# Tight bounds for Parameterized Complexity of Cluster Editing*

**Fedor V. Fomin[1], Stefan Kratsch[2], Marcin Pilipczuk[3], Michał Pilipczuk[1], and Yngve Villanger[1]**

1 **Department of Informatics, University of Bergen, Bergen, Norway,** `{fomin,michal.pilipczuk,yngve.villanger}@ii.uib.no`

2 **MPI Informatics, Saarbrücken, Germany,** `skratsch@mpi-inf.mpg.de`

3 **Institute of Informatics, University of Warsaw, Poland,** `malcin@mimuw.edu.pl`

## Abstract

In the Correlation Clustering problem, also known as Cluster Editing, we are given an undirected graph $G$ and a positive integer $k$; the task is to decide whether $G$ can be transformed into a cluster graph, i.e., a disjoint union of cliques, by changing at most $k$ adjacencies, that is, by adding or deleting at most $k$ edges. The motivation of the problem stems from various tasks in computational biology (Ben-Dor et al., Journal of Computational Biology 1999) and machine learning (Bansal et al., Machine Learning 2004). Although in general Correlation Clustering is APX-hard (Charikar et al., FOCS 2003), the version of the problem where the number of cliques may not exceed a prescribed constant $p$ admits a PTAS (Giotis and Guruswami, SODA 2006).

We study the parameterized complexity of Correlation Clustering with this restriction on the number of cliques to be created. We give an algorithm that

- in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + n + m)$ decides whether a graph $G$ on $n$ vertices and $m$ edges can be transformed into a cluster graph with exactly $p$ cliques by changing at most $k$ adjacencies.

We complement these algorithmic findings by the following, surprisingly tight lower bound on the asymptotic behavior of our algorithm. We show that unless the Exponential Time Hypothesis (ETH) fails

- for any constant $0 \leq \sigma \leq 1$, there is $p = \Theta(k^\sigma)$ such that there is no algorithm deciding in time $2^{o(\sqrt{pk})} \cdot n^{\mathcal{O}(1)}$ whether an $n$-vertex graph $G$ can be transformed into a cluster graph with at most $p$ cliques by changing at most $k$ adjacencies.

Thus, our upper and lower bounds provide an asymptotically tight analysis of the multivariate parameterized complexity of the problem for the whole range of values of $p$ from constant to a linear function of $k$.

**1998 ACM Subject Classification** F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

**Keywords and phrases** parameterized complexity, cluster editing, correlation clustering, subexponential algorithms, tight bounds

**Digital Object Identifier** 10.4230/LIPIcs.STACS.2013.32

* The authors from University of Bergen are supported by the European Research Council (ERC) via grant Rigorous Theory of Preprocessing, reference 267959 and by the Research Council of Norway. The third author is supported by the National Science Centre grant N206 567140 and Foundation for Polish Science.

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 32–43
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

# 1 Introduction

*Correlation clustering*, also known as *clustering with qualitative information* or *cluster editing*, is the problem to cluster objects based only on the qualitative information concerning similarity between pairs of them. For every pair of objects we have a binary indication whether they are similar or not. The task is to find a partition of the objects into clusters minimizing the number of similarities between different clusters and non-similarities inside of clusters. The problem was introduced by Ben-Dor, Shamir, and Yakhini [6] motivated by problems from computational biology, and, independently, by Bansal, Blum, and Chawla [5], motivated by machine learning problems concerning document clustering according to similarities. The correlation version of clustering was studied intensively, including [1, 3, 4, 13, 14, 24, 34].

The graph-theoretic formulation of the problem is the following. A graph $K$ is a *cluster graph* if every connected component of $K$ is a complete graph. Let $G = (V, E)$ be a graph; then $F \subseteq V \times V$ is called a *cluster editing set* for $G$ if $G \triangle F = (V, E \triangle F)$ is a cluster graph. Here $E \triangle F$ is the symmetric difference between $E$ and $F$. In the optimization version of the problem the task is to find a cluster editing set of minimum size. Constant factor approximation algorithms for this problem were obtained in [1, 5, 13]. On the negative side, the problem is known to be NP-complete [34] and, as was shown by Charikar, Guruswami, and Wirth [13], also APX-hard.

Giotis and Guruswami [24] initiated the study of clustering when the maximum number of clusters that we are allowed to use is stipulated to be a fixed constant $p$. As observed by them, this type of clustering is well-motivated in settings where the number of clusters might be an external constraint that has to be met. It appeared that $p$-clustering variants posed new and non-trivial challenges. In particular, in spite of the APX-hardness of the general case, Giotis and Guruswami [24] gave a PTAS for this version of the problem.

A cluster graph $G$ is called a *$p$-cluster graph* if it has exactly $p$ connected components or, equivalently, if it is a disjoint union of exactly $p$ cliques. Similarly, a set $F$ is a *$p$-cluster editing set* of $G$, if $G \triangle F$ is a $p$-cluster graph. In parameterized complexity, correlation clustering and its restriction to bounded number of clusters were studied under the names Cluster Editing and $p$-Cluster Editing, respectively.

Cluster Editing *Parameter:* $k$.
*Input:* A graph $G = (V, E)$ and a non-negative integer $k$.
*Question:* Is there a cluster editing set for $G$ of size at most $k$?

$p$-Cluster Editing *Parameters:* $p, k$.
*Input:* A graph $G = (V, E)$ and non-negative integers $p$ and $k$.
*Question:* Is there a $p$-cluster editing set for $G$ of size at most $k$?

The parameterized version of Cluster Editing, and variants of it, were studied intensively [7, 8, 9, 10, 11, 16, 20, 25, 27, 28, 31, 33]. The problem is solvable in time $\mathcal{O}(1.62^k + n + m)$ [7] and it has a kernel with $2k$ vertices [12, 15] (see Section 2 for the definition of a kernel). Shamir et al. [34] showed that $p$-Cluster Editing is NP-complete for every fixed $p \geq 2$. A kernel with $(p + 2)k + p$ vertices was given by Guo [26].

## Our results

We study the impact of the interaction between $p$ and $k$ on the parameterized complexity of $p$-Cluster Editing. Our main algorithmic result is the following.

▶ **Theorem 1.** *$p$-Cluster Editing is solvable in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + m + n)$.*

It is straightforward to modify our algorithm to work also in the following variants of the problem, where each edge and non-edge is assigned some edition cost: either *(i)* all costs are at least one and $k$ is the bound on the maximum total cost of the solution, or *(ii)* we ask for a set of at most $k$ edits of minimum cost. Let us also remark that, by Theorem 1, if $p = o(k)$ then $p$-Cluster Editing can be solved in $2^{o(k)}n^{\mathcal{O}(1)}$ time, and thus it belongs to complexity class SUBEPT defined by Flum and Grohe [21, Chapter 16]. Until very recently, the only problems known to be in the class SUBEPT were the problems with additional constraints on the input, like being a planar, $H$-minor-free, or tournament graph [2, 17]. However, recent algorithmic developments indicate that the structure of the class SUBEPT is much more interesting than expected. It appears that some parameterized problems related to chordal graphs, like Minimum Fill-in or Chordal Graph Sandwich, are also in SUBEPT [23].

We would like to remark that $p$-Cluster Editing can be also solved in worse time complexity $\mathcal{O}((pk)^{\mathcal{O}(\sqrt{pk})} + m + n)$ using simple guessing arguments. One such algorithm is based on the following observation: Suppose that, for some integer $r$, we know at least $2r+1$ vertices from each cluster. Then, if an unassigned vertex has at most $r$ incident modifications, we know precisely to which cluster it belongs: it is adjacent to at least $r+1$ vertices already assigned to its cluster and at most $r$ assigned to any other cluster. On the other hand, there are at most $2k/r$ vertices with more than $r$ incident modifications. Thus (i) guessing $2r+1$ vertices from each cluster (or all of them, if there are less than $2r+1$), and (ii) guessing all vertices with more than $r$ incident modifications, together with their alignment to clusters, results in at most $n^{(2r+1)p}n^{2k/r}p^{2k/r}$ subcases. By pipelining it with the kernelization of Guo [26] and with simple reduction rules that ensure $p \leq 6k$ (see Section 3.1 for details), we obtain the claimed time complexity for $r \sim \sqrt{k/p}$.

An approach via *chromatic coding*, introduced by Alon et al. [2], also leads to an algorithm with running time $\mathcal{O}(2^{\mathcal{O}(p\sqrt{k}\log p)} + n + m)$. However, one needs to develop new concepts to construct an algorithm for $p$-Cluster Editing with complexity bound as promised in Theorem 1, and thus obtain a subexponential complexity for every sublinear $p$.

The crucial observation is that a $p$-cluster graph, for $p = \mathcal{O}(k)$, has $2^{\mathcal{O}(\sqrt{pk})}$ edge cuts of size at most $k$ (henceforth called *$k$-cuts*). As in a YES-instance to the $p$-Cluster Editing problem each $k$-cut is a $2k$-cut of a $p$-cluster graph, we infer a similar bound on the number of cuts if we are dealing with a YES-instance. This allows us to use dynamic programming over the set of $k$-cuts. Pipelining this approach with a kernelization algorithm for $p$-Cluster Editing proves Theorem 1.

A new and active direction in parameterized complexity is the pursuit of asymptotically tight bounds on the complexity of problems. In several cases, it is possible to obtain a complete analysis by providing matching lower (complexity) and upper (algorithmic) bounds. We refer to the recent survey of Marx [32], where recent developments in the area are discussed, and the "optimality program" is announced among the main future research directions in parameterized complexity. The most widely used complexity assumption for such tight lower bounds is the *Exponential Time Hypothesis (ETH)*, which posits that no subexponential-time algorithms for $k$-CNF-SAT or CNF-SAT exist [29].

Following this direction, we complement Theorem 1 with two lower bounds. Our first, main lower bound is based on the following technical Theorem 2, which shows that the exponential time dependence of our algorithm is asymptotically tight for any choice of parameters $p$ and $k$, where $p = \mathcal{O}(k)$. As one can provide polynomial-time reduction rules that ensure that $p \leq 6k$ (see Section 3.1 for details), this provides a full and tight picture of the multivariate parameterized complexity of $p$-Cluster Editing: we have asymptotically matching upper and lower bounds on the whole interval between $p$ being a constant and linear in $k$. To the best of our knowledge, this is the first fully multivariate and tight complexity analysis of a parameterized problem.

▶ **Theorem 2.** *For any $\varepsilon > 0$ there is $\delta > 0$ and a polynomial-time algorithm that, given positive integers $p$ and $k$ and a 3-CNF-SAT formula $\Phi$ with $n$ variables and $m$ clauses, such that $k, n \geq \varepsilon p$ and $n, m \leq \sqrt{pk}/\varepsilon$, computes a graph $G$ and integer $k'$, such that $k' \leq \delta k$, $|V(G)| \leq \delta\sqrt{pk}$, and*

- *if $\Phi$ is satisfiable then there is a $6p$-cluster graph $G_0$ with $V(G) = V(G_0)$ and such that $|E(G) \triangle E(G_0)| \leq k'$;*
- *if there exists a $p'$-cluster graph $G_0$ with $p' \leq 6p$, $V(G) = V(G_0)$ and $|E(G) \triangle E(G_0)| \leq k'$, then $\Phi$ is satisfiable.*

As the statement of Theorem 2 may look technical, we gather its two main consequences in Corollaries 3 and 4. We state both corollaries in terms of an easier $p_{\leq}$-Cluster Editing problem, where the number of clusters has to be at most $p$ instead of precisely equal to $p$. Clearly, this version can be solved by an algorithm for $p$-Cluster Editing with an additional $p$ overhead in time complexity by trying all possible $p' \leq p$, so the lower bound holds also for harder $p$-Cluster Editing; however, we are not aware of any reduction in the opposite direction. In both corollaries we use the fact that existence of a subexponential, in both the number of variables and clauses, algorithm for verifying satisfiability of 3-CNF-SAT formulas would violate ETH [29].

▶ **Corollary 3** (♠[1]). *Unless ETH fails, for every $0 \leq \sigma \leq 1$, there is $p = \Theta(k^\sigma)$ such that $p_{\leq}$-Cluster Editing is not solvable in time $2^{o(\sqrt{pk})}|V(G)|^{\mathcal{O}(1)}$.*

▶ **Corollary 4** (♠). *Unless ETH fails, for every constant $p \geq 6$, there is no algorithm solving $p_{\leq}$-Cluster Editing in time $2^{o(\sqrt{k})}|V(G)|^{\mathcal{O}(1)}$ or $2^{o(|V(G)|)}$.*

Note that Theorem 2 and Corollary 3 do not rule out possibility that the general Cluster Editing is solvable in subexponential time. Our second, complementary lower bound shows that when the number of clusters is not constrained, then the problem cannot be solved in subexponential time unless ETH fails. This disproves the conjecture of Cao and Chen [12]. We note that Theorem 5 was independently obtained by Komusiewicz in his PhD thesis [30].

▶ **Theorem 5** (♠). *Unless ETH fails,* Cluster Editing *cannot be solved in time $2^{o(k)}n^{\mathcal{O}(1)}$.*

Clearly, by Theorem 1, the reduction of Theorem 5 must produce an instance where the number of clusters in any solution, if there exists any, is $\Omega(k)$. Therefore, intuitively the hard instances of Cluster Editing are those where every cluster needs just a constant number of adjacent editions to be extracted.

## 2 Preliminaries

We use $n$ to denote the number of vertices and $m$ the number of edges in the input graph $G$. For graphs $G, H$ with $V(G) = V(H)$, by $\mathcal{H}(G, H)$ we denote the number of edge modifications needed to obtain $H$ from $G$, i.e., $\mathcal{H}(G, H) = |E(G) \triangle E(H)|$. By $E(X, Y)$ we denote the set of edges having one endpoint in $X$ and second in $Y$.

[1] Due to space constraints, the proofs of all statements marked with ♠ are omitted. The full version of this paper is available at http://arxiv.org/abs/1112.4419.

A parameterized problem $\Pi$ is a subset of $\Gamma^* \times \mathbb{N}$ for some finite alphabet $\Gamma$. An instance of a parameterized problem consists of $(x, k)$, where $k$ is called the parameter. A central notion in parameterized complexity is *fixed-parameter tractability (FPT)* which means, for a given instance $(x, k)$, solvability in time $f(k) \cdot p(|x|)$, where $f$ is an arbitrary computable function of $k$ and $p$ is a polynomial in the input size. We refer to the book of Downey and Fellows [19] for further reading on parameterized complexity.

A *kernelization algorithm* for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that given $(x, k) \in \Gamma^* \times \mathbb{N}$ outputs in time polynomial in $|x| + k$ a pair $(x', k') \in \Gamma^* \times \mathbb{N}$, called a *kernel* such that $(x, k) \in \Pi$ if and only if $(x', k') \in \Pi$, $|x'| \leq g(k)$, and $k' \leq k$, where $g$ is some computable function.

We also need the following result of Guo [26].

▶ **Proposition 6** ([26]). $p$-Cluster Editing admits a kernel with $(p + 2)k + p$ vertices. The running time of the kernelization algorithm is $\mathcal{O}(n + m)$, where $n$ is the number of vertices and $m$ the number of edges in the input graph $G$.

## 3 A subexponential algorithm for $p$-Cluster Editing

In this section we prove Theorem 1, that is, we show a $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + n + m)$-time algorithm for $p$-Cluster Editing.

### 3.1 Reduction for large $p$

The first step of our algorithm is an application of the kernelization algorithm by Guo [26] (Proposition 6) followed by some additional preprocessing rules that ensure that $p \leq 6k$. These additional rules are encapsulated in the following technical lemma.

▶ **Lemma 7** (♠). *There exists a polynomial time algorithm that, given an instance $(G, p, k)$ of $p$-Cluster Editing, outputs an equivalent instance $(G', p', k)$, where $G'$ is an induced subgraph of $G$ and $p' \leq 6k$.*

The key idea behind Lemma 7 is the observation that if $p > 2k$, then at least $p - 2k$ clusters in the final cluster graph cannot be touched by the solution, hence they must have been present as isolated cliques already in the beginning. Hence, if $p > 6k$ then we have to already see $p - 2k > 4k$ isolated cliques; otherwise, we may safely provide a negative answer. Although these cliques may be still merged (to decrease the number of clusters) or split (to increase the number of clusters), we can apply greedy arguments to identify a clique that may be safely assumed to be untouched by the solution. Hence we can remove it from the graph and decrement $p$ by one. Although the greedy arguments seem very intuitive, their formal proofs turn out to be somewhat technical.

### 3.2 Small cuts

We now proceed to the algorithm itself. Let us introduce the key notion.

▶ **Definition 8.** Let $G = (V, E)$ be an undirected graph. A partition $(V_1, V_2)$ of $V$ is called a *$k$-cut of $G$* if $|E(V_1, V_2)| \leq k$.

▶ **Lemma 9.** *$k$-cuts of a graph $G$ can be enumerated with polynomial time delay.*

**Proof.** We follow the standard branching. We order the vertices arbitrarily, start with empty $V_1$, $V_2$ and for each consecutive vertex $v$ we branch into two subcases: we put $v$ either into $V_1$

or into $V_2$. Once the alignment of all vertices is decided, we output the partition. However, each time we put a vertex in one of the sets, we run a polynomial-time max-flow algorithm to check whether the minimum edge cut between $V_1$ and $V_2$ constructed so far is at most $k$. If not, then we terminate this branch as it certainly cannot result in any solutions found. Thus, we always pursue a branch that results in at least one feasible solution, and finding the next solution occurs within a polynomial number of steps. ◀

Intuitively, $k$-cuts of the graph $G$ form the search space of the algorithm. Therefore, we would like to bound their number. We do this by firstly bounding the number of cuts of a cluster graph, and then using the fact that a YES-instance is not very far from some cluster graph. We begin with the following bound on binomial coefficients.

▶ **Lemma 10** (♠)**.** *If $a, b$ are nonnegative integers, then $\binom{a+b}{a} \leq 2^{2\sqrt{ab}}$.*

▶ **Lemma 11.** *Let $K$ be a cluster graph containing at most $p$ clusters, where $p \leq 6k$. Then the number of $k$-cuts of $K$ is at most $2^{8\sqrt{pk}}$.*

**Proof.** By slightly abusing the notation, assume that $K$ has exactly $p$ clusters, some of which may be empty. Let $C_1, C_2, \ldots, C_p$ be these clusters and $c_1, c_2, \ldots, c_p$ be their sizes, respectively. We firstly establish a bound on the number of partitions $(V_1, V_2)$ such that the cluster $C_i$ contains $x_i$ vertices from $V_1$ and $y_i$ from $V_2$. Then we discuss how to bound the number of ways of selecting pairs $x_i, y_i$ summing up to $c_i$ for which the number of $k$-cuts is positive. Multiplying the obtained two bounds gives us the claim.

Having fixed the numbers $x_i, y_i$, the number of ways in which the cluster $C_i$ can be partitioned is equal to $\binom{x_i+y_i}{x_i}$. Note that $\binom{x_i+y_i}{x_i} \leq 2^{2\sqrt{x_i y_i}}$ by Lemma 10. Observe that there are $x_i y_i$ edges between $V_1$ and $V_2$ inside the cluster $C_i$, so if $(V_1, V_2)$ is a $k$-cut, then $\sum_{i=1}^{p} x_i y_i \leq k$. By applying the Cauchy-Schwarz inequality we infer that $\sum_{i=1}^{p} \sqrt{x_i y_i} \leq \sqrt{p} \cdot \sqrt{\sum_{i=1}^{p} x_i y_i} \leq \sqrt{pk}$. Therefore, the number of considered cuts is bounded by

$$\prod_{i=1}^{p} \binom{x_i + y_i}{x_i} \leq 2^{2\sum_{i=1}^{p}\sqrt{x_i y_i}} \leq 2^{2\sqrt{pk}}.$$

Moreover, observe that $\min(x_i, y_i) \leq \sqrt{x_i y_i}$; hence, $\sum_{i=1}^{p} \min(x_i, y_i) \leq \sqrt{pk}$. Thus, the choice of $x_i, y_i$ can be modeled by first choosing for each $i$, whether $\min(x_i, y_i)$ is equal to $x_i$ or to $y_i$, and then expressing $\lfloor\sqrt{pk}\rfloor$ as the sum of $p+1$ nonnegative numbers: $\min(x_i, y_i)$ for $1 \leq i \leq p$ and the rest, $\lfloor\sqrt{pk}\rfloor - \sum_{i=1}^{p} \min(x_i, y_i)$. The number of choices in the first step is equal to $2^p \leq 2^{\sqrt{6pk}}$, and in the second is equal to $\binom{\lfloor\sqrt{pk}\rfloor+p}{p} \leq 2^{\sqrt{pk}+\sqrt{6pk}}$. Therefore, the number of possible choices of $x_i, y_i$ is bounded by $2^{(1+2\sqrt{6})\sqrt{pk}} \leq 2^{6\sqrt{pk}}$. Hence, the total number of $k$-cuts is bounded by $2^{6\sqrt{pk}} \cdot 2^{2\sqrt{pk}} = 2^{8\sqrt{pk}}$, as claimed. ◀

▶ **Lemma 12.** *If $(G, p, k)$ is a YES-instance of $p$-*Cluster Editing *with $p \leq 6k$, then the number of $k$-cuts of $G$ is bounded by $2^{8\sqrt{2pk}}$.*

**Proof.** Let $K$ be a cluster graph with at most $p$ clusters such that $\mathcal{H}(G, K) \leq k$. Observe that every $k$-cut of $G$ is also a $2k$-cut of $K$, as $K$ differs from $G$ by at most $k$ edge modifications. The claim follows from Lemma 11. ◀

## 3.3 The algorithm

**Proof of Theorem 1.** Let $(G = (V, E), p, k)$ be the given $p$-Cluster Editing instance. By making use of Proposition 6, we can assume that $G$ has at most $(p+2)k+p$ vertices, thus all the factors polynomial in the size of $G$ can be henceforth hidden within the $2^{\mathcal{O}(\sqrt{pk})}$ factor. Application of Proposition 6 gives the additional $\mathcal{O}(n+m)$ summand to the complexity. By further usage of Lemma 7 we can also assume that $p \leq 6k$. Note that application of Lemma 7 can spoil the bound $|V(G)| \leq (p+2)k+p$ as $p$ can decrease; however the number of vertices of the graph is still bounded in terms of initial $p$ and $k$.

We now enumerate $k$-cuts of $G$ with polynomial time delay. If we exceed the bound $2^{8\sqrt{2pk}}$ given by Lemma 12, we know that we can safely answer NO, so we immediately terminate the computation and give a negative answer. Therefore, we can assume that we have computed the set $\mathcal{N}$ of all $k$-cuts of $G$ and $|\mathcal{N}| \leq 2^{8\sqrt{2pk}}$.

Assume that $(G,p,k)$ is a YES-instance and let $K$ be a cluster graph with at most $p$ clusters such that $\mathcal{H}(G,K) \leq k$. Again, let $C_1, C_2, \ldots, C_p$ be the clusters of $K$. Observe that for every $j \in \{0,1,2,\ldots,p\}$, the partition $\left(\bigcup_{i=1}^{j} V(C_i), \bigcup_{i=j+1}^{p} V(C_i)\right)$ has to be the $k$-cut with respect to $G$, as otherwise there would be more than $k$ edges that need to be deleted from $G$ in order to obtain $K$. This observation enables us to use a dynamic programming approach on the set of cuts.

We construct a directed graph $D$, whose vertex set is equal to $\mathcal{N} \times \{0,1,2,\ldots,p\} \times \{0,1,2,\ldots,k\}$; note that $|V(D)| = 2^{\mathcal{O}(\sqrt{pk})}$. We create arcs going from $((V_1,V_2),j,\ell)$ to $((V_1',V_2'),j+1,\ell')$, where $V_1 \subsetneq V_1'$ (hence $V_2 \supsetneq V_2'$), $j \in \{0,1,2,\ldots,p-1\}$ and $\ell' = \ell + |E(V_1, V_1' \setminus V_1)| + |\overline{E}(V_1' \setminus V_1, V_1' \setminus V_1)|$ ($(V,\overline{E})$ is the complement of the graph $G$). The arcs can be constructed in $2^{\mathcal{O}(\sqrt{pk})}$ time by checking for all the pairs of vertices whether they should be connected. We claim that the answer to the instance $(G,p,k)$ is equivalent to reachability of any of the vertices of form $((V,\emptyset),p,\ell)$ from the vertex $((\emptyset,V),0,0)$.

In one direction, if there is a path from $((\emptyset,V),0,0)$ to $((V,\emptyset),p,\ell)$ for some $\ell \leq k$, then the consecutive sets $V_1' \backslash V_1$ along the path form clusters $C_i$ of a cluster graph $K$, whose editing distance to $G$ is accumulated on the last coordinate, thus bounded by $k$. In the second direction, if there is a cluster graph $K$ with clusters $C_1, C_2, \ldots, C_p$ within editing distance at most $k$ from $G$, then vertices $\left(\left(\bigcup_{i=1}^{j} V(C_i), \bigcup_{i=j+1}^{p} V(C_i)\right), j, \mathcal{H}\left(G\left[\bigcup_{i=1}^{j} V(C_i)\right], K\left[\bigcup_{i=1}^{j} V(C_i)\right]\right)\right)$ form a path from $((\emptyset,V),0,0)$ to $((V,\emptyset),p,\mathcal{H}(G,K))$. Note that all these triples are indeed vertices of the graph $D$, as $\left(\bigcup_{i=1}^{j} V(C_i), \bigcup_{i=j+1}^{p} V(C_i)\right)$ are $k$-cuts of $G$.

Reachability in a directed graph can be tested in linear time with respect to the number of vertices and arcs. We can now apply this algorithm to the graph $D$ and conclude solving the $p$-Cluster Editing instance in $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + n + m)$ time. ◀

## 4 Multivariate lower bound

This section is devoted to sketching the proof of Theorem 2. As the provided reduction is very technical, in this extended abstract we only provide the construction of the graph $G$, explaining also all the necessary intuition, and sketch the completeness implication, i.e., how to translate a satisfying assignment of $\Phi$ into a $6p$-cluster graph $G_0$ close to $G$. To ease the presentation, in this extended abstract we show the proof for $\varepsilon = 1$.

### 4.1 Preprocessing of the formula

We start with a step that regularizes the input formula $\Phi$, while increasing its size only by a constant factor. The purpose of this step is to ensure that, when we translate a satisfying assignment of $\Phi$ into a cluster graph $G_0$ in the completeness step, the clusters are of the same size, and therefore contain the minimum possible number of edges. This property is crucial in the argumentation of the soundness step. The proof of the following lemma consists of several steps that ensure consecutive properties of formula $\Phi'$ by syntactic modifications, like copying variables and clauses.

▶ **Lemma 13** (♠). *There exists a polynomial-time algorithm that, given a 3-CNF formula $\Phi$ with $n$ variables and $m$ clauses and an integer $p \leq n$, constructs a 3-CNF formula $\Phi'$ with $n'$ variables and $m'$ clauses together with a partition of the variable set $\mathrm{Vars}(\Phi')$ into $p$ parts $\mathrm{Vars}^r$, $1 \leq r \leq p$, such that the following properties hold:*

*(a) $\Phi'$ is satisfiable iff $\Phi$ is;*

*(b) in $\Phi'$ every clause contains exactly three literals corresponding to different variables;*

*(c) in $\Phi'$ every variable appears exactly three times positively and exactly three times negatively;*

*(d) $n'$ is divisible by $p$ and, for each $1 \leq r \leq p$, we have $|\mathrm{Vars}^r| = n'/p$ (i.e., the variables are split evenly between the parts $\mathrm{Vars}^r$);*

*(e) if $\Phi'$ is satisfiable, then there exists a satisfying assignment of $\mathrm{Vars}(\Phi')$ with the property that in each part $\mathrm{Vars}^r$ the numbers of variables set to true and to false are equal.*

*(f) $n' + m' = \mathcal{O}(n + m)$.*

## 4.2 Construction

We now sketch how to compute the graph $G$ and the integer $k'$ from the formula $\Phi'$ given by Lemma 13. As Lemma 13 increases the size of the formula by a constant factor, we have that $n', m' = \mathcal{O}(\sqrt{pk})$ and $|\mathrm{Vars}^r| = n'/p = \mathcal{O}(\sqrt{k/p})$ for $1 \leq r \leq p$. The idea is to pack the variables from each part $\mathrm{Vars}^r$, for $1 \leq r \leq p$, into group gadgets, each costing 6 cliques. Evaluation of the variables from each part corresponds to some clustering strategy inside the group gadget. The clauses are encoded by additional groups of vertices, whose connections to group gadgets ensure that they can be split among the clusters optimally iff at least one literal satisfies the clause.

We proceed with the description of group gadgets. Let $L = 1000 \cdot \left(1 + \frac{n'}{p}\right) = \mathcal{O}(\sqrt{k/p})$. For each part $\mathrm{Vars}^r$, $1 \leq r \leq p$, we create six cliques $Q^r_\alpha$, $1 \leq \alpha \leq 6$, each of size $L$. Let $\mathcal{Q}$ be the set of all vertices of all cliques $Q^r_\alpha$. In this manner we have $6p$ cliques. Intuitively, if we seek for a $6p$-cluster graph close to $G$, then the cliques are large enough so that merging two cliques is too expensive — in the intended solution we have exactly one clique in each cluster. One may view the construction as a procedure of assigning vertices not from $\mathcal{Q}$ to different cliques $Q^r_\alpha$.

For every variable $x \in \mathrm{Vars}^r$, we create six vertices $w^x_{1,2}, w^x_{2,3}, \ldots, w^x_{5,6}, w^x_{6,1}$. Connect them into a cycle in this order; this cycle is called a 6*-cycle for the variable* $x$. Moreover, for each $1 \leq \alpha \leq 6$ and $v \in V(Q^r_\alpha)$, create edges $vw^x_{\alpha-1,\alpha}$ and $vw^x_{\alpha,\alpha+1}$ (we assume that the indices behave cyclically, i.e., $w^x_{6,7} = w^x_{6,1}$, $Q^r_7 = Q^r_1$ etc.). Let $\mathcal{W}$ be the set of all vertices $w^x_{\alpha,\alpha+1}$ for all variables $x$. Intuitively, the cheapest way to cut the 6-cycle for variable $x$ is to assign the vertices $w^x_{\alpha,\alpha+1}$, $1 \leq \alpha \leq 6$, all either to the clusters with cliques with only odd indices or only with even indices. Choosing even indices corresponds to setting $x$ to false, while choosing odd ones corresponds to setting $x$ to true, and both choices lead to saving exactly 3 editions inside the 6-cycle. By property (e) of formula $\Phi'$ we know that if $\Phi'$ is satisfiable, then in some satisfying assignment exactly half of the variables in each group are assigned true value, and half false. For this satisfying assignment, each clique $Q^r_\alpha$ will be assigned exactly the same number of vertices from $\mathcal{W}$.

We now proceed with the description of the encoding of the clauses. Let $r(x)$ be the index of the part that contains variable $x$, that is, $x \in \mathrm{Vars}^{r(x)}$. In each clause $C$ we (arbitrarily) enumerate variables: for $1 \leq \eta \leq 3$, let $\mathrm{var}(C,\eta)$ be the variable in the $\eta$-th literal of $C$, and $\mathrm{sgn}(C,\eta) = 0$ if the $\eta$-th literal is negative and $\mathrm{sgn}(C,\eta) = 1$ otherwise.

For every clause $C$ create nine vertices: $s^C_{\beta,\xi}$ for $1 \leq \beta, \xi \leq 3$. Let $\mathcal{S}$ be the set of all the vertices created in this manner. Let us first focus on vertices $s^C_{1,1}, s^C_{1,2}, s^C_{1,3}$.

- For each $1 \leq \eta \leq 3$ and each $\xi \in \{1,2,3\}$, create an edge $s^C_{1,\xi} w^{\mathrm{var}(C,\eta)}_{2\eta-1,2\eta}$;
- for each $1 \leq \eta \leq 3$ connect $s^C_{1,1}$ to all the vertices of one of the cliques adjacent to $w^{\mathrm{var}(C,\eta)}_{2\eta-1,2\eta}$ depending on the sign of the $\eta$-th literal in $C$, that is, the clique $Q^{r(\mathrm{var}(C,\eta))}_{2\eta-\mathrm{sgn}(C,\eta)}$;
- for each $1 \leq \eta \leq 3$ and $\xi \in \{2,3\}$, connect $s^C_{1,\xi}$ to all vertices of both cliques the vertex $w^{\mathrm{var}(C,\eta)}_{2\eta-1,2\eta}$ is adjacent to, that is, the cliques $Q^{r(\mathrm{var}(C,\eta))}_{2\eta-1}$ and $Q^{r(\mathrm{var}(C,\eta))}_{2\eta}$.

In this manner, vertex $s^C_{1,1}$ is adjacent to three cliques $Q^r_\alpha$, while $s^C_{1,2}$ and $s^C_{1,3}$, which are twins, are adjacent to six of them. Assuming that each clique $Q^r_\alpha$ is in a different cluster, we need to edit two connections to the cliques for vertex $s^C_{1,1}$, and five for each of vertices $s^C_{1,2}$, $s^C_{1,3}$. Checking satisfaction of the assignment is performed on the edges between $s^C_{1,1}$ and vertices from $\mathcal{W}$. The crucial observation is that:

- if at least one of the literals in the clause is satisfied, then at least one of the three vertices from $\mathcal{W}$ adjacent to $s^C_{1,1}$ is already assigned to a clique that is connected to $s^C_{1,1}$.
- if none of the literals of the clause is satisfied, then all the vertices from $\mathcal{W}$, to which $s^C_{1,1}$ is adjacent, are assigned to cliques not connected to $s^C_{1,1}$.

Hence, if the first possibility takes place, we can save one edition by not changing adjacency between $s^C_{1,1}$ and the corresponding vertex from $\mathcal{W}$. However, if the second possibility takes place, we need to change all three adjacencies, unless we want to separate $s^C_{1,1}$ from all the three adjacent cliques $Q^r_\alpha$, which is too expensive.

Vertices $s^C_{1,2}$ and $s^C_{1,3}$ help us to balance the sizes of the clusters, as we may assign them to any clique that is adjacent to them. For example, if $s^C_{1,1}$ was assigned to $Q^{r(x)}_1$, then we can assign $s^C_{1,2}$ to $Q^{r(y)}_3$ and $s^C_{1,3}$ to $Q^{r(z)}_6$. The construction of vertices $\{s^C_{2,1}, s^C_{2,2}, s^C_{2,3}\}$ and $\{s^C_{3,1}, s^C_{3,2}, s^C_{3,3}\}$ follow the same rules, but the lower indices of the cliques and vertices from $\mathcal{W}$ to which the constructed vertices are adjacent, are cyclically shifted by 2 and 4, respectively. In this manner we are able to ensure the following properties: if the assignment satisfies clause $C$, then vertices $s^C_{\beta,\xi}$ can be assigned to the cliques so that (i) each vertex is assigned to a clique it is connected to, (ii) for each vertex we save one edition on editing adjacencies to vertices from $\mathcal{W}$, (iii) each clique with an odd lower index is assigned one vertex if the corresponding literal appears positively in $C$, and zero otherwise, (iv) each clique with an even lower index is assigned one vertex if the corresponding literal appears negatively in $C$, and zero otherwise. By property (c) of the formula $\Phi'$ we know that for the satisfying assignment all the cliques are assigned exactly the same number of vertices from $\mathcal{S}$.

This concludes the construction. We note that $|V(G)| = 6pL + \mathcal{O}(n' + m') = \mathcal{O}(\sqrt{pk})$.

We now calculate the budget $k'$ for edge editions in the created instance. Then we argue why in case of existence of a satisfying assignment there is a set of at most $k'$ edge editions that turns $G$ into a $6p$-cluster graph. (The argument for the converse is deferred to the full version.) In the constructed solution all the cliques $Q^r_\alpha$ will be in different clusters.

To make the presentation more clear, we split this budget into few summands. Let

$$k_{\mathcal{Q}-\mathcal{Q}} = 0, \qquad k_{\mathcal{Q}-\mathcal{WS}} = (6n' + 36m')L, \qquad k^{\mathrm{all}}_{\mathcal{WS}-\mathcal{WS}} = 6p\binom{\frac{6n'+9m'}{6p}}{2},$$

$$k^{\mathrm{exist}}_{\mathcal{WS}-\mathcal{WS}} = 6n' + 27m', \qquad k^{\mathrm{save}}_{\mathcal{W}-\mathcal{W}} = 3n', \qquad k^{\mathrm{save}}_{\mathcal{W}-\mathcal{S}} = 9m',$$

and finally

$$k' = k_{\mathcal{Q}-\mathcal{Q}} + k_{\mathcal{Q}-\mathcal{WS}} + k^{\mathrm{all}}_{\mathcal{WS}-\mathcal{WS}} + k^{\mathrm{exist}}_{\mathcal{WS}-\mathcal{WS}} - 2k^{\mathrm{save}}_{\mathcal{W}-\mathcal{W}} - 2k^{\mathrm{save}}_{\mathcal{W}-\mathcal{S}}.$$

Note that, as $p \leq k$, $L = \mathcal{O}(\sqrt{k/p})$ and $n', m' = \mathcal{O}(\sqrt{pk})$, we have $k' = \mathcal{O}(k)$.

The intuition behind this split is as follows. The intended solution for the $p$-Cluster Editing instance $(G, 6p, k')$ creates no edges between the cliques $Q^r_\alpha$, each clique is contained in its own cluster, and $k_{\mathcal{Q}-\mathcal{Q}} = 0$. For each $v \in \mathcal{W} \cup \mathcal{S}$, the vertex $v$ is assigned to a cluster with one clique adjacent to $v$; $k_{\mathcal{Q}-\mathcal{WS}}$ accumulates the cost of removal of other edges in $E(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})$. Finally, we count the editions in $(\mathcal{W} \cup \mathcal{S}) \times (\mathcal{W} \cup \mathcal{S})$ in an indirect way. First we cut all edges of $E(\mathcal{W} \cup \mathcal{S}, \mathcal{W} \cup \mathcal{S})$ (summand $k^{\text{exist}}_{\mathcal{WS}-\mathcal{WS}}$). We group the vertices of $\mathcal{W} \cup \mathcal{S}$ into clusters and add edges between vertices in each cluster; the summand $k^{\text{all}}_{\mathcal{WS}-\mathcal{WS}}$ corresponds to the cost of this operation when all the clusters are of the same size (and the number of edges is minimum possible, due to the convexity of function $t \to \binom{t}{2}$). Finally, in summands $k^{\text{save}}_{\mathcal{W}-\mathcal{W}}$ and $k^{\text{save}}_{\mathcal{W}-\mathcal{S}}$ we count how many edges are removed and then added again in this process: $k^{\text{save}}_{\mathcal{W}-\mathcal{W}}$ corresponds to saving three edges from each 6-cycle in $E(\mathcal{W}, \mathcal{W})$ and $k^{\text{save}}_{\mathcal{W}-\mathcal{S}}$ corresponds to saving one edge in $E(\mathcal{W}, \mathcal{S})$ per each vertex $s^C_{\beta,\xi}$. By the described properties of clause encoding it directly follows, that a satisfying assignment can be translated into an edition set of size at most $k'$.

Having sketched the completeness proof, we would like to intuitively describe the difficulties that arise in the proof of soundness, i.e., that the existence of a $p'$-cluster graph within edition distance at most $k'$, for $p' \leq 6p$, implies that $\Phi'$ is satisfiable. If we assume that the solution behaves 'sensibly', then the minimal possible budget given for $k^{\text{all}}_{\mathcal{WS}-\mathcal{WS}}$ and the properties of clause encoding already ensure that it translates to an assignment satisfying $\Phi'$. Unfortunately, we need to argue also that the solution does not 'cheat'; the main two ways of cheating are (i) trying to merge two cliques $Q^r_\alpha$, (ii) trying to separate a vertex $s^C_{\beta,\xi}$ from all the adjacent cliques. Clearly, each of these operations is locally suboptimal, but we need to guarantee that one cheat cannot lead to a lot of further savings. For example, merging two cliques $Q^r_\alpha$ implies that some vertices $s^C_{\beta,\xi}$ may be separated from less cliques they are adjacent to, than intended.

Usually, one copes with such problems by creating several 'layers' of the budget and ensuring that all the possible savings from any cheating cannot compensate even cost of one cheat. In our setting, making cliques $Q^r_\alpha$ much bigger would solve the problem. However, then we would need to increase the budget as well and the reduction would yield a weaker lower bound. Instead, we have to provide an extremely careful bookkeeping analysis of the possible shape of the solution in order to show that, indeed, the possible gains from cheating cannot amortise the costs.

## 5 Conclusion and open questions

We gave an algorithm that solves $p$-Cluster Editing in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + n + m)$ and complemented it by a multivariate lower bound, which shows that the running time of our algorithm is asymptotically tight for all $p$ sublinear in $k$.

In our multivariate lower bound it is crucial that the cliques and clusters are arranged in groups of six. However, the drawback of this construction is that Theorem 2 settles the time complexity of $p$-Cluster Editing problem only for $p \geq 6$ (Corollary 4). It does not seem unreasonable that, for example, the 2-Cluster Editing problem, already NP-complete [34], may have enough structure to allow an algorithm with running time $\mathcal{O}(2^{o(\sqrt{k})} + n + m)$. Can we construct such an algorithm or refute its existence under ETH?

Secondly, we would like to point out an interesting link between the subexponential parameterized complexity of the problem and its approximability. When the number of clusters drops from linear to sublinear in $k$, we obtain a phase transition in parameterized complexity from exponential to subexponential. As far as approximation is concerned, we know that bounding the number of clusters by a constant allows us to construct a PTAS [24], whereas the general problem is APX-hard [13]. The mutual drop of the parameterized complexity of a problem — from exponential to subexponential — and of approximability — from APX-hardness to admitting a PTAS — can be also observed for many hard problems when the input is constrained by additional topological bounds, for instance excluding a fixed pattern as a minor [17, 18, 22]. It is therefore an interesting question, whether $p$-Cluster Editing also admits a PTAS when the number of clusters is bounded by a non-constant, yet sublinear function of $k$, for instance $p = \sqrt{k}$.

## Acknowledgements

We thank Christian Komusiewicz for pointing us to the recent results on Cluster Editing [7, 31] and his thesis [30]. Moreover, we thank Pål Grønås Drange, M. S. Ramanujan and Saket Saurabh for helpful discussions.

## References

1 Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: ranking and clustering. In *Proc. of STOC'05*, pages 684–693. ACM, 2005.

2 Noga Alon, Daniel Lokshtanov, and Saket Saurabh. Fast FAST. In *Proc. of ICALP'09*, volume 5555 of *Lecture Notes in Comput. Sci.*, pages 49–58. Springer, 2009.

3 Noga Alon, Konstantin Makarychev, Yury Makarychev, and Assaf Naor. Quadratic forms on graphs. In *Proc. of STOC'05*, pages 486–493. ACM, 2005.

4 Sanjeev Arora, Eli Berger, Elad Hazan, Guy Kindler, and Muli Safra. On non-approximability for quadratic programs. In *Proc. of FOCS'05*, pages 206–215. IEEE Computer Society, 2005.

5 Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

6 Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

7 Sebastian Böcker. A golden ratio parameterized algorithm for cluster editing. In *Proc. of IWOCA'11*, pages 85–95, 2011.

8 Sebastian Böcker, Sebastian Briesemeister, Quang Bao Anh Bui, and Anke Truß. A fixed-parameter approach for weighted cluster editing. In *Proc. of APBC'08*, volume 6 of *Advances in Bioinformatics and Computational Biology*, pages 211–220, 2008.

9 Sebastian Böcker, Sebastian Briesemeister, and Gunnar W. Klau. Exact algorithms for cluster editing: Evaluation and experiments. *Algorithmica*, 60(2):316–334, 2011.

10 Sebastian Böcker and Peter Damaschke. Even faster parameterized cluster deletion and cluster editing. *Inf. Process. Lett.*, 111(14):717–721, 2011.

11 Hans L. Bodlaender, Michael R. Fellows, Pinar Heggernes, Federico Mancini, Charis Papadopoulos, and Frances A. Rosamond. Clustering with partial information. *Theor. Comput. Sci.*, 411(7-9):1202–1211, 2010.

12 Yixin Cao and Jianer Chen. Cluster editing: Kernelization based on edge cuts. In *Proc. of IPEC'10*, volume 6478 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 2010.

13 Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. In *Proc. of FOCS'03*, pages 524–533. IEEE Computer Society, 2003.

14 Moses Charikar and Anthony Wirth. Maximizing quadratic programs: Extending Grothendieck's inequality. In *Proc. of FOCS'04*, pages 54–60. IEEE Computer Society, 2004.

15 Jianer Chen and Jie Meng. A $2k$ kernel for the cluster editing problem. *Journal of Computer and System Sciences*, 78(1):211 – 220, 2012.

16 Peter Damaschke. Fixed-parameter enumerability of cluster editing and related problems. *Theory Comput. Syst.*, 46(2):261–283, 2010.

17 Erik D. Demaine, Fedor V. Fomin, Mohammadtaghi Hajiaghayi, and Dimitrios M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and $H$-minor-free graphs. *Journal of the ACM*, 52(6):866–893, 2005.

18 Erik D. Demaine and Mohammadtaghi Hajiaghayi. Bidimensionality: New connections between FPT algorithms and PTASs. In *Proc. of SODA'05*, pages 590–601, 2005.

19 R. G. Downey and M. R. Fellows. *Parameterized complexity*. Springer-Verlag, New York, 1999.

20 Michael R. Fellows, Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. Graph-based data clustering with overlaps. *Discrete Optimization*, 8(1):2–17, 2011.

21 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin, 2006.

22 Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, and Saket Saurabh. Bidimensionality and EPTAS. In *Proc. of SODA'11*, pages 748–759. SIAM, 2011.

23 Fedor V. Fomin and Yngve Vilanger. Subexponential parameterized algorithm for minimum fill-in. In *Proc. of SODA'12*, pages 1737–1746. SIAM, 2012.

24 Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. In *Proc. of SODA'06*, pages 1167–1176. ACM Press, 2006.

25 Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory Comput. Syst.*, 38(4):373–392, 2005.

26 Jiong Guo. A more effective linear kernelization for cluster editing. *Theor. Comput. Sci.*, 410(8-10):718–726, 2009.

27 Jiong Guo, Iyad A. Kanj, Christian Komusiewicz, and Johannes Uhlmann. Editing graphs into disjoint unions of dense clusters. *Algorithmica*, 61(4):949–970, 2011.

28 Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, and Johannes Uhlmann. A more relaxed model for graph-based data clustering: s-plex cluster editing. *SIAM J. Discrete Math.*, 24(4):1662–1683, 2010.

29 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63(4):512–530, 2001.

30 Christian Komusiewicz. *Parameterized Algorithmics for Network Analysis: Clustering & Querying*. PhD thesis, Technische Universität Berlin, 2011. Available at `http://fpt.akt.tu-berlin.de/publications/diss-komusiewicz.pdf`.

31 Christian Komusiewicz and Johannes Uhlmann. Alternative parameterizations for cluster editing. In *Proc. of SOFSEM'11*, volume 6543 of *Lecture Notes in Computer Science*, pages 344–355. Springer, 2011.

32 Dániel Marx. What's next? future directions in parameterized complexity. In Hans L. Bodlaender, Rod Downey, Fedor V. Fomin, and Dániel Marx, editors, *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *Lecture Notes in Computer Science*, pages 469–496. Springer, 2012.

33 Fábio Protti, Maise Dantas da Silva, and Jayme Luiz Szwarcfiter. Applying modular decomposition to parameterized cluster editing problems. *Theory Comput. Syst.*, 44(1):91–104, 2009.

34 Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.