leq i \leq k$, and y_{ij} , one for each pair of colors $1 \leq i < j \leq k$. We identify the vertices V_i with the integers $\{1, \dots, n_i\}$ in an arbitrary way. We restrict x_i to $\{1, \dots, n_i\}$ using the clause $\text{NAE}(x_i \leq 0, x_i \leq 0, x_i \leq n_i)$, and write $x_i = u$ if the number x_i corresponds to vertex u . Analogously, we can identify the edges $uv \in E_{i,j}$ with numbers in $\{1, \dots, |E_{i,j}|\}$ and write $y_{ij} = uv$ if we pick the number corresponding to edge uv . Consider the following constraints (for any edge uv , with u of color i and v of color j), $y_{ij} = uv \Rightarrow x_i = u$, $y_{ij} = uv \Rightarrow x_j = v$. If we can encode these constraints with NAE-clauses, then any satisfying assignment of the constructed NAE-INTEGGER-3-SAT instance corresponds to a clique in G , as all chosen pairs y_{ij} correspond to edges, and edges sharing a color i picked the same vertex x_i . We focus on the first constraint; the second is similar. Note that the constraint is equivalent to $y_{ij} = uv \Rightarrow x_i \geq u$, $y_{ij} = uv \Rightarrow x_i \leq u$. Again, w.l.o.g., we focus on the first of these constraints. It is equivalent to $y_{ij} < uv \vee y_{ij} > uv \vee x_i \geq u$, which in turn can be written as $\text{NAE}(y_{ij} < uv, y_{ij} > uv, x_i \geq u)$, since $y_{ij} < uv, y_{ij} > uv$ cannot both be true. Note that we can replace any inequality $x < a$ by $x \leq a - 1$ (and similarly for $x > a$). Hence, we can encode all desired constraints if we may use “ \leq ” and “ \geq ” inequalities, not only “ \leq ” inequalities, as is the case in the definition of NAE-INTEGGER-3-SAT.

In the remainder of this proof, we reduce NAE-INTEGGER-3-SAT with “ \leq ” and “ \geq ” inequalities to the original variant with only “ \leq ” inequalities. Given any instance of NAE-INTEGGER-3-SAT with both types of inequalities, for any variable x we introduce a new variable \bar{x} . For any $1 \leq v \leq n$, we add the constraint $\text{NAE}(x \leq v, x \leq v, \bar{x} \leq n - v)$. This enforces $\bar{x} = n + 1 - x$. Finally, we replace any inequality $x \geq v$ by $\bar{x} \leq n + 1 - v$. This yields an equivalent NAE-INTEGGER-3-SAT instance with only “ \leq ” inequalities. ◀

Proof of Theorem 3. We first give a reduction from NAE-INTEGGER-3-SAT to EDGE STEINER MULTICUT (satisfying $p = 3$ and $\text{tw}(G) = 2$). Consider an instance of NAE-INTEGGER-3-SAT on variables x_1, \dots, x_k taking values in $\{1, \dots, n\}$ with clauses C_1, \dots, C_m . Take k paths consisting of n edges and identify their start nodes (to a common node s) and end nodes (to a common node t), respectively. The resulting graph G has $\text{tw}(G) = 2$, since it is not a tree, but becomes a tree after deleting s (or t). Let v_j^i be the j -th node on the i -th path from s to t , so that $v_0^i = s$ and $v_n^i = t$. For each clause $\text{NAE}(x_{i_1} \leq a_1, x_{i_2} \leq a_2, x_{i_3} \leq a_3)$ we introduce a terminal set $\{v_{a_1}^{i_1}, v_{a_2}^{i_2}, v_{a_3}^{i_3}\}$ (note that we can assume $0 \leq a_j \leq n$ without loss of generality). Further, we let $\{s, t\}$ be a terminal set and set the cut size to k , i.e., we allow to delete k edges. This finishes the construction. In order to separate s from t we need to cut at least one edge of each of the k paths that connect s and t , and because the cut size is k we have to delete *exactly one* edge per path. Say we delete the x_i -th edge on the i -th path. This splits G into two components, one containing s and the other containing t . Note that

we separate nodes v_j^i and $v_{j'}^{i'}$ by cutting at $x_i \leq j$ and $x_{i'} > j'$ (or with both inequalities the other way round), since then v_j^i is in the t -component and $v_{j'}^{i'}$ in the s -component. Hence, the following are equivalent:

- the terminal set $\{v_{a_1}^{i_1}, v_{a_2}^{i_2}, v_{a_3}^{i_3}\}$ is disconnected;
- some pair of nodes in this set is disconnected;
- among the inequalities $x_{i_j} \leq a_j$, $j = 1, 2, 3$, one is true and one is false;
- the clause $\text{NAE}(x_{i_1} \leq a_1, x_{i_2} \leq a_2, x_{i_3} \leq a_3)$ is satisfied.

Therefore, the given NAE-INTEGGER-3-SAT instance is equivalent to the constructed EDGE STEINER MULTICUT instance.

We can adapt this construction to prove hardness of NODE STEINER MULTICUT; hardness of RESTR. NODE STEINER MULTICUT then follows from Lemma 4. ◀

We contrast the above theorem with the following result by showing that STEINER MULTICUT is fixed-parameter tractable for the parameter t if the graph has bounded treewidth, through an MSOL-formula.

► **Theorem 13.** NODE STEINER MULTICUT, EDGE STEINER MULTICUT, and RESTR. NODE STEINER MULTICUT are fixed-parameter tractable for the parameter $t + \text{tw}(G)$.

5 Hardness for Cutsizes k and Number of Terminal Sets t

In this section, we consider the STEINER MULTICUT problem on general graphs parameterized by $k+t$. We show that both node deletion versions of the problem, NODE STEINER MULTICUT and RESTR. NODE STEINER MULTICUT, are $W[1]$ -hard for this parameter.

► **Theorem 14.** NODE STEINER MULTICUT and RESTR. NODE STEINER MULTICUT are $W[1]$ -hard for the parameter $k+t$.

Proof. We present a parameterized reduction from the MULTICOLORED CLIQUE problem [18] to NODE STEINER MULTICUT. Let (H, k) be an instance of MULTICOLORED CLIQUE, and let V_i and $E_{i,j}$ be as in the definition of MULTICOLORED CLIQUE. We then create the following instance of NODE STEINER MULTICUT. First, we subdivide each edge of H , and let $N_{i,j}$ denote the set of nodes that were created when subdividing the edges of $E_{i,j}$. Then, add a complete graph C with $2k$ nodes, where we denote the nodes of C by c_1, \dots, c_{2k} , and make all nodes of V_i adjacent to c_{2i-1} and c_{2i} for each $i = 1, \dots, k$. Let G denote the resulting graph. Observe that $G[V(H)]$ and $G[\bigcup_{i,j} N_{i,j}]$ are both independent sets of G . We then create terminal sets $T_i = V_i \cup \{c_{2i-1}\}$ and $T'_i = V_i \cup \{c_{2i}\}$, and terminal sets $T_{i,j} = N_{i,j} \cup V(C)$. Let $\mathcal{T} = \{T_i, T'_i \mid i = 1, \dots, k\} \cup \{T_{i,j} \mid i \neq j, i, j = 1, \dots, k\}$. Then the created instance is (G, \mathcal{T}, k) .

Suppose that (H, k) is a “yes”-instance of MULTICOLORED CLIQUE, and let K denote a clique of H such that $V(K) \cap V_i \neq \emptyset$ for each i . Pick a node $v_i \in V(K) \cap V_i$ and let $S = \{v_i \mid i = 1, \dots, k\}$. Observe that v_i disconnects terminal sets T_i and T'_i . Further, if we let $n_{i,j}$ denote the subdivision node of the edge $(v_i, v_j) \in E(H)$, then v_i and v_j disconnect $n_{i,j}$ from the rest of $T_{i,j}$. Finally, $|S| = k$. Therefore, (G, \mathcal{T}, k) is a “yes”-instance of NODE STEINER MULTICUT.

Suppose that (G, \mathcal{T}, k) is a “yes”-instance of NODE STEINER MULTICUT, and let $S \subseteq V(G)$ denote a node Steiner multicut of G with respect to terminal sets \mathcal{T} such that $|S| \leq k$. We claim that $H[S]$ is a multicolored clique of H . First, observe that to disconnect T_i and T'_i , we need that $S \cap T_i \neq \emptyset$ and $S \cap T'_i \neq \emptyset$. Since $|S| \leq k$, we know that $S \cap T_i \cap T'_i \neq \emptyset$ for each $i = 1, \dots, k$. This implies that $|S| = k$, that $S \subseteq V(H)$, and that $S \cap V_i \neq \emptyset$ for each

$i = 1, \dots, k$. It remains to show that $H[S]$ is a clique. Let v_i denote the node in $S \cap V_i$. Suppose that $(v_i, v_j) \notin E(H)$ for some i, j . Then consider the terminal set $T_{i,j}$, and observe that for any node $n \in N_{i,j}$ at least one endpoint of the edge corresponding to n is not in S . Since $S \cap V(C) = \emptyset$, this implies that $T_{i,j}$ is not disconnected by S , a contradiction. It follows that $H[S]$ is a clique, and thus (H, k) is a “yes”-instance of MULTICOLORED CLIQUE.

For RESTR. NODE STEINER MULTICUT hardness follows from the statement for NODE STEINER MULTICUT and Lemma 4. ◀

6 Steiner Multicuts in Trees

We state the following results on trees; the proofs are in the full version. The first theorem generalizes the algorithm for NODE MULTICUT on trees [4].

► **Theorem 15.** NODE STEINER MULTICUT *can be decided in linear time on trees.*

The next theorems follow from a reduction from HITTING SET [15, 14].

► **Theorem 16.** EDGE STEINER MULTICUT *and* RESTR. NODE STEINER MULTICUT *are* $W[2]$ -*hard on trees for the parameter* k .

► **Theorem 17.** EDGE STEINER MULTICUT *and* RESTR. NODE STEINER MULTICUT *have no polynomial kernel on trees for the parameter* $k + t$, *unless the polynomial hierarchy collapses to the third level.*

Note that the above theorem complements the FPT result of Theorem 1.

The final theorem uses a branching strategy in which an incident edge or neighbor needs to be chosen for the lowest common ancestor of the terminals in any terminal set.

► **Theorem 18.** EDGE STEINER MULTICUT *and* RESTR. NODE STEINER MULTICUT *are fixed-parameter tractable on trees for the parameter* $k + p$.

7 Discussion

We provided a comprehensive computational complexity analysis of the STEINER MULTICUT problem with respect to fundamental parameters, culminating in either a fixed-parameter algorithm or a $W[1]$ -hardness result for *every* combination of parameters. This way, we generalize known tractability results for special cases of STEINER MULTICUT, and chart the boundary of tractability for other cases. See Table 1 for a complete overview.

We leave several interesting questions for future research. A possible extension is to consider directed graphs. Already MULTICUT is $W[1]$ -hard in this case [32] for parameter cut size k , even on acyclic directed graphs [27]. On the other hand, MULTICUT is fixed-parameter tractable for the parameter $k + t$ in directed acyclic graphs [27]. It would be interesting whether this result generalizes to STEINER MULTICUT.

Another possible extension is to investigate which problems admit polynomial kernels. While we have resolved many kernelization questions in the full version of this paper, several open problems remain, in particular whether there is a polynomial kernel for the parameters $k + t + p$ on general graphs. Answers in this research direction might shed new light on some long-standing open questions [9] on the existence of polynomial kernels for MULTICUT for parameter $k + t$ (currently, only a kernel of size $k^{O(\sqrt{t})}$ is known [28], and there is no kernel of size polynomial in k only [10]).

Acknowledgements. We thank Magnus Wahlström for an insight that helped in proving Lemma 12.

References

- 1 Ajit Agrawal, Philip Klein, and R. Ravi. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Comput.*, 24(3):440–456, 1995.
- 2 Adi Avidor and Michael Langberg. The multi-multiway cut problem. *Theoret. Comput. Sci.*, 377(1-3):35–42, 2007.
- 3 Nicolas Bousquet, Jean Daligault, and Stéphan Thomassé. Multicut is FPT. In *Proc. STOC 2011*, pages 459–468, New York, NY, USA, 2011. ACM.
- 4 Gruia Călinescu, Cristina G. Fernandes, and Bruce Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. *J. Algorithms*, 48(2):333–359, 2003.
- 5 Yixin Cao, Jianer Chen, and Jia-Hao Fan. An $O^*(1.84^k)$ parameterized algorithm for the multiterminal cut problem. *Information Processing Letters*, 114(4):167–173, 2014.
- 6 Shuchi Chawla, Robert Krauthgamer, Ravi Kumar, Yuval Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Comput. Complexity*, 15(2):94–114, 2006.
- 7 Jianer Chen, Yang Liu, and Songjian Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.
- 8 Rajesh Chitnis, Marek Cygan, MohammadTaghi Hajiaghayi, Marcin Pilipczuk, and Michał Pilipczuk. Designing FPT algorithms for cut problems using randomized contractions. In *Proc. FOCS 2012*, pages 460–469, Washington, DC, USA, 2012. IEEE Computer Society.
- 9 Marek Cygan, Łukasz Kowalik, and Marcin Pilipczuk. Open problems from workshop on kernels, April 2013. <http://worker2013.mimuw.edu.pl/slides/worker-opl.pdf>.
- 10 Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Magnus Wahlström. Clique cover and graph separation: new incompressibility results. *ACM Transactions on Computation Theory (TOCT)*, 6(2), 2014.
- 11 Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *ACM Transactions on Computation Theory (TOCT)*, 5(1), 2013.
- 12 E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.
- 13 G. B. Dantzig and D. R. Fulkerson. On the max-flow min-cut theorem of networks. In *Linear inequalities and related systems*, Annals of Mathematics Studies, no. 38, pages 215–221. Princeton University Press, Princeton, N. J., 1956.
- 14 Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and IDs. In *Proc. ICALP 2009*, volume 5555 of *Lecture Notes Comput. Sci.*, pages 378–389. Springer, Berlin, 2009.
- 15 R. G. Downey and M. R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.
- 16 Rodney G. Downey, Vladimir Estivill-Castro, Michael Fellows, Elena Prieto, and Frances A. Rosamond. Cutting up is hard to do: The parameterised complexity of k -cut and related problems. *Electr. Notes Theor. Comput. Sci.*, 78(0):209 – 222, 2003.
- 17 P. Elias, A. Feinstein, and C.E. Shannon. A note on the maximum flow through a network. *Information Theory, IRE Transactions on*, 2(4):117–119, 1956.
- 18 Michael R. Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoret. Comput. Sci.*, 410(1):53–61, 2009.

- 19 L. R. Ford, Jr. and D. R. Fulkerson. Maximal flow through a network. *Canad. J. Math.*, 8:399–404, 1956.
- 20 Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput.*, 25(2):235–251, 1996.
- 21 Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optim.*, 8(1):61–71, 2011.
- 22 Jiong Guo, Falk Hüffner, Erhan Kenar, Rolf Niedermeier, and Johannes Uhlmann. Complexity and exact algorithms for multicut. In *Proc. SOFSEM 2006*, volume 3831 of *Lecture Notes Comput. Sci.*, pages 303–312, Berlin, 2006. Springer.
- 23 Jiong Guo and Rolf Niedermeier. Fixed-parameter tractability and data reduction for multicut in trees. *Networks*, 46(3):124–135, 2005.
- 24 Ken-ichi Kawarabayashi and Mikkel Thorup. The minimum k -way cut of bounded size is fixed-parameter tractable. In *Proc. FOCS 2011*, pages 160–169. IEEE Computer Soc., Los Alamitos, CA, 2011.
- 25 Philip N. Klein, Serge A. Plotkin, Satish Rao, and Éva Tardos. Approximation algorithms for Steiner and directed multicuts. *J. Algorithms*, 22(2):241–269, 1997.
- 26 Jochen Könemann, Stefano Leonardi, Guido Schäfer, and Stefan H. M. van Zwam. A group-strategyproof cost sharing mechanism for the Steiner Forest game. *SIAM J. Comput.*, 37(5):1319–1341, 2008.
- 27 Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, and Magnus Wahlström. Fixed-parameter tractability of multicut in directed acyclic graphs. In *Proc. ICALP 2012*, volume 7391 of *Lecture Notes Comput. Sci.*, pages 581–593. Springer, Berlin, 2012.
- 28 Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *Proc. FOCS 2012*, pages 450–459, 2012.
- 29 Lap Chi Lau. An approximate max-Steiner-tree-packing min-Steiner-cut theorem. *Combinatorica*, 27(1):71–90, 2007.
- 30 Dániel Marx. Parameterized graph separation problems. *Theoret. Comput. Sci.*, 351(3):394–406, 2006.
- 31 Dániel Marx, Barry O’Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Trans. Algorithms*, 9(4):30:1–30:35, October 2013.
- 32 Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *Proc. STOC 2011*, pages 469–478, New York, NY, USA, 2011. ACM.
- 33 Mingyu Xiao. Simple and improved parameterized algorithms for multiterminal cuts. *Theory Comput. Syst.*, 46(4):723–736, 2010.
- 34 Mihalis Yannakakis, Paris C. Kanellakis, Stavros S. Cosmadakis, and Christos H. Papadimitriou. Cutting and partitioning a graph after a fixed pattern (extended abstract). In *Proc. ICALP 1983*, volume 154 of *Lecture Notes Comput. Sci.*, pages 712–722, London, UK, UK, 1983. Springer-Verlag.
- 35 Bo Yu and Joseph Cheriyan. Approximation algorithms for feasible cut and multicut problems. In *Proc. ESA 1995*, volume 979 of *Lecture Notes Comput. Sci.*, pages 394–408. Springer, Berlin, 1995.