Clique cover and graph separation: New incompressibility results*

Marek Cygan† Stefan Kratsch‡§ Marcin Pilipczuk† Michał Pilipczuk¶
Magnus Wahlström‖

January 27, 2015

Abstract

The field of kernelization studies polynomial-time preprocessing routines for hard problems in the framework of parameterized complexity. In this paper we show that, unless the polynomial hierarchy collapses to its third level, the following parameterized problems do not admit a polynomial-time preprocessing algorithm that reduces the size of an instance to polynomial in the parameter:

- Edge Clique Cover, parameterized by the number of cliques,
- Directed Edge/Vertex Multiway Cut, parameterized by the size of the cutset, even in the case of two terminals,
- Edge/Vertex Multicut, parameterized by the size of the cutset, and
- k-Way Cut, parameterized by the size of the cutset.

1 Introduction

In order to cope with the NP-hardness of many natural combinatorial problems, various algorithmic paradigms such as brute-force, approximation, or heuristics may be applied. However, while the paradigms are quite different, there is a commonly used opening move of first applying polynomial-time preprocessing routines, before making sacrifices in either exactness or runtime. The aim of the field of kernelization is to provide a rigorous mathematical framework for analyzing such preprocessing algorithms. One of its core features is to provide quantitative performance guarantees for preprocessing via the framework of parameterized complexity, a feature easily seen to be infeasible in classical complexity (cf. [38]).

In the framework of parameterized complexity an instance x of a parameterized problem comes with an integer parameter k, formally, a parameterized problem is defined as $Q \subseteq \Sigma^* \times \mathbb{N}$ for

*An extended abstract of this work appeared in ICALP 2012 [16].
†University of Warsaw, Poland, {cygan,malcin}@mimuw.edu.pl
‡Technical University Berlin, Germany, stefan.kratsch@tu-berlin.de
§Work done while at Utrecht University supported by the Netherlands Organization for Scientific Research (NWO), project "KERNELS: Combinatorial Analysis of Data Reduction".
¶Bergen University, Norway, michal.pilipczuk@gmail.com
‖Max-Planck-Institute for Informatics, Saarbrücken, Germany, wahl@mpi-inf.mpg.de

some finite alphabet Σ. We say that a problem is *fixed parameter tractable* (*FPT*) if there exists an algorithm solving any instance (x, k) in time $f(k)\text{poly}(|x|)$ for some (usually exponential) computable function f. It is known that a problem is FPT if and only if it has a kernelization, which is defined as follows: A *kernelization* (*kernel* for short) for a problem Q is a polynomial time preprocessing routine that takes an instance (x, k) and in time polynomial in $|x| + k$ produces an equivalent instance (x', k') (i.e., $(x, k) \in Q$ if and only if $(x', k') \in Q$) such that $|x'| + k' \leq g(k)$ for some computable function g. The function g is the *size of the kernel*, and if it is polynomial, we say that Q admits a polynomial kernel. If g is small, after preprocessing even an exponential-time brute-force algorithm might be feasible. Therefore, small kernels, with g being linear or polynomial, are of big interest. Polynomial kernels for a wide range of problems have been developed for the last few decades; see the surveys of Guo and Niedermeier [37], Bodlaender [4], and Lokshtanov et al. [51].

Still, a framework for proving kernelization lower bounds was discovered only recently by Bodlaender et al. [5], with the backbone theorem proven by Fortnow and Santhanam [27]. The crux of the framework is the following idea of a composition. Assume we are able to combine in polynomial time an arbitrary number of instances $x_1, x_2, \ldots, x_t$ of an NP-complete problem L into a single instance (x, k) of a parameterized problem Q such that $(x, k) \in Q$ if and only if at least one of the instances x_i is in L, while k is bounded polynomially in $\max_i |x_i|$. If such a *composition* algorithm was pipelined with a polynomial kernel for the problem Q, we would obtain an OR-distillation of the NP-complete language L: the resulting instance is of size polynomial in $\max_i |x_i|$, possibly significantly smaller than t, but encodes a disjunction of all input instances x_i (i.e., an OR-distillation is a compression of the logical OR of the instances). As proven by Fortnow and Santhanam [27], the existence of such an algorithm would imply NP $\subseteq$ coNP/poly, which is known to cause a collapse of the polynomial hierarchy to its third level [10, 67].

The astute reader may have noticed that the above description of a composition is actually using the slightly newer notion of a cross-composition [**?**]. This generalization of the original lower bound framework will be the main ingredient of our proofs. The framework of kernelization lower bounds was also extended by Dell and van Melkebeek [22] to allow excluding kernels of some particular exponent in the polynomial. Recently, Dell and Marx [21] and, independently, Hermelin and Wu [39] simplified this approach and applied it to various packing problems.

The aforementioned (cross-)composition algorithm is sometimes called an *OR-composition*, as opposed to an *AND-composition*, where we require that the output instance (x, k) is in Q if and only if *all* input instances belong to L. Various problems have been shown to be AND-compositional, with the most important example being the problem of determining whether an input graph has treewidth no larger than the parameter [5]. It was conjectured by Bodlaender et al. [5] that no NP-complete problem admits an AND-distillation, which would be a result of pipelining an AND-composition with a polynomial kernel. In a recent breakthrough work Drucker [25] proved the conjecture assuming that NP $\not\subseteq$ coNP/poly (among other results).

Although the framework of kernelization lower bounds has been applied successfully multiple times over the last four years, there are still many important problems where the existence of a polynomial kernel is widely open. A major reason for this situation is that the application of the idea of a composition (or an appropriate reduction, see [7]) is far from being automatic. To obtain a composition algorithm, usually one needs to carefully choose the starting language L (for example, the choice of the starting language is crucial for compositions of Dell and Marx [21], and the core idea of the composition algorithms for connectivity problems in degenerate graphs [17] is to use

Graph Motif as a starting point) or invent sophisticated gadgets to merge the instances (for example, the colors and IDs technique introduced by Dom et al. [23] or the idea of an instance selector, used mainly for structural parameters [**?**, 6]).

Our results. The main contribution of this paper is a proof of non-existence of polynomial kernels for four important problems.

Theorem 1. *Unless NP $\subseteq$ coNP/poly,* Edge Clique Cover*, parameterized by the number of cliques, as well as* Directed Multiway Cut, Multicut*, and k-*Way Cut*, parameterized by the size of the cutset, do not admit polynomial kernelizations.*

The common theme of our compositions is a very careful choice of starting problems. Not only do we select particular NP-complete problems, but we also restrict instances given as the input, to make them satisfy certain conditions that allow designing cross-compositions. Each time we constrain the set of input instances of an NP-complete problem we need to prove that the problem remains NP-complete. Even though this paper is about negative results, in our constructions we use intuition derived from the design of parameterized algorithms techniques, including iterative compression (in case of Edge Clique Cover) introduced by Reed et al. [63] and important separators (in case of Multicut) defined by Marx [54].

For the three cut problems listed in Theorem 1 our kernelization hardness results complement recent developments in the design of algorithm parameterized by the size of the cutset [8, 15, 55, 45]. In the following we give some motivation and related work for each of the four problems.

Edge clique cover. In the Edge Clique Cover problem the goal is to cover the edges of an input graph G with at most k cliques all of which are subgraphs of G. This problem, NP-complete even in very restricted graph classes [12, 40, 57], is also known as Covering by Cliques (GT17), Intersection Graph Basis (GT59) [28], and Keyword Conflict [46]. It has multiple applications in various areas in practice, such as computational geometry [1], applied statistics [33, 58], and compiler optimization [59]. In particular, Edge Clique Cover is equivalent to the problem of finding a representation of a graph G as an intersection model with at most k elements in the universe [26, 34, 64]. Therefore, an algorithm for Edge Clique Cover may be used to reveal a structure in a complex real-world network [35]. Due to its importance, the Edge Clique Cover problem was studied from various perspectives, including approximation upper and lower bounds [2, 52], heuristics [3, 33, 46, 47, 58, 59], and polynomial-time algorithms for special graph classes [40, 41, 53, 57].

From the point of view of parameterized complexity, Edge Clique Cover was studied by Gramm et al. [32]. A simple kernelization algorithm is known that reduces the size of the graph to at most 2^k vertices; the best known fixed-parameter algorithm is a brute-force search on this 2^k-vertex kernel. The question of a polynomial kernel for Edge Clique Cover, probably first verbalized by Gramm et al. [32], was repeatedly asked in the parameterized complexity community, for example on the last Workshop on Kernels (WorKer, Vienna, 2011). We provide an AND-cross-composition from (unparameterized) Edge Clique Cover to Edge Clique Cover parameterized by k, thereby establishing that a polynomial kernelization is unlikely to exist.

Multicut and directed multiway cut. With Multicut and Directed Multiway Cut we move on to the family of graph separation problems. The central problems of this area are two natural generalizations of the $s-t$ cut problem, namely Multiway Cut and Multicut. In the first problem we are given a graph G with designated terminals and we are to delete at most p edges (or vertices, depending on the variant) so that the terminals end up in different connected components. In the Multicut problem we consider a more general setting where the input graph contains terminal *pairs* and we need to separate all pairs of terminals.

As generalizations of the well-known $s-t$ cut problem, Multiway Cut and Multicut received a lot of attention in past decades. Multiway Cut is NP-complete even for the case of three terminals [20], thus the same holds for Multicut with three terminal pairs. Both problems were intensively studied from the approximation perspective [11, 29, 30, 43, 56]. The graph separation problems became one of the most important subareas in parameterized complexity after Marx introduced the concept of important separators [54]. This technique turns out to be very robust, and is now a key ingredient in fixed-parameter algorithms for various problems such as variants of the Feedback Vertex Set problem [14, 19] or Almost 2-SAT [62]. A long line of research on Multiway Cut in the parameterized setting include [13, 18, 36, 54, 60, 61, 66]; the currently fastest algorithm runs in $\mathcal{O}(2^p n^{\mathcal{O}(1)})$ time [18]. It is not very hard to prove that Multicut, parameterized by both the number of terminals and the size of the cutset, is FPT-reducible to Multiway Cut [54]. Fixed-parameter tractability of Multicut parameterized by the size of the cutset only, after being a big open problem for a few years, was finally resolved positively in 2010 [8, 55].

In directed graphs Multiway Cut is NP-complete even for two terminals [30]. Very recently Chitnis et al. [15] showed that Directed Multiway Cut is fixed-parameter tractable. The directed version of Multicut, parameterized by the size of the cutset, is $W[1]$-hard [55] (i.e., an existence of a fixed-parameter algorithm is unlikely). The parameterized complexity of Directed Multicut with fixed number of terminal pairs or with the number of terminal pairs as an additional parameter remains open. However, a subset of the present authors recently proved that Directed Multicut parameterized by both number of terminals and the size of the cutset is fixed-parameter tractable when restricted to directed acyclic input graphs [48].

Although the picture of the fixed-parameter tractability of graph separation problems becomes more and more complete, very little is known about polynomial kernelization. Very recently, Kratsch and Wahlström came up with an application of matroid theory to graph separation problems. They were able to obtain randomized polynomial kernels for Odd Cycle Transversal [49], Almost 2-SAT, and Multiway Cut and Multicut restricted to a bounded number of terminals, among others [50]. We are not aware of any other results on kernelization of graph separation problems.

We prove that Directed Multiway Cut, even in the case of two terminals, as well as Multicut, parameterized by the size of the cutset, are OR-compositional, thus a polynomial kernel for any of these two problems would cause a collapse of the polynomial hierarchy.

The k-way cut problem. The last part of this work is devoted to another generalization of the s-t cut problem, but with a slightly different flavor. The k-Way Cut problem is defined as follows: given an undirected graph G and integers k and s, remove at most s edges from G to obtain a graph with at least k connected components. This problem has applications in numerous areas of computer science, such as finding cutting planes for the traveling salesman problem, clustering-related settings (e.g., VLSI design), or network reliability [9]. In general, k-Way Cut is NP-complete [31] but solvable in polynomial time for fixed k: a long line of research [31, 42, 44, 65] led to a deterministic algorithm running in time $\mathcal{O}(mn^{2k-2})$. The dependency on k in the exponent is probably unavoidable: from the parameterized perspective, the k-Way Cut problem parameterized by k is $W[1]$-hard [24]. Moreover, the node-deletion variant is also $W[1]$-hard when parameterized by s [54]. Somewhat surprisingly, in 2011 Kawarabayashi and Thorup presented a fixed-parameter algorithm for (edge-deletion) k-Way Cut parameterized by s [45]. In this paper we complete the parameterized picture of the edge-deletion k-Way Cut problem parameterized by s by showing that it is OR-compositional and, therefore, a polynomial kernelization algorithm is unlikely to exist.

Organization of the paper. In Section 2 we give some basic notation and recall the necessary kernelization lower bound tools. In Sections 3 through 6 we prove the claimed kernelization lower bounds by giving cross-compositions to the target problems. In Section 3 we give an AND-cross-composition for the Edge Clique Cover problem. In Section 4 we provide an OR-cross-composition for Directed Multiway Cut. In Section 5 we provide an OR-cross-composition for Multicut. Section 6 gives an OR-cross-composition from Clique to k-Way Cut. We conclude in Section 7.

2 Preliminaries

Notation. We use standard graph notation. For a graph G, by $V(G)$ and $E(G)$ we denote its vertex and edge set (or arc set in case of directed graphs), respectively. For $v \in V(G)$, its neighborhood $N_G(v)$ is defined by $N_G(v) = \{u : uv \in E(G)\}$, and $N_G[v] = N_G(v) \cup \{v\}$ is the closed neighborhood of v. We extend this notation to subsets of vertices: $N_G[X] = \bigcup_{v \in X} N_G[v]$ and $N_G(X) = N_G[X] \setminus X$. For $X \subseteq V(G)$ by $\delta_G(X)$ we denote the set of edges in G with one endpoint in X and the other in $V(G) \setminus X$. For simplicity for a single vertex v we let $\delta(v) = \delta(\{v\})$. We omit the subscripts if no confusion is possible. For a set $X \subseteq V(G)$ by $G[X]$ we denote the subgraph of G induced by X. For a set X of vertices or edges of G, by $G \setminus X$ we denote the graph with the vertices or edges of X removed; in case of a vertex removal, we remove also all its incident edges. For sets $X, Y \subseteq V(G)$, the set $E(X, Y)$ contains all edges of G that have one endpoint in X and the second endpoint in Y. In particular, $E(X, X) = E(G[X])$ and $E(X, V(G) \setminus X) = \delta_G(X)$. For a (directed) graph G by an st-path we denote any path that starts in s and ends in t.

For two disjoint vertex sets S, T by an S–T cut we denote any set of edges, whose removal ensures that there is no path from a vertex in S to a vertex in T in the considered graph. By minimum S–T cut we denote an S–T cut of minimum cardinality.

Parameterized complexity. In the parameterized complexity setting, an instance comes with an integer parameter k — formally, a parameterized problem Q is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet Σ. We say that the problem is *fixed parameter tractable* (*FPT*) if there exists an algorithm solving any instance (x, k) in time $f(k)\text{poly}(|x|)$ for some (usually exponential) computable function f. It is known that a problem is FPT if and only if it is kernelizable: a kernelization algorithm for a problem Q takes an instance (x, k) and in time polynomial in $|x| + k$ produces an equivalent instance (x', k') (i.e., $(x, k) \in Q$ if and only if $(x', k') \in Q$) such that $|x'| + k' \leq g(k)$ for some computable function g. The function g is the *size of the kernel*, and if it is polynomial, we say that Q admits a polynomial kernel.

Kernelization lower bounds framework. We use the cross-composition technique introduced by Bodlaender et al. [**?**] which builds upon work of Bodlaender et al. [5], Fortnow and Santhanam [27], and Drucker [25].

Definition 1 (Polynomial equivalence relation [?])**.** An equivalence relation $\mathcal{R}$ on Σ^* is called a *polynomial equivalence relation* if (1) there is an algorithm that given two strings $x, y \in \Sigma^*$ decides whether $\mathcal{R}(x, y)$ in $(|x| + |y|)^{\mathcal{O}(1)}$ time; (2) for any finite set $S \subseteq \Sigma^*$ the equivalence relation $\mathcal{R}$ partitions the elements of S into at most $(\max_{x \in S} |x|)^{\mathcal{O}(1)}$ classes.

Definition 2 (AND/OR-cross-composition [?])**.** Let $L \subseteq \Sigma^*$ be a language, let $\mathcal{R}$ be a polynomial equivalence relation on Σ^*, and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. An *OR-cross-composition of L into Q* (with respect to $\mathcal{R}$) is an algorithm that, given t instances $x_1, x_2, \ldots, x_t \in \Sigma^*$ of L belonging to the same equivalence class of $\mathcal{R}$, takes time polynomial in $\sum_{i=1}^{t} |x_i|$ and outputs an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ such that:

"PB": The parameter value k is polynomially bounded in $\max_i |x_i| + \log t$.

"OR": The instance (y, k) is yes for Q if and only if *at least one* instance x_i is yes for L.

An *AND-cross-composition of L into Q* (with respect to $\mathcal{R}$) is an algorithm that, instead, fulfills Properties "PB" and "AND".

"AND": The instance (y, k) is yes for Q if and only if *all* instances x_i are yes for L.

We say that L OR-cross-composes, respectively AND-cross-composes, into Q if a cross-composition algorithm of the relevant type exists for a suitable relation $\mathcal{R}$.

Let us recall that OR-cross-compositions are commonly called cross-compositions, without the preposition, due to the prevalence of this type in the literature. It is known that both forms of cross-composition can be used to rule out polynomial kernelizations and compressions under the assumption that the polynomial hierarchy does not collapse.

Theorem 2 ([?], Corollary 1)**.** *If an NP-hard language L AND/OR-cross-composes into the parameterized problem Q, then Q does not admit a polynomial kernelization or polynomial compression unless NP $\subseteq$ coNP/poly and the polynomial hierarchy collapses.*

Observe that any polynomial equivalence relation is defined on all words over the alphabet Σ and for this reason whenever we define a cross-composition, we should also define how the relation behaves on words that do not represent instances of the problem. In all our constructions the defined relation puts all malformed instances into one equivalence class, and the corresponding cross-composition outputs a trivial NO-instance, given a sequence of malformed instances. Thus, in the rest of this paper, we silently ignore the existence of malformed instances.

3 Clique Cover

This section addresses the Edge Clique Cover problem parameterized by the maximum number k of cliques to be used in the cover. The problem is formally defined as follows.

Edge Clique Cover
Input: An undirected graph G and an integer k.
Task: Does there exist a set of k subgraphs of G, such that each subgraph is a clique and each edge of G is contained in at least one of these subgraphs?

We give an AND-cross-composition for Edge Clique Cover parameterized by k, thereby ruling out polynomial kernelizations and compressions.

Theorem 3. Edge Clique Cover *AND-cross-composes to* Edge Clique Cover *parameterized by k.*

Proof. For the equivalence relation $\mathcal{R}$ we take a relation that puts two instances (G_1, k_1), (G_2, k_2) of Edge Clique Cover in the same equivalence class if and only if $k_1 = k_2$ and the number of vertices in G_1 is equal to the number of vertices in G_2. Therefore, in the rest of the proof we assume that we are given a sequence $(G_i, k)_{i=0}^{t-1}$ of Edge Clique Cover instances that are in the same equivalence class of $\mathcal{R}$ (let us point out that in this proof we number everything starting from zero). Let n be the number of vertices in each of the instances. W.l.o.g. we assume that $n = 2^{h_n}$ for a positive integer h_n, since otherwise we may add isolated vertices to each instance. Moreover, we assume that $t = 2^{h_t}$ for some positive integer h_t, since we may copy some instance if needed, while increasing the number of instances at most by a factor two.

Now we construct an instance (G^*, k^*), where k^* is polynomial in $n + k + h_t$. Initially as G^* we take the disjoint union of the graphs G_i for $i = 0, \ldots, t-1$ with added edges between every pair of vertices from G_i and G_j for $i \neq j$. Next, in order to cover all the edges between different instances with few cliques we introduce the following construction. Let us assume that the vertex set of G_i is $V_i = \{v_0^i, \ldots, v_{n-1}^i\}$. For each $0 \leq a < n$, for each $0 \leq b < n$ and for each $0 \leq r < h_t$ we add to G^* a vertex $w(a, b, r)$ which is adjacent to exactly one vertex in each V_i, that is v_ℓ^i where $\ell = (a + b\lfloor \frac{i}{2^r} \rfloor) \bmod n$. By W we denote the set of all added vertices $w(a, b, r)$. As the new parameter k^* we set $k^* = |W| + k = n^2 h_t + k$. Note that W is an independent set in G^* and, moreover, each vertex in W is non-isolated.

Let us assume that for each $i = 0, \ldots, t-1$ the instance (G_i, k) is a YES-instance. To show that (G^*, k^*) is a YES-instance we create a set $\mathcal{C}$ of k^* cliques. We split all the edges of G^* into the following groups: (i) edges incident to vertices of W, (ii) edges between two different graphs G_i,G_j, and (iii) edges in each graph G_i. For each vertex $w \in W$ we add to $\mathcal{C}$ the subgraph $G^*[N[w]]$, which is a clique since every two vertices from two different graphs G_i, G_j are adjacent. Moreover, let $\mathcal{C}_i = \{C_0^i, \ldots, C_{k-1}^i\}$ be any solution for the instance (G_i, k). For each $\ell = 0, \ldots, k-1$ we add to $\mathcal{C}$ a clique $G^* \left[\bigcup_{i=0}^{t-1} C_\ell^i\right]$. Clearly all the edges mentioned in (i) and (iii) are covered. Consider any two vertices $v_x^i \in V_i$ and $v_y^j \in V_j$ for $i < j$. Let r be the greatest integer such that $(j - i)$ is divisible by 2^r. Note that $0 \leq r < h_t$ and $z := \lfloor \frac{j}{2^r} \rfloor - \lfloor \frac{i}{2^r} \rfloor \equiv 1 \pmod 2$ since otherwise $(j-i)$ would be divisible by 2^{r+1}. Consequently, there exists $0 \leq b < n$ satisfying the congruence $bz \equiv y - x \pmod n$, since the greatest common divisor of z and n is equal to one (recall that n is a power of 2). Therefore, when we set $a = y - b\lfloor \frac{j}{2^r} \rfloor$ we obtain

$$a + b\lfloor \frac{i}{2^r} \rfloor \equiv b(\lfloor \frac{i}{2^r} \rfloor - \lfloor \frac{j}{2^r} \rfloor) + y \equiv y - bz \equiv x \pmod n$$
$$a + b\lfloor \frac{j}{2^r} \rfloor \equiv y \pmod n$$

and both v_x^i, v_y^j belong to the clique of $\mathcal{C}$ containing the vertex $w(a, b, r)$.

Now let us assume that (G^*, k^*) is a YES-instance and let $\mathcal{C}$ be a set of at most k^* cliques in G^* that cover every edge in G^*. We define $\mathcal{C}' \subseteq \mathcal{C}$ as the set of these cliques in $\mathcal{C}$ which contain at least two vertices from some set V_i. Since W is an independent set in G^*, edges incident to two different vertices in W need to be covered by two different cliques in $\mathcal{C}$. Moreover, no clique in $\mathcal{C}'$

contains a vertex from W, because each vertex in W is incident to exactly one vertex in each V_i. Therefore, $|\mathcal{C}'| \leq |\mathcal{C}| - |W| \leq k$ and a set $\mathcal{C}_i = \{X \cap V_i : X \in \mathcal{C}'\}$ for $i = 0, \dots, t-1$ is a solution for (G_i, k), as no clique in $\mathcal{C} \setminus \mathcal{C}'$ covers an edge between two vertices in V_i for any $i = 0, \dots, t-1$. Hence each instance (G_i, k) is a YES-instance. □

As a consequence, by Theorem 2 we obtain the following result.

Corollary 1. *There is no polynomial kernel or compression for the* Edge Clique Cover *problem parameterized by k unless NP $\subseteq$ coNP/poly.*

4 Directed Multiway Cut

In the Directed Multiway Cut problem we want to disconnect every pair of terminals in a directed graph. The problem was previously studied in the following two versions.

> Directed Edge Multiway Cut
> **Input:** A directed graph $G = (V, A)$, a set of terminals $T \subseteq V$ and an integer p.
> **Task:** Does there exist a set S of at most p arcs in A, such that in $G \setminus S$ there is no path between any pair of terminals in T?

> Directed Vertex Multiway Cut
> **Input:** A directed graph $G = (V, A)$, a set of terminals $T \subseteq V$, a set of forbidden vertices $V^\infty \supseteq T$ and an integer p.
> **Task:** Does there exist a set S of at most p vertices in $V \setminus V^\infty$, such that in $G \setminus S$ there is no path between any pair of terminals in T?

As a side note, observe that by replacing each vertex of $V^\infty \setminus T$ with a $p+1$-clique (i.e., a graph on $p+1$ vertices pairwise connected by arcs in both directions), one can reduce the above Directed Vertex Multiway Cut version to a version, where the solution is allowed to remove any nonterminal vertex. Moreover, it is well known, that given an instance I of one of the two problems above, one can in polynomial time create an equivalent instance I' of the other problem, where both the number of terminals and the value of p remain unchanged (e.g. see [15]). Therefore we show a cross-composition to Directed Vertex Multiway Cut and as a corollary we prove that Directed Edge Multiway Cut also does not admit a polynomial kernel. The starting point is the following restricted variant of Directed Vertex Multiway Cut, which we prove to be NP-complete with respect to Karp reductions.

> Promised Directed Vertex 2-Multiway Cut
> **Input:** A directed graph $G = (V, A)$, two terminals $T = \{s_1, s_2\}$, a set of forbidden vertices $V^\infty \supseteq T$ and an integer p. Moreover, after removing any set of at most $p/2$ vertices of $V \setminus V^\infty$, both an $s_1 s_2$-path and an $s_2 s_1$-path remain.
> **Task:** Does there exist a set S of at most p vertices in $V \setminus V^\infty$, such that in $G \setminus S$ there is no $s_1 s_2$-path nor $s_2 s_1$-path?

The assumption that any set of size at most $p/2$ cannot hit all the paths from s_1 to s_2 (and similarly from s_2 to s_1) will help us in constructing a cross-composition. Note that this property can be efficiently verified for any input by computing a min s_1–s_2 cut and a min s_2–s_1 cut and checking that both cuts have size exceeding $p/2$. It follows that the problem is contained in NP.

Lemma 1. Promised Directed Vertex 2-Multiway Cut *is NP-complete with respect to Karp reductions.*

Proof. To prove that the problem is NP-hard we use the NP-completeness result of Garg et al. [30] for Directed Vertex Multiway Cut with two terminals. Consider an instance $I = (G, T = \{s_1, s_2\}, V^\infty, p)$ of Directed Vertex Multiway Cut. As the graph G' we take G with $z = p + 1$ vertices $\{u_1, \ldots, u_z\}$ added. In G' for $i = 1, \ldots, z$ we add the following four arcs $\{(s_1, u_i), (u_i, s_1), (u_i, s_2), (s_2, u_i)\}$. Let $I' = (G', T, V^\infty, p + z)$ be an instance of Promised Directed Vertex 2-Multiway Cut. Since after removal of less than z vertices in G' at least one vertex u_i remains, we infer that I' is indeed a Promised Directed Vertex 2-Multiway Cut instance. To prove that I is a YES-instance if and only if I' is a YES-instance it is enough to observe that any solution in I' contains all the vertices $\{u_1, \ldots, u_z\}$. □

Equipped with the Promised Directed Vertex 2-Multiway Cut problem definition, we are ready to show a cross-composition into Directed Vertex Multiway Cut parameterized by p.

Theorem 4. Promised Directed Vertex 2-Multiway Cut *cross-composes into* Directed Vertex Multiway Cut *with two terminals, parameterized by the size of the cutset p.*

Proof. For the equivalence relation $\mathcal{R}$, we take a relation that groups instances according to the value of p, i.e., $(G_i, T_i, V_i^\infty, p_i)$ and $(G_j, T_j, V_j^\infty, p_j)$ are in the same equivalence class in $\mathcal{R}$ if and only if $p_i = p_j$. We assume that we are given a sequence $I_i = (G_i, T_i = \{s_1^i, s_2^i\}, V_i^\infty, p)_{i=1}^t$ of Promised Directed Vertex 2-Multiway Cut instances that are in the same equivalence class of $\mathcal{R}$. As the graph G' we take the disjoint union of all the graphs G_i. Moreover for each $i = 1, \ldots, t-1$, in G' we identify the vertices s_2^i and s_1^{i+1}. Let $I' = (G', \{s_1^1, s_2^t\}, \bigcup_{i=1}^t V_i^\infty, p)$ be an instance of Directed Vertex Multiway Cut. Note that $\bigcup_{i=1}^t V_i^\infty$ contains both terminals from all input instances.

Let us assume that there exists $1 \leq i_0 \leq t$ such that I_{i_0} is a YES-instance of Promised Directed Vertex 2-Multiway Cut, and let $S \subseteq V(G_i) \setminus V_i^\infty$ be any solution for I_{i_0}. Since any $s_1^1 s_2^t$-path and any $s_2^t s_1^1$-path in G' goes through both $s_1^{i_0}$ and $s_2^{i_0}$, we observe that $G' \setminus S$ is a solution for I' and, consequently, I' is a YES-instance.

In the other direction, let us assume that I' is a YES-instance. Let $S \subseteq V(G) \setminus \bigcup_{i=1}^t V_i^\infty$ be any solution for I'. Observe that if the set S contains at most $p/2$ vertices of $V(G_i) \setminus V_i^\infty$ for some $1 \leq i \leq t$, then $S \setminus V(G_i)$ is also a solution for I', since after removing at most $p/2$ vertices of $V(G_i)$ there is still a path both from s_1^i to s_2^i and from s_2^i to s_1^i. Because $|S| \leq p$, we infer that w.l.o.g. S contains only vertices of a single set $V(G_{i_0})$ for some $1 \leq i_0 \leq t$. Therefore, I_{i_0} is a YES-instance. □

The equivalence of Directed Vertex Multiway Cut and Directed Edge Multiway Cut together with Theorem 2 give us the following corollary.

Corollary 2. Directed Vertex Multiway Cut *and* Directed Edge Multiway Cut *do not admit polynomial kernels or compressions when parameterized by p unless NP $\subseteq$ coNP/poly, even in the case of two terminals.*

5 Multicut

In this section we prove that both the edge and vertex versions of the MULTICUT problem do not admit a polynomial kernel, when parameterized by the size of the cutset.

> EDGE (VERTEX) MULTICUT
> **Input:** An undirected graph $G = (V, E)$, a set of pairs of terminals $\mathcal{T} = \{(s_1, t_1), \ldots, (s_k, t_k)\}$ and an integer p.
> **Task:** Does there exist a set $S \subseteq E$ ($S \subseteq V$) such that no connected component of $G \setminus S$ contains both vertices s_i and t_i, for some $1 \leq i \leq k$?

It is known that the vertex version of the MULTICUT problem is at least as hard as the edge version.

Lemma 2 (folklore)**.** *There is a polynomial time algorithm, which given an instance $I = (G, \mathcal{T}, p)$ of* EDGE MULTICUT *produces an instance $I' = (G', \mathcal{T}', p)$ of* VERTEX MULTICUT*, such that I is a YES-instance iff I' is a YES-instance.*

In order to show a cross-composition into the MULTICUT problem parameterized by p we consider the following restricted variant of the MULTIWAY CUT problem with three terminals.

> 3-MULTIWAY CUT
> **Input:** An undirected graph $G = (V, E)$, a set of three terminals $T = \{s_1, s_2, s_3\} \subseteq V$ and an integer p.
> **Task:** Does there exist a set S of at most p edges in E, such that in $G \setminus S$ there is no path between any pair of terminals in T?

> PROMISED 3-MULTIWAY CUT
> **Input:** An undirected graph $G = (V, E)$, a set of three terminals $T = \{s_1, s_2, s_3\} \subseteq V$ and an integer p. An instance satisfies: (i) $\deg(s_1) = \deg(s_2) = \deg(s_3) = d > 0$, (ii) for each $j = 1, 2, 3$ and any non-empty set $X \subseteq V \setminus T$ we have $|\delta(X \cup \{s_j\})| > d$, and (iii) $d \leq p < 2d$.
> **Task:** Does there exist a set S of at most p edges in E, such that in $G \setminus S$ there is no path between any pair of terminals in T?

Condition (i) ensures that degrees of all the terminals are equal, whereas condition (ii) guarantees that the set of edges incident to a terminal s_j is the only minimum size s_j–$(T \setminus \{s_j\})$ cut. Having both (i) and (ii), condition (iii) verifies whether an instance is not a trivially YES- or NO-instance, because by (i) and (ii) there is no solution of size less than d and removing all the edges incident to two terminals always gives a solution of size at most $2d$.

Lemma 3. PROMISED 3-MULTIWAY CUT *is NP-complete with respect to Karp reductions.*

Proof. To prove the lemma we may observe that the first NP-hardness reduction to the MULTIWAY CUT problem by Dahlhaus et al. [20] in fact yields a PROMISED 3-MULTIWAY CUT instance. For sake of completeness, we present here how to reduce an arbitrary instance of the MULTIWAY CUT problem with three terminals to a PROMISED 3-MULTIWAY CUT instance.

Let $I = (G, T = \{s_1, s_2, s_3\}, p)$ be an instance of 3-MULTIWAY CUT. As observed by Marx [54], we can assume that for each terminal s_i the cut $\delta(s_i)$ is the only minimum cardinality s_i–$(T \setminus$

$\{s_i\}$) cut, since otherwise w.l.o.g. we may contract some edge incident to s_i obtaining a smaller equivalent instance. Therefore condition (ii) would be satisfied if only degrees of terminals were equal. Let G_1, G_2, G_3 be three copies of the graph G, where terminals in the i-th copy are denoted by $T_i = \{s_1^i, s_2^i, s_3^i\}$. Construct a graph G' as a disjoint union of G_1, G_2 and G_3. Next in G' we identify vertices $\{s_1^1, s_2^2, s_3^3\}$ into a single vertex s_1', similarly identify vertices $\{s_2^1, s_3^2, s_1^3\}$ into a single vertex s_2', and finally identify vertices $\{s_3^1, s_1^2, s_1^3\}$ into a single vertex s_3'. Let $I' = (G', T' = \{s_1', s_2', s_3'\}, p' = 3p\}$. Observe that due to the performed identification I' is a YES-instance of Multiway Cut if and only if I is a YES-instance of Multiway Cut. Therefore, to finish the reduction it suffices to argue that I' satisfies (i), (ii) and (iii).

Let $d = \sum_{i=1}^{3} \deg_G(s_i)$. Note that in G' for each $i = 1, 2, 3$ we have $\deg_{G'}(s_i') = d$, hence condition (i) is satisfied. Observe that if there exists $1 \le j \le 3$ and s_j'–$(T' \setminus \{s_j'\})$ cut in G' of size at most d, which is different from $\delta(s_j')$, then there exists $1 \le r \le 3$ and s_r–$(T \setminus \{s_r\})$ cut in G of size at most $\deg_G(s_r)$ which is different from $\delta_G(s_r)$, a contradiction. Hence condition (ii) is satisfied. Unfortunately, it is possible that $p' \le d$ or $p' \ge 2d$. However, if $p' \ge 2d$ then clearly I' is a YES-instance (we can remove edges incident to two terminals), and hence I is a YES-instance. On the other hand if $p' < d$, then I' (and consequently I) is a NO-instance, since any s_1'–$\{s_2', s_3'\}$ cut has size at least d. Therefore, if condition (iii) is not satisfied, then in polynomial time we can compute the answer for the instance I, and as the instance I' we set a trivial YES- or NO-instance. □

Theorem 5. Promised 3-Multiway Cut *cross-composes into* Edge Multicut *parameterized the size of the cutset* p.

Proof. For the equivalence relation $\mathcal{R}$, we take a relation that groups instances according to the values of p and d, i.e., (G_i, T_i, p_i) and (G_j, T_j, p_j) are in the same equivalence class in $\mathcal{R}$ if and only if $p_i = p_j$ and the degree of each terminal in G_i equals the degree of each terminal in G_j. We assume that we are given a sequence $I_i = (G_i, T_i = \{s_1^i, s_2^i, s_3^i\}, p)_{i=0}^{t-1}$ of Promised 3-Multiway Cut instances that are in the same equivalence class of $\mathcal{R}$ (note that we number instances starting from 0). Let d be the degree of each terminal in each of the instances. W.l.o.g. we assume that $t \ge 5$ is an odd integer, since we may copy some instances if needed, and let $h = (t-1)/2$.

Construction. Let G' be the disjoint union of all graphs G_i for $i = 0, \ldots, t-1$. For each $i = 0, \ldots, t-1$ we add d parallel edges between vertices s_2^i and $s_1^{(i+1) \bmod t}$. To the set $\mathcal{T}$ we add exactly t pairs, that is for each $i = 0, \ldots, t-1$ we add to $\mathcal{T}$ the pair $(s_i' = s_3^i, t_i' = s_3^{(i+h) \bmod t})$. We set $p' = p + d$ and $I' = (G', \mathcal{T}, p')$ is the constructed Edge Multicut instance. Note that in order to avoid using parallel edges it is enough to subdivide them.

Analysis. First assume that there exists $0 \le i_0 < t$ such that I_{i_0} is a YES-instance of Promised 3-Multiway Cut and let $S \subseteq E(G_{i_0})$ be any solution for I_{i_0}. Let S_1 be the set of edges in G' between $s_2^{(i_0+h) \bmod t}$ and $s_1^{(i_0+h+1) \bmod t}$ (see Fig. 1). We prove that $S' = S \cup S_1$ is a solution for I'. Observe that $|S'| = |S| + |S_1| \le p + d$. Consider any pair $(s', t') \in \mathcal{T}$ such that $s', t' \ne s_3^{i_0}$. Note that in $G' \setminus S'$ there is neither an $s_1^{i_0} s_2^{i_0}$-path, nor an $s_2^{(i_0+h) \bmod t} s_1^{(i_0+h+1) \bmod t}$-path. Therefore, there is no $s't'$-path in $G' \setminus S'$. Moreover, in $G' \setminus S'$ there is neither an $s_3^{i_0} s_1^{i_0}$-path, nor an $s_3^{i_0} s_2^{i_0}$-path. Consequently, for each $(s', t') \in \mathcal{T}$, where $s' = s_3^{i_0}$ or $t' = s_3^{i_0}$, there is no $s't'$-path in $G' \setminus S'$, so I' is a YES-instance of Edge Multicut.

Now assume that I' is a YES-instance and our goal is to show that for some $0 \le i < t$ the instance I_i is a YES-instance. Let $E_i = E(G_i)$ and let E_i' be the set of edges between s_2^i

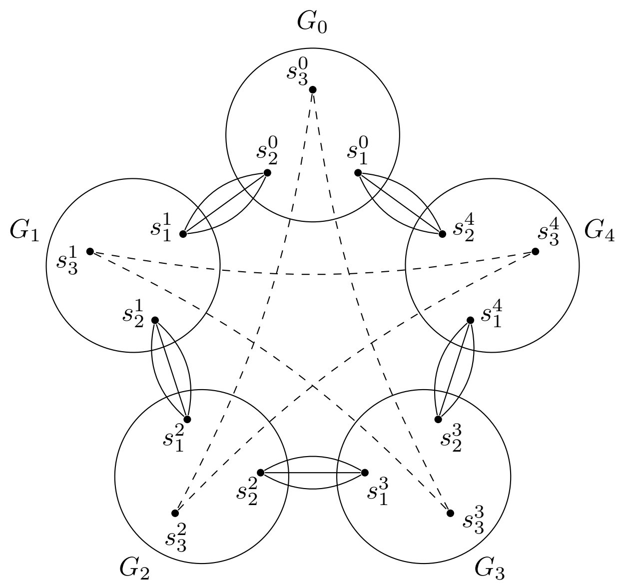

Figure 1: Construction of the graph G' for $t = 5$ and $d = 3$. Dashed edges represent pairs of vertices in $\mathcal{T}$.

and $s_1^{(i+1) \bmod t}$ in G'. Let $S' \subseteq E(G')$ be any solution for I'. Note that if for some E'_i, where $0 \leq i < t$, the set S' contains less than d edges from the set E'_i, then $S' \setminus E'_i$ is also a solution for I'. By conditions (i) and (ii) of the Promised 3-Multiway Cut problem definition we have the following: if for some $0 \leq i < t$ the set S' contains less than d edges from the set E_i, then $S' \setminus E_i$ is also a solution for I'. Indeed, if S' contains less than d edges from E_i, then in the graph $G' \setminus E_i$ all the vertices s_1^i, s_2^i, s_3^i are in the same connected component, since otherwise for some $a \in T_i$ there would be an a–$(T_i \setminus \{a\})$ cut in G_i of size smaller than d. Let us recall that $|S'| \leq p' = p + d < 3d$. Therefore, w.l.o.g. we may assume that the set S' has a non-empty intersection with at most two sets from the set $\mathcal{E} = \{E_0, \ldots, E_{t-1}, E'_0, \ldots, E'_{t-1}\}$. Moreover we assume that if S' has non-empty intersection with some set from $\mathcal{E}$, then this intersection is of size at least d.

Case 1. Consider the case, when S' has an empty intersection with each of the sets E_i for $0 \leq i < t$. Since $|E'_i| = d$ and $p' \geq 2d$, w.l.o.g. S' has a non-empty intersection with exactly two sets E'_{i_0}, E'_{i_1} for $i_0 \neq i_1$. Since t is odd, in the graph $G' \setminus S'$ either there is an $s_3^{i_0} s_3^{(i_0-h) \pmod t}$-path or an $s_3^{(i_0+1) \bmod t} s_3^{(i_0+1+h) \bmod t}$-path. Hence we have a contradiction.

Case 2. Next assume that S' has a non-empty intersection with some set E_{i_0} for $0 \leq i_0 < t$. By symmetry w.l.o.g. we may assume that $i_0 = 0$. Since the set S' hits all the $s_3^1 s_3^{h+1}$-paths as well as all the $s_3^h s_3^{t-1}$-paths in the graph G', we infer that S' has non-empty intersection with exactly one of the sets E_h, E'_h, E_{h+1}.

Case 2.1. In this case we assume that S' has a non-empty intersection with E'_h. Since S' hits all $s_3^0 s_3^h$-paths in G', in $G_0 \setminus S'$ there is no $s_3^0 s_2^0$-path. Similarly, since S' hits all $s_3^0 s_3^{h+1}$-paths in G', in $G_0 \setminus S'$ there is no $s_3^0 s_1^0$-path. Moreover, since S' hits all $s_3^{t-1} s_3^{h-1}$-paths in G', in $G_0 \setminus S'$ there is no $s_1^0 s_2^0$-path. Since $|S'| \leq p' = p + d$ and $|S' \cap E'_h| = d$, we infer that $|S' \cap E_0| \leq p$, and, consequently, I_0 is a YES-instance.

Case 2.2. Since S' has a non-empty intersection with one of the sets E_h, E_{h+1}, by symmetry we assume that $S' \cap E_h \neq \emptyset$. Recall that $t \geq 5$, and hence $h > 1$. Since S' hits all $s_3^1 s_3^{h+1}$-paths and all $s_3^1 s_3^{t+1-h}$-paths in G', we infer that in $G_0 \setminus S'$ there is no $s_1^0 s_2^0$-path and in $G_h \setminus S'$ there is no $s_1^h s_2^h$-path. Moreover, S' hits all $s_3^0 s_3^{h+1}$-paths and all $s_3^h s_3^{t-1}$-paths in G'; therefore, in $G_0 \setminus S'$ there is no $s_3^0 s_1^0$-path and in $G_h \setminus S'$ there is no $s_3^h s_2^h$-path. Finally since S' hits all $s_3^0 s_3^h$-paths in G', either in $G_0 \setminus S'$ there is no $s_3^0 s_2^0$-path, or in $G_h \setminus S'$ there is no $s_3^h s_1^h$-path. Since $|S'| \leq p + d$, $|S' \cap E_0| \geq d$ and $|S' \cap E_h| \geq d$, we infer that $|S' \cap E_0| \leq p$ and $|S' \cap E_h| \leq p$. Consequently, either I_0 or I_h is a YES-instance, which finishes the proof of Theorem 5. □

6 k-Way Cut

In this section we study the k-Way Cut problem, defined as follows.

> k-Way Cut
> **Input:** An undirected connected graph G and integers k and s.
> **Task:** Does there exist a set X of at most s edges in G such that $G \setminus X$ has at least k connected components?

The k-Way Cut problem, parameterized by s, was proven to be fixed-parameter tractable by Kawarabayashi and Thorup [45]. The problem is $W[1]$-hard when parameterized by k [24], as well as when we allow vertex deletions instead of edge deletions, and parameterize by s [54].

Note that in the problem definition we assume that the input graph is connected and, therefore, for $k > s + 1$ the input instances are trivial. However, if we are given an instance (G, k, s) where G has $c > 1$ connected components, we can easily reduce it to the connected version: we add to G a complete graph on $s + 2$ vertices (so that no two vertices of the complete graph can be separated by a cut of size s), connect one vertex from each connected component of G to all vertices of the complete graph, and decrease k by $c - 1$. Thus, by restricting ourselves to connected graphs G we do not make the problem easier.

The main result of this section is that k-Way Cut, parameterized by s, does not admit a polynomial kernel (unless NP $\subseteq$ coNP/poly). We show a cross-composition from the Clique problem, well-known to be NP-complete.

> Clique
> **Input:** An undirected graph G and an integer ℓ.
> **Task:** Does G contain a clique on ℓ vertices as a subgraph?

Theorem 6. Clique *cross-composes to* k-Way Cut *parameterized by* s.

Proof. We start by defining a relation $\mathcal{R}$ on Clique input instances as follows: (G, ℓ) is in relation $\mathcal{R}$ with (G', ℓ') if $\ell = \ell'$, $|V(G)| = |V(G')|$, and $|E(G)| = |E(G')|$. Clearly, $\mathcal{R}$ is a polynomial equivalence relation. Thus, in the designed cross-composition, we may assume that we are given t instances (G_i, ℓ), $1 \leq i \leq t$, of the Clique problem and $|V(G_i)| = n$, $|E(G_i)| = m$ for all $1 \leq i \leq t$. Moreover, we assume that $m \geq \binom{\ell}{2}$ and $1 < \ell \leq n$, as otherwise all input instances are trivial.

We first consider a weighted version of the k-Way Cut problem where each edge may have a positive integer weight and the cutset X needs to be of total weight at most s. The weights in our construction are polynomial in n and m. At the end we show how to reduce the weighted version to the unweighted one.

We start by defining the following integer values

$$k = n - \ell + 1, \qquad w_1 = m, \qquad w_2 = m\binom{n}{2},$$

$$s = w_2(n-\ell) + w_1\left(\binom{n}{2} - \binom{\ell}{2}\right) + m - \binom{\ell}{2}.$$

Note that $s < w_2(n-\ell+1)$ and $s < w_2(n-\ell) + w_1(\binom{n}{2} - \binom{\ell}{2} + 1)$.

For each graph G_i, $1 \leq i \leq t$, we define a graph G_i' as a complete graph on n vertices with vertex set $V(G_i)$, where the edge uv has weight $w_1 + 1$ if $uv \in E(G_i)$ and weight w_1 otherwise. We construct a graph G as follows. We take the disjoint union of all graphs G_i' for $1 \leq i \leq t$, add a root vertex r, and for each $1 \leq i \leq t$, $v \in V(G_i')$ we add an edge rv of weight w_2. Clearly G is connected, s is polynomial in n and m, and the graph G can be constructed in polynomial time. We claim that (G, k, s) is a weighted k-Way Cut YES-instance if and only if at least one of the input Clique instances (G_i, ℓ) is a YES-instance.

First, assume that for some $1 \leq i \leq t$, the Clique instance (G_i, ℓ) is a YES-instance. Let $C \subseteq V(G_i)$ be a witness: $|C| = \ell$ and $G_i[C]$ is a clique. Consider a set $X \subseteq E(G)$ containing all edges of G incident to $V(G_i') \setminus C$. Clearly, $G \setminus X$ contains $k = n - \ell + 1$ connected components: we have one large connected component with vertex set $(V(G) \setminus V(G_i')) \cup C$ and each of $n - \ell$ vertices of $V(G_i') \setminus C$ is an isolated vertex in $G \setminus X$. Let us now count the total weight of edges in X. The set X contains $n - \ell$ edges of weight w_2 that connect $V(G_i') \setminus C$ to the root r. Moreover, X contains $\binom{n}{2} - \binom{\ell}{2}$ edges of G_i', of weight w_1 or $w_1 + 1$. Since $G_i[C]$ is a clique, only $m - \binom{\ell}{2}$ of the edges in X are of weight $w_1 + 1$. Thus the total weight of edges in X is equal to

$$w_2(n-\ell) + w_1\left(\binom{n}{2} - \binom{\ell}{2}\right) + m - \binom{\ell}{2} = s.$$

In the other direction, let $X \subseteq E(G)$ be a solution to the k-Way Cut instance (G, k, s). Let Z be the connected component of $G \setminus X$ that contains the root r. Let $Y \subseteq V(G)$ be the set of vertices that are not in Z. If $v \in Y$, then X contains the edge rv of weight w_2. As $s < w_2(n - \ell + 1)$, we have $|Y| \leq n - \ell$. As $k = n - \ell + 1$, we infer that $G \setminus X$ contains $n - \ell + 1$ connected components: Z and $n - \ell$ isolated vertices. That is, $|Y| = n - \ell$ and all vertices in Y are isolated in $G \setminus X$. Note that X must include all $n - \ell$ edges of weight w_2 that connect the root r with the vertices of Y.

The next step is to prove that all vertices of Y are contained in one of the graphs G_i'. To this end, let $a_i = |Y \cap V(G_i')|$ for $1 \leq i \leq t$. Note that $X \cap E(G_i')$ contains at least $\binom{a_i}{2} + a_i(n - a_i)$ edges of weight w_1 or $w_1 + 1$. Thus, the number of edges of weight w_1 or $w_1 + 1$ contained in X is at least:

$$\begin{aligned}
\sum_{i=1}^{t}\left(\binom{a_i}{2} + a_i(n-a_i)\right) &= \left(n - \frac{1}{2}\right)\sum_{i=1}^{t} a_i - \frac{1}{2}\sum_{i=1}^{t} a_i^2 = (n-\ell)\left(n - \frac{1}{2}\right) - \frac{1}{2}\sum_{i=1}^{t} a_i^2 \\
&\geq (n-\ell)\left(n - \frac{1}{2}\right) - \frac{1}{2}\left(\sum_{i=1}^{t} a_i\right)^2 \\
&= (n-\ell)\left(n - \frac{1}{2}\right) - \frac{1}{2}(n-\ell)^2 \\
&= \binom{n}{2} - \binom{\ell}{2}.
\end{aligned}$$

As $s < w_2(n-\ell)+w_1\left(\binom{n}{2}-\binom{\ell}{2}+1\right)$, we infer that the number of edges in X of weight w_1 or w_1+1 is exactly $\binom{n}{2}-\binom{\ell}{2}$. This is only possible if $\sum_{i=1}^t a_i^2 = (\sum_{i=1}^t a_i)^2$. As a_i are nonnegative integers, we infer that only one value a_i is positive.

Thus $Y \subseteq V(G_i')$ for some $1 \leq i \leq t$. Let $C = V(G_i) \setminus Y$. Note that $|C| = \ell$. The set X contains all $\binom{n}{2}-\binom{\ell}{2}$ edges of G_i' that are incident to Y. As the total weight of the edges of X is at most s, X contains at most $m - \binom{\ell}{2}$ edges of weight w_1+1. We infer that there are at least $\binom{\ell}{2}$ edges in the graph $G_i[C]$, that $G_i[C]$ is a clique, and that (G_i, ℓ) is a YES-instance of the Clique problem.

To finish the proof, we show how to reduce the weighted version of the k-Way Cut problem to the unweighted one. We replace each vertex u with a complete graph H_u on $s+2$ vertices and for each edge uv of weight w we add to the graph w arbitrarily chosen edges between H_u and H_v (note that in our construction all weights are smaller than s). Note that this reduction preserves the connectivity of the graph G. Let X be a solution to the unweighted instance (G, k, s) constructed in this way. As no cut of size at most s can separate two vertices of H_u, each clique H_u is contained in one connected component of $G \setminus X$. Moreover, to separate H_u from H_v, the set X needs to include all w edges between H_u and H_w. Thus, the constructed unweighted instance is indeed equivalent to the weighted one. Note that in the presented cross-composition the edge weights were polynomial in n and m, so the presented reduction can be performed in polynomial time. □

By applying Theorem 2 we obtain the following corollary.

Corollary 3. *k-Way Cut parameterized by s does not admit a polynomial kernel or compression unless NP $\subseteq$ coNP/poly.*

7 Conclusion and open problems

We have shown that four important parameterized problems do not admit a kernelization algorithm with a polynomial guarantee on the output size unless NP $\subseteq$ coNP/poly and the polynomial hierarchy collapses. We would like to mention here some open problems very closely related to our work.

- The OR-composition for Directed Multiway Cut in the case of two terminals excludes the existence of a polynomial kernel for most graph separation problems in directed graphs. There are two important cases not covered by this result: one is the Multicut problem in directed acyclic graphs, and the second is a variant of Directed Multiway Cut where the cutset is allowed to contain terminals.
- Both our OR-compositions for Multicut use a number of terminal pairs that is linear in the number of input instances. Is Multicut parameterized by both the size of the cutset and the number of terminal pairs similarly hard to kernelize?

Acknowledgements. We would like to thank Jakub Onufry Wojtaszczyk for some early discussions on the kernelization of the graph separation problems.

References

[1] Pankaj K. Agarwal, Noga Alon, Boris Aronov, and Subhash Suri. Can visibility graphs be represented compactly? *Discrete & Computational Geometry*, 12:347–365, 1994.

[2] Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.

[3] Michael Behrisch and Anusch Taraz. Efficiently covering complex networks with cliques of similar vertices. *Theor. Comput. Sci.*, 355(1):37–47, 2006.

[4] Hans L. Bodlaender. Kernelization: New upper and lower bound techniques. In *IWPEC*, volume 5917 of *Lecture Notes in Computer Science*, pages 17–37. Springer, 2009.

[5] Hans L. Bodlaender, Rodney G. Downey, Michael R. Fellows, and Danny Hermelin. On problems without polynomial kernels. *J. Comput. Syst. Sci.*, 75(8):423–434, 2009.

[6] Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Preprocessing for treewidth: A combinatorial analysis through kernelization. *SIAM J. Discrete Math.*, 27(4):2108–2142, 2013.

[7] Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theor. Comput. Sci.*, 412(35):4570–4578, 2011.

[8] Nicolas Bousquet, Jean Daligault, and Stéphan Thomassé. Multicut is FPT. In *STOC*, pages 459–468, 2011.

[9] Michel Burlet and Olivier Goldschmidt. A new and improved algorithm for the 3-cut problem. *Oper. Res. Lett.*, 21(5):225–227, 1997.

[10] Jin-yi Cai, Venkatesan T. Chakaravarthy, Lane A. Hemaspaandra, and Mitsunori Ogihara. Competing provers yield improved karp-lipton collapse results. *Inf. Comput.*, 198(1):1–23, 2005.

[11] Gruia Calinescu, Howard J. Karloff, and Yuval Rabani. An improved approximation algorithm for multiway cut. *J. Comput. Syst. Sci.*, 60(3):564–574, 2000.

[12] Maw-Shang Chang and Haiko Müller. On the tree-degree of graphs. In *WG*, volume 2204 of *Lecture Notes in Computer Science*, pages 44–54. Springer, 2001.

[13] Jianer Chen, Yang Liu, and Songjian Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.

[14] Jianer Chen, Yang Liu, Songjian Lu, Barry O'Sullivan, and Igor Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.

[15] Rajesh Hemant Chitnis, MohammadTaghi Hajiaghayi, and Dániel Marx. Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset. *SIAM J. Comput.*, 42(4):1674–1696, 2013.

[16] Marek Cygan, Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Clique cover and graph separation: New incompressibility results. In *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 254–265. Springer, 2012.

[17] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Kernelization hardness of connectivity problems in d-degenerate graphs. *Discrete Applied Mathematics*, 160(15):2131–2141, 2012.

[18] Marek Cygan, Marcin Pilipczuk, Michał Pilipczuk, and Jakub Onufry Wojtaszczyk. On multiway cut parameterized above lower bounds. *TOCT*, 5(1):3, 2013.

[19] Marek Cygan, Marcin Pilipczuk, Michal Pilipczuk, and Jakub Onufry Wojtaszczyk. Subset feedback vertex set is fixed-parameter tractable. *SIAM J. Discrete Math.*, 27(1):290–309, 2013.

[20] Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.

[21] Holger Dell and Dániel Marx. Kernelization of packing problems. In *SODA*, pages 68–81, 2012.

[22] Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *STOC*, pages 251–260. ACM, 2010.

[23] Michael Dom, Daniel Lokshtanov, and Saket Saurabh. Incompressibility through colors and IDs. In *ICALP (1)*, volume 5555 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2009.

[24] Rodney G. Downey, Vladimir Estivill-Castro, Michael R. Fellows, Elena Prieto, and Frances A. Rosamond. Cutting up is hard to do: the parameterized complexity of k-cut and related problems. *Electr. Notes Theor. Comput. Sci.*, 78:209–222, 2003.

[25] Andrew Drucker. New limits to classical and quantum instance compression. In *FOCS*, pages 609–618. IEEE Computer Society, 2012.

[26] Paul Erdös, A. W. Goodman, and Lajos Posa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112, 1966.

[27] Lance Fortnow and Rahul Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *J. Comput. Syst. Sci.*, 77(1):91–106, 2011.

[28] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman, 1979.

[29] 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.

[30] Naveen Garg, Vijay V. Vazirani, and Mihalis Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.

[31] O. Goldschmidt and D. S. Hochbaum. A polynomial algorithm for the k-cut problem for fixed k. *Math. Oper. Res.*, 19(1):24–37, 1994.

[32] Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Data reduction and exact algorithms for clique cover. *ACM Journal of Experimental Algorithmics*, 13, 2008.

[33] Jens Gramm, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Hans-Peter Piepho, and Ramona Schmid. Algorithms for compact letter displays: Comparison and evaluation. *Computational Statistics & Data Analysis*, 52(2):725–736, 2007.

[34] Jonathan L. Gross and Jay Yellen. *Graph Theory and its Applications*. CRC Press, 2006.

[35] Jean-Loup Guillaume and Matthieu Latapy. Bipartite structure of all complex networks. *Inf. Process. Lett.*, 90(5):215–221, 2004.

[36] Sylvain Guillemot. FPT algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61–71, 2011.

[37] Jiong Guo and Rolf Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[38] Danny Harnik and Moni Naor. On the compressibility of *NP* instances and cryptographic applications. *SIAM J. Comput.*, 39(5):1667–1713, 2010.

[39] Danny Hermelin and Xi Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In *SODA*, pages 104–113, 2012.

[40] Douglas N. Hoover. Complexity of graph covering problems for graphs of low degree. *Journal of Combinatorial Mathematics and Combinatorial Computing*, 11:187–208, 1992.

[41] Wen-Lian Hsu and Kuo-Hui Tsai. Linear time algorithms on circular-arc graphs. *Inf. Process. Lett.*, 40(3):123–129, 1991.

[42] Yoko Kamidoi, Noriyoshi Yoshida, and Hiroshi Nagamochi. A deterministic algorithm for finding all minimum k-way cuts. *SIAM J. Comput.*, 36(5):1329–1341, 2007.

[43] David R. Karger, Philip N. Klein, Clifford Stein, Mikkel Thorup, and Neal E. Young. Rounding algorithms for a geometric embedding of minimum multiway cut. *Math. Oper. Res.*, 29(3):436–461, 2004.

[44] David R. Karger and Clifford Stein. A new approach to the minimum cut problem. *J. ACM*, 43(4):601–640, 1996.

[45] Ken-ichi Kawarabayashi and Mikkel Thorup. The minimum k-way cut of bounded size is fixed-parameter tractable. In *FOCS*, pages 160–169, 2011.

[46] Eduardo Kellerman. Determination of keyword conflict. *IBM Technical Disclosure Bulletin*, 16(2):544–546, 1973.

[47] Lawrence T. Kou, Larry J. Stockmeyer, and Chak-Kuen Wong. Covering edges by cliques with regard to keyword conflicts and intersection graphs. *Communications of the ACM*, 21(2):135–139, 1978.

[48] Stefan Kratsch, Marcin Pilipczuk, Michal Pilipczuk, and Magnus Wahlström. Fixed-parameter tractability of multicut in directed acyclic graphs. In *ICALP (1)*, volume 7391 of *Lecture Notes in Computer Science*, pages 581–593. Springer, 2012.

[49] Stefan Kratsch and Magnus Wahlström. Compression via matroids: a randomized polynomial kernel for odd cycle transversal. In *SODA*, pages 94–103, 2012.

[50] Stefan Kratsch and Magnus Wahlström. Representative sets and irrelevant vertices: New tools for kernelization. In *FOCS*, pages 450–459. IEEE Computer Society, 2012.

[51] Daniel Lokshtanov, Neeldhara Misra, and Saket Saurabh. Kernelization – preprocessing with a guarantee. In Hans Bodlaender, Rod Downey, Fedor Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *Lecture Notes in Computer Science*, pages 129–161. Springer Berlin / Heidelberg, 2012. 10.1007/978-3-642-30891-8_10.

[52] Carsten Lund and Mihalis Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[53] Shaohan Ma, Walter D. Wallis, and Julin Wu. Clique covering of chordal graphs. *Utilitas Mathematica*, 36:151–152, 1989.

[54] Dániel Marx. Parameterized graph separation problems. *Theor. Comput. Sci.*, 351(3):394–406, 2006.

[55] Dániel Marx and Igor Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. In *STOC*, pages 469–478. ACM, 2011.

[56] Joseph Naor and Leonid Zosin. A 2-approximation algorithm for the directed multiway cut problem. *SIAM J. Comput.*, 31(2):477–482, 2001.

[57] James B. Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Mathematicae (Proceedings)*, 80(5):406–424, 1977.

[58] Hans-Peter Piepho. An algorithm for a letter-based representation of all-pairwise comparisons. *Journal of Computational and Graphical Statistics*, 13(2):456–466, 2004.

[59] Subramanian Rajagopalan, Manish Vachharajani, and Sharad Malik. Handling irregular ILP within conventional VLIW schedulers using artificial resource constraints. In *CASES*, pages 157–164, 2000.

[60] Igor Razgon. Computing multiway cut within the given excess over the largest minimum isolating cut. *CoRR*, abs/1011.6267, 2010.

[61] Igor Razgon. Large isolating cuts shrink the multiway cut. *CoRR*, abs/1104.5361, 2011.

[62] Igor Razgon and Barry O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.

[63] Bruce A. Reed, Kaleigh Smith, and Adrian Vetta. Finding odd cycle transversals. *Oper. Res. Lett.*, 32(4):299–301, 2004.

[64] Fred S. Roberts. Applications of edge coverings by cliques. *Discrete Applied Mathematics*, 10(1):93 – 109, 1985.

[65] Mikkel Thorup. Minimum k-way cuts via deterministic greedy tree packing. In *STOC*, pages 159–166. ACM, 2008.

[66] Mingyu Xiao. Simple and improved parameterized algorithms for multiterminal cuts. *Theory Comput. Syst.*, 46(4):723–736, 2010.

[67] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983.