Journal of Computer and System Sciences 80 (2014) 1430–1447

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Tight bounds for parameterized complexity of Cluster Editing with a small number of clusters ☆

Fedor V. Fomin [a], Stefan Kratsch [b], Marcin Pilipczuk [c], Michał Pilipczuk [a,*], Yngve Villanger [a]

[a] Department of Informatics, University of Bergen, Norway
[b] Technical University Berlin, Germany
[c] Institute of Informatics, University of Warsaw, Poland

ARTICLE INFO

Article history:
Received 3 May 2013
Received in revised form 3 January 2014
Accepted 8 April 2014
Available online 16 April 2014

Keywords:
Cluster editing
Correlation clustering
Parameterized complexity
Subexponential-time algorithms
Exponential-time hypothesis

ABSTRACT

In the Cluster Editing problem, also known as Correlation Clustering, we are given an undirected n-vertex graph G and a positive integer k. The task is to decide if G can be transformed into a cluster graph, i.e., a disjoint union of cliques, by changing at most k adjacencies, i.e. by adding/deleting at most k edges. We give a subexponential-time parameterized algorithm that in time $2^{\mathcal{O}(\sqrt{pk})} + n^{\mathcal{O}(1)}$ decides whether G can be transformed into a cluster graph with exactly p cliques by changing at most k adjacencies. Our algorithmic findings are complemented by the following tight lower bound on the asymptotic behavior of our algorithm. We show that unless ETH fails, for any constant $0 < \sigma \leq 1$, 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 G can be transformed into a cluster graph with at most p cliques by changing at most k adjacencies.

1. Introduction

Cluster editing, also known as *clustering with qualitative information* or *correlation clustering*, is the problem to cluster objects based only on the qualitative information concerning similarity between their pairs. 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 [1,3,4,15,16,29,39].

☆ A preliminary version of this work [26] has appeared in the Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science, STACS 2013. The first and the fourth authors have received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement No. 267959. This work has been done while the second author was at the Utrecht University, the Netherlands, and supported by the Dutch Research Foundation (NWO). The third author is supported by the National Science Centre grant N206 567140 and the Foundation For Polish Science.

* Corresponding author.
E-mail addresses: fomin@ii.uib.no (F.V. Fomin), stefan.kratsch@tu-berlin.de (S. Kratsch), malcin@mimuw.edu.pl (M. Pilipczuk), michal.pilipczuk@ii.uib.no (M. Pilipczuk), yngve.villanger@ii.uib.no (Y. Villanger).

http://dx.doi.org/10.1016/j.jcss.2014.04.015
0022-0000/

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 the minimum size. Constant factor approximation algorithms for this problem were obtained in [1,5,15]. On the negative side, the problem is known to be NP-complete [39] and, as was shown by Charikar, Guruswami, and Wirth [15], also APX-hard.

Giotis and Guruswami [29] 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 [29] 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 [7,9–12,18,22,30,32,33,35,38]; in particular, there is a recent survey of Böcker and Baumbach [8]. The problem is solvable in time $\mathcal{O}(1.62^k + |V(G)| + |E(G)|)$ [7] and it has a kernel with $2k$ vertices [14,17] (see Section 2 for the definition of a kernel). On the negative side, Komusiewicz and Uhlmann [35] have shown that existence of a *subexponential-time parameterized algorithm*, i.e., with running time $2^{o(k)} \cdot |V(G)|^{\mathcal{O}(1)}$, would contradict the Exponential Time Hypothesis of Impagliazzo et al. [34].

As for the p-Cluster Editing problem, Shamir et al. [39] have shown that the problem is NP-complete for every fixed $p \geq 2$. A kernel with $(p+2)k + p$ vertices was given by Guo [31].

Our results 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. 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 [34]. For more information about this "optimality program", we refer to a survey of Lokshtanov et al. [36] and to an appropriate section of the recent survey of Marx [37].

The complexity class SUBEPT defined by Flum and Grohe [23, Chapter 16] comprises all parameterized problems that are solvable in subexponential parameterized time, i.e., in time $2^{o(k)} \cdot n^{\mathcal{O}(1)}$ for inputs of length n and parameter k. 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 [2,19]. 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 [28].

Based on the similarities with problems on tournaments, it had been conjectured by Cao and Chen [13] that Cluster Editing also belongs to SUBEPT. Unfortunately, this conjecture has been recently disproved by Komusiewicz and Uhlmann [35]: Cluster Editing does not admit a $2^{o(k)} \cdot |V(G)|^{\mathcal{O}(1)}$ algorithm unless ETH fails. We remark that in our study we have obtained the same result independently. Our reduction is very similar in principles to the one given by Komusiewicz and Uhlmann, however the graph in the obtained instance of Cluster Editing has maximum degree 5, instead of 6 as is the case in [35]. Consequently, our reduction shows that Cluster Editing cannot be solved in subexponential time even on graphs of maximum degree 5. We believe that this improvement is of minor importance, and hence we refrain from presenting this result in order not to reiterate already published material. An interested reader is invited to the arXiv version [25] of this work for details of our reduction.

It is therefore an interesting question whether stipulating the target number of clusters can lead to a better time complexity, as was the case for the approximation viewpoint. In this work we answer this question in affirmative, and we extend our study to show a tight multivariate analysis of the p-Cluster Editing problem. Our main algorithmic result is the following.

Theorem 1. *The p-Cluster Editing problem can be solved in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + |V(G)| + |E(G)|)$.*

By Theorem 1, if $p = o(k)$ then p-Cluster Editing can be solved in $2^{o(k)} \cdot |V(G)|^{\mathcal{O}(1)}$ time, and thus it belongs to SUBEPT. 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 editing 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 the minimum cost.

We would like to remark that p-Cluster Editing can be also solved in worse time complexity $\mathcal{O}((pk)^{\mathcal{O}(\sqrt{pk})} + |V(G)| + |E(G)|)$ 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 $|V(G)|^{(2r+1)p}|V(G)|^{2k/r}p^{2k/r}$ subcases. By pipelining it with the kernelization of Guo [31] and with simple reduction rules that ensure that $p \leq 6k$ (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 $\mathcal{O}(2^{\mathcal{O}(p\sqrt{k}\log p)} + |V(G)| + |E(G)|)$. 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 for proving Theorem 1 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.

We complement Theorem 1 with a lower bound 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. *There is a polynomial-time algorithm that, given positive integers p and k and a 3-CNF formula Φ with n variables and m clauses, such that $k, n \geq p$ and $n, m \leq \sqrt{pk}$, computes a graph G and integer k', such that $k' = \mathcal{O}(k)$, $|V(G)| = \mathcal{O}(\sqrt{pk})$, and*

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

As the statement of Theorem 2 may look technical, we gather its two main consequences in Theorems 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 the harder p-Cluster Editing problem. 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 formulas would violate ETH [34].

Theorem 3. *Unless ETH fails, for every $0 \leq \sigma \leq 1$ there is a function $p(k) \in \Theta(k^{\sigma})$ such that $p_{\leq}$-Cluster Editing restricted to instances where $p = p(k)$ is not solvable in time $2^{o(\sqrt{pk})} \cdot |V(G)|^{\mathcal{O}(1)}$.*

Proof. Assume we are given a 3-CNF formula Φ with n variables and m clauses. If $n < m$, then $\lceil (m-n)/2 \rceil$ times perform the following operation: add three new variables x, y and z, and clause $(x \vee y \vee z)$. In this way we preserve the satisfiability of Φ, increase the size of Φ at most by a constant multiplicative factor, and ensure that $n \geq m$.

Take now $k = \lceil n^{\frac{2}{1+\sigma}} \rceil$ and $p = \lceil n^{\frac{2\sigma}{1+\sigma}} \rceil$. As $n \geq m$ and $0 \leq \sigma \leq 1$, we have $k, n \geq p$ and $n, m \leq \sqrt{pk}$, but $n + m = \Omega(\sqrt{pk})$. Invoke Theorem 2 for the formula Φ and parameters p and k, obtaining a graph G and a parameter $k' = \mathcal{O}(k)$. Note that $6p \in \Theta(k^{\sigma})$. Apply the assumed algorithm for the $p_{\leq}$-Cluster Editing problem to the instance $(G, 6p, k')$. In this way we resolve the satisfiability of Φ in time $2^{o(\sqrt{pk})} \cdot (n+m)^{\mathcal{O}(1)} = 2^{o(n+m)}$, which contradicts ETH. □

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

Proof. We prove the theorem for $p = 6$; the claim for larger values of p can be proved easily by taking the graph obtained in the reduction and introducing additional $p - 6$ cliques of its size.

Assume we are given a 3-CNF formula Φ with n variables and m clauses. Take $k = \max(n, m)^2$, invoke Theorem 2 for the formula Φ and parameters 1 and k, obtaining a graph G and a parameter $k' = \mathcal{O}(k)$. Note that $|V(G)| = \mathcal{O}(\sqrt{k})$. Apply the assumed algorithm for the $p_{\leq}$-Cluster Editing problem to the instance $(G, 6, k')$. In this way we resolve the satisfiability of Φ in time $2^{o(\sqrt{k})} \cdot (n+m)^{\mathcal{O}(1)} = 2^{o(n+m)}$, contradicting ETH. □

Note that Theorems 2 and 3 are formally incomparable with the result of Komusiewicz and Uhlmann [35] that the general Cluster Editing problem does not admit a subexponential-time parameterized algorithm. Clearly, by Theorem 1, the reduction proving this statement 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 edits to be extracted.

Organization of the paper In Section 2 we establish the notation and recall classic notions and results that will be used throughout the paper. Section 3 contains the description of the subexponential algorithm for p-Cluster Editing, i.e., the proof of Theorem 1. Section 4 is devoted to the multivariate lower bound, i.e., the proof of Theorem 2. In Section 5 we gather some concluding remarks and propositions for further work.

2. Preliminaries

We denote by $G = (V, E)$ a finite, undirected, and simple graph with vertex set $V(G) = V$ and edge set $E(G) = E$. For a nonempty subset $W \subseteq V$, the subgraph of G induced by W is denoted by $G[W]$. We say that a vertex set $W \subseteq V$ is *connected* if $G[W]$ is connected. The *open neighborhood* of a vertex v is $N(v) = \{u \in V : uv \in E\}$ and the *closed neighborhood* is $N[v] = N(v) \cup \{v\}$. For a vertex set $W \subseteq V$ we put $N(W) = \bigcup_{v \in W} N(v) \setminus W$ and $N[W] = N(W) \cup W$. 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 $E_G(X, Y)$ we denote the set of edges in G that have one endpoint in X and the second one in Y; whenever G is clear from the context, we drop the subscript.

By somehow abusing the notation, we will sometimes use notation uv both to denote edges and non-edges, that is, uv is then simply an unordered pair of different vertices of the given graph. Moreover, for the sake of simplicity we will often treat sets of ordered pairs as sets of unordered pairs. Thus, for instance, for vertex sets X, Y by $X \times Y$ we denote the set of all xy such that $x \in X$ and $y \in Y$. The formal meaning of these terms will be always clear from the context.

A parameterized problem Π is a subset of $\Gamma^* \times \mathbb{N}$ for some finite alphabet Γ. An instance of a parameterized problem consists of (x, k), where x is the input and k is an integer parameter that is supposed to resemble the hardness of input x. We say that a parameterized problem is *fixed-parameter tractable* (*FPT*) if it can be solved by an algorithm working in time $f(k) \cdot |x|^{\mathcal{O}(1)}$, where f is an arbitrary computable function of k. This framework can be naturally generalized to handle multiple parameters instead of just one. We are usually interested in designing FPT algorithms that have as low dependency of the running time on the parameter as possible. We refer to the book of Downey and Fellows [21] for further reading on parameterized complexity.

A *kernelization algorithm* for a parameterized problem $\Pi \subseteq \Gamma^* \times \mathbb{N}$ is an algorithm that, given an instance $(x, k) \in \Gamma^* \times \mathbb{N}$, 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 $(x', k') \in \Pi$, and $|x'|, k' \leq g(k)$ for some computable function g. In our algorithm we need the following result of Guo [31].

Theorem 5. *(See [31].) p-Cluster Editing admits a kernel with $(p+2)k + p$ vertices. More precisely, there exists an algorithm that, given an instance (G, p, k) of p-Cluster Editing, runs in time $\mathcal{O}(|V(G)| + |E(G)|)$ and outputs an equivalent instance (G', p', k') of p-Cluster Editing such that $k' \leq k$, $p' \leq p$, and $|V(G')| \leq (p'+2)k' + p'$.*

We remark that Guo [31] does not mention explicitly that the algorithm can only decrease the values of p and k, but this can be easily checked by examining the single reduction rule presented in [31].

The following lemma is used in our lower bound. Its proof is almost identical to the proof of Lemma 1 in [31], and we provide it here for the reader's convenience.

Lemma 6. *Let $G = (V, E)$ be an undirected graph and $X \subseteq V$ be a set of vertices such that $G[X]$ is a clique and each vertex in X has the same set of neighbors outside X (i.e., $N_G[v] = N_G[X]$ for each $v \in X$). Let $F \subseteq V \times V$ be a set such that $G \triangle F$ is a cluster graph where the vertices of X are in at least two different clusters. Then there exists $F' \subseteq V \times V$ such that: (i) $|F'| < |F|$, (ii) $G \triangle F'$ is a cluster graph with no larger number of clusters than $G \triangle F$, (iii) in $G \triangle F'$ the clique $G[X]$ is contained in one cluster.*

Proof. For a vertex $v \in X$, let $F(v) = \{u \notin X : vu \in F\}$. Note that, since $N_G[v] = N_G[X]$ for all $v \in X$, we have $F(v) = F(w)$ for every pair of vertices $v, w \in X$ that belong to the same cluster in $G \triangle F$.

Let Z be the vertex set of a cluster in $G \triangle F$ such that there exists $v \in Z \cap X$ with the smallest $|F(v)|$. Construct F' as follows: take F, and for each $w \in X$ replace all elements of F incident with w with $\{uw : u \in F(v)\}$. In other words, we

modify F by moving all vertices of $X \setminus Z$ to the cluster Z. Clearly, $G \triangle F'$ is a cluster graph, X is contained in one cluster in $G \triangle F'$ and $G \triangle F'$ contains no more clusters than $G \triangle F$. To finish the proof, we need to show that $|F'| < |F|$. The sets F and F' contain the same set of elements not incident with X. As $|F(v)|$ was minimum possible, for each $w \in X$ we have $|F(w)| \geq |F'(w)|$. As X was split between at least two connected components of $G \triangle F$, F contains at least one edge of $G[X]$, whereas F' does not. We infer that $|F'| < |F|$ and the lemma is proven. □

3. A subexponential algorithm for p-Cluster Editing

In this section we prove Theorem 1, that is, we show an $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + |V(G)| + |E(G)|)$-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 [31] (Theorem 5), followed by some additional preprocessing rules that ensure that $p \leq 6k$. These additional rules are encapsulated in the following lemma; the rest of this subsection is devoted to its proof.

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 $p' \leq 6k$.*

Before we proceed to the formal argumentation, let us provide some intuition. 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 already at the beginning as isolated cliques (i.e., connected components of the graph that are cliques). 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. In fact, the techniques we employ to prove Lemma 7 are quite standard in the context of Cluster Editing, even though the proof itself is lengthy. Therefore, at a first read we suggest skipping the remainder of this section and proceeding directly to Section 3.2, where we begin to explain the core of our algorithm.

We now proceed to the formal proof of Lemma 7. Let us fix some optimal solution F, i.e., a subset of $V \times V$ of the minimum cardinality such that $G \triangle F$ is a p-cluster graph.

Consider the case when $p > 6k$. Observe that only $2k$ out of p resulting clusters in $G \triangle F$ can be adjacent to any pair from the set F. Hence at least $p - 2k$ clusters must be already present in the graph G as connected components being cliques. Therefore, if G contains less than $p - 2k$ connected components that are cliques, then (G, p, k) is a NO-instance.

Rule 1. If G contains less than $p - 2k$ connected components that are cliques, answer NO.

Since $p > 6k$, if Rule 1 was not triggered, then we have more than $4k$ connected components that are cliques. The goal now is to apply greedy arguments to identify a component that can be safely assumed to be untouched. As a first step, consider a situation when G contains more than $2k$ isolated vertices. Then at least one of these vertices is not incident to an element of F, thus we may delete one isolated vertex and decrease p by one.

Rule 2. If G contains $2k + 1$ isolated vertices, pick one of them, say v, and delete it from G. The new instance is $(G \setminus v, p - 1, k)$.

We are left with the case where G contains more than $2k$ connected components that are cliques, but not isolated vertices. At least one of these cliques is untouched by F. Note that even though the number of cliques is large, some of them may be merged with other cliques (to decrease the number of connected components), or split into more cliques (to increase the number of connected components), and we have no a priori knowledge about which clique will be left untouched. We argue that in both cases, we can greedily merge or split the smallest possible clique. Thus, without loss of generality, we can assume that the largest connected component of G that is a clique is left untouched in $G \triangle F$. We reduce the input instance (G, p, k) by deleting this clique and decreasing p by one.

Rule 3. If G contains $2k + 1$ isolated cliques that are not isolated vertices, pick a clique C of the largest size and delete it from G. The new instance is $(G \setminus C, p - 1, k)$.

We formally verify safeness of Rule 3 by proving the following lemma. Without loss of generality, we may assume that the solution F, among all solutions of the minimum cardinality, has the minimum possible number of edits incident to the connected components of G that are cliques of the largest size.

Lemma 8. *Let $D_1, D_2, \ldots, D_\ell$ be connected components of G that are cliques, but not isolated vertices. Assume that $\ell \geq 2k+1$. Then there exists a component D_i that has the largest size among $D_1, D_2, \ldots, D_\ell$ and none of the pairs from F is incident to any vertex of D_i.*

Proof. Let $C_1, C_2, \ldots, C_p$ be clusters of $G \triangle F$. We say that cluster C_i *contains* component D_j if $V(D_j) \subseteq V(C_i)$, and component D_j *contains* cluster C_i if $V(C_i) \subseteq V(D_j)$. Moreover, we say that these containments are *strict* if $V(D_j) \subsetneq V(C_i)$ or $V(C_i) \subsetneq V(D_j)$, respectively.

Claim 1. *For every cluster C_i and component D_j, either $V(C_i) \cap V(D_j) = \emptyset$, C_i contains D_j or D_j contains C_i.*

Proof. We need to argue that the situation when sets $V(C_i) \cap V(D_j)$, $V(C_i) \setminus V(D_j)$, $V(D_j) \setminus V(C_i)$ are simultaneously nonempty is impossible. Assume otherwise, and without loss of generality assume further that $|V(C_i) \setminus V(D_j)|$ is the largest possible. As $V(D_j) \setminus V(C_i) \neq \emptyset$, take some $C_{i'} \neq C_i$ such that $V(C_{i'}) \cap V(D_j) \neq \emptyset$. By the choice of C_i we have that $|V(C_i) \setminus V(D_j)| \geq |V(C_{i'}) \setminus V(D_j)|$ (note that $V(C_{i'}) \setminus V(D_j)$ is possibly empty). Consider a new cluster graph H obtained from $G \triangle F$ by moving $V(C_i) \cap V(D_j)$ from the cluster C_i to the cluster $C_{i'}$. Clearly, H still has p clusters as $V(C_i) \setminus V(D_j)$ is nonempty. Moreover, the editing set F' that comprises edits that need to be applied to obtain H from G, differs from F as follows: it additionally contains $(V(C_i) \cap V(D_j)) \times (V(C_{i'}) \setminus V(D_j))$, but does not contain $(V(C_i) \cap V(D_j)) \times (V(C_i) \setminus V(D_j))$ nor $(V(C_i) \cap V(D_j)) \times (V(C_{i'}) \cap V(D_j))$. As $|V(C_i) \setminus V(D_j)| \geq |V(C_{i'}) \setminus V(D_j)|$, we have that

$$\left|\left(V(C_i) \cap V(D_j)\right) \times \left(V(C_{i'}) \setminus V(D_j)\right)\right| \leq \left|\left(V(C_i) \cap V(D_j)\right) \times \left(V(C_i) \setminus V(D_j)\right)\right|$$

and

$$\left|\left(V(C_i) \cap V(D_j)\right) \times \left(V(C_{i'}) \cap V(D_j)\right)\right| > 0.$$

Hence $|F'| < |F|$, which is a contradiction with the minimality of F. This settles Claim 1. □

We say that a component D_j is *embedded* if some cluster C_i strictly contains it. Moreover, we say that a component D_j is *broken* if it strictly contains some cluster; Claim 1 implies that then $V(D_j)$ is the union of vertex sets of the clusters it strictly contains. Component D_j is said to be *untouched* if none of the pairs from F is incident to a vertex from D_j. Claim 1 proves that every cluster is either embedded, broken or untouched.

Claim 2. *It is impossible that some component D_j is broken and some other $D_{j'}$ is embedded.*

Proof. Assume, otherwise, that some component D_j is broken and some other $D_{j'}$ is embedded. Let C_{i_1}, C_{i_2} be any two clusters contained in D_j and let $C_{i'}$ be the cluster that strictly contains $D_{j'}$. Consider a new cluster graph H obtained from $G \triangle F$ by merging clusters C_{i_1}, C_{i_2} and splitting cluster $C_{i'}$ into clusters on vertex sets $V(C_{i'}) \setminus V(D_{j'})$ and $V(D_{j'})$. As $V(C_{i'}) \setminus V(D_{j'}) \neq \emptyset$, H is still a p-cluster graph. Moreover, the editing set F' that comprises edits that need to be applied to obtain H from G, differs from F by not containing $V(C_{i_1}) \times V(C_{i_2})$ and $(V(C_{i'}) \setminus V(D_{j'})) \times V(D_{j'})$. Both of these sets are nonempty, so $|F'| < |F|$, which is a contradiction with minimality of F. This settles Claim 2. □

Claim 2 implies that either none of the components is broken, or none is embedded. We firstly prove that in the first case the lemma holds. Note that as $\ell > 2k$, at least one component D_j is untouched.

Claim 3. *If none of the components $D_1, D_2, \ldots, D_\ell$ is broken, then there is an untouched component D_j with the largest number of vertices among $D_1, D_2, \ldots, D_\ell$.*

Proof. Assume, otherwise, that all the components with the largest number of vertices are not untouched, hence they are embedded. Take any such component D_j and let $D_{j'}$ be any untouched component; by the assumption we infer that $|V(D_j)| > |V(D_{j'})|$. Let C_i be the cluster that strictly contains D_j and let $C_{i'}$ be the cluster corresponding to the (untouched) component $D_{j'}$. Consider a cluster graph H obtained from $G \triangle F$ by exchanging sets $V(D_j)$ and $V(D_{j'})$ between clusters C_i and $C_{i'}$. Observe that the editing set F' that comprises edits that need to be applied to obtain H from G, differs from F by not containing $(V(C_i) \setminus V(D_j)) \times V(D_j)$ but containing $(V(C_i) \setminus V(D_j)) \times V(D_{j'})$. However, $|V(D_j)| > |V(D_{j'})|$ and $|V(C_i) \setminus V(D_j)| > 0$, so $|F'| < |F|$. This contradicts the minimality of F and settles Claim 3. □

We are left with the case when all the clusters are broken or untouched.

Claim 4. *If none of the components $D_1, D_2, \ldots, D_\ell$ is embedded, then there is an untouched component D_j with the largest number of vertices among $D_1, D_2, \ldots, D_\ell$.*

Proof. Assume, otherwise, that all the components with the largest number of vertices are not untouched, hence they are broken. Take any such component D_j and let $D_{j'}$ be any untouched component; by the assumption we infer that

$|V(D_j)| > |V(D_{j'})|$. Assume that D_j is broken into $q+1$ clusters ($q \geq 1$) of sizes $a_1, a_2, \ldots, a_{q+1}$, where $\sum_{i=1}^{q+1} a_i = |V(D_j)|$. The number of edits needed inside component D_j is hence equal to

$$\binom{|V(D_j)|}{2} - \sum_{i=1}^{q+1} \binom{a_i}{2} \geq \binom{|V(D_j)|}{2} - \binom{|V(D_j)|-q}{2} - q\binom{1}{2} = \binom{|V(D_j)|}{2} - \binom{|V(D_j)|-q}{2}.$$

The inequality follows from the convexity of the function $t \to \binom{t}{2}$. We now consider two cases.

Assume first that $|V(D_{j'})| > q$. Let us modify the editing set F into F' by altering edits inside components D_j and $D_{j'}$ as follows: instead of breaking D_j into $q+1$ components and leaving $D_{j'}$ untouched, leave D_j untouched and break $D_{j'}$ into $q+1$ components by creating q singleton clusters and one cluster of size $|V(D_{j'})| - q$. Similar calculations to the ones presented in the paragraph above show that the editing cost inside components D_j and $D_{j'}$ is equal to $\binom{|V(D_{j'})|}{2} - \binom{|V(D_{j'})|-q}{2} < \binom{|V(D_j)|}{2} - \binom{|V(D_j)|-q}{2}$. Hence, we can obtain the same number of clusters with a strictly smaller editing set, a contradiction with the minimality of F.

Assume now that $|V(D_{j'})| \leq q$. Let us modify the editing set F into F' by altering edits inside components D_j and $D_{j'}$ as follows: instead of breaking D_j into $q+1$ components and leaving $D_{j'}$ untouched, we break $D_{j'}$ totally into $|V(D_{j'})|$ singleton clusters and break D_j into $q - |V(D_{j'})| + 1$ singleton clusters and one of size $|V(D_j)| - q + |V(D_{j'})| - 1$. Clearly, we get the same number of clusters in this manner. Similar calculations as before show that the number of new edits needed inside components D_j and $D_{j'}$ is equal to $\binom{|V(D_{j'})|}{2} + \binom{|V(D_j)|}{2} - \binom{|V(D_j)|-q+|V(D_{j'})|-1}{2}$; it may be readily checked that this value is always not larger than $\binom{|V(D_j)|}{2} - \binom{|V(D_j)|-q}{2}$ for $|V(D_{j'})| \geq 1$ and $|V(D_j)| \geq q+1$. Recall now that components $D_1, D_2, \ldots, D_\ell$ are not independent vertices, so $|V(D_{j'})| \geq 2$ and F' is obtained by removing from F some edits that were incident to the vertex set of D_j, and inserting at most the same number of edits, out of which at least one is incident only to vertices of $D_{j'}$. Hence, we can obtain the same number of clusters using a not larger editing set that has a smaller number of edits incident to components of G that are cliques of the largest size. This contradicts the choice of F.

We have obtained a contradiction in both cases, so Claim 4 follows. □

Claims 3 and 4 imply the thesis of the lemma. □

Clearly, an instance on which none of Rules 1–3 may be triggered, has $p \leq 6k$. This proves Lemma 7.

3.2. Bounds on binomial coefficients

In the running time analysis we need some combinatorial bounds on binomial coefficients. More precisely, we use the following inequality.

Lemma 9. *If a, b are nonnegative integers, then $\binom{a+b}{a} \leq 2^{2\sqrt{ab}}$.*

We start with the following simple fact that can be also derived from the standard entropy lemma [24, Lemma 3.13]. For convenience we provide a short self-contained proof.

Lemma 10. *If a, b are positive integers, then $\binom{a+b}{a} \leq \frac{(a+b)^{a+b}}{a^a b^b}$.*

Proof. Consider the following random experiment. Take a universe U containing $a+b$ elements, and select a random subset X of U by including each element of U in X independently at random with probability $\frac{a}{a+b}$. Consider now the event that $|X| = a$, and observe that

$$\mathbb{P}\big(|X| = a\big) = \binom{a+b}{a} \cdot \left(\frac{a}{a+b}\right)^a \cdot \left(\frac{b}{a+b}\right)^b.$$

Since $\mathbb{P}(|X| = a) \leq 1$, we have that $\binom{a+b}{a} \cdot (\frac{a}{a+b})^a \cdot (\frac{b}{a+b})^b \leq 1$, which is equivalent to the claim. □

We proceed to the proof of Lemma 9.

Proof of Lemma 9. Firstly, observe that the claim is trivial for $a = 0$ or $b = 0$; hence, we can assume that $a, b > 0$. Moreover, without losing generality assume that $a \leq b$. Let us denote $\sqrt{ab} = \ell$ and $\frac{a}{b} = t$, then $0 < t \leq 1$. By Lemma 10 we have that

$$\binom{a+b}{a} \le \frac{(a+b)^{a+b}}{a^a b^b} = \left(1+\frac{b}{a}\right)^a \cdot \left(1+\frac{a}{b}\right)^b$$
$$= \left[\left(1+\frac{1}{t}\right)^{\frac{\sqrt{t}}{1}} \cdot \left(1+\frac{t}{1}\right)^{\frac{1}{\sqrt{t}}}\right]^\ell.$$

Let us denote $g(x) = x\ln(1+x^{-2}) + x^{-1}\ln(1+x^2)$. As $\binom{a+b}{a} \le e^{\ell \cdot g(\sqrt{t})}$, it suffices to prove that $g(x) \le 2\ln 2$ for all $0 < x \le 1$. Observe that

$$\begin{aligned} g'(x) &= \ln\bigl(1+x^{-2}\bigr) - x\cdot 2x^{-3}\cdot\frac{1}{1+x^{-2}} - x^{-2}\ln\bigl(1+x^2\bigr) + x^{-1}\cdot 2x\cdot\frac{1}{1+x^2} \\ &= \ln\bigl(1+x^{-2}\bigr) - \frac{2}{1+x^2} - x^{-2}\ln\bigl(1+x^2\bigr) + \frac{2}{1+x^2} \\ &= \ln\bigl(1+x^{-2}\bigr) - x^{-2}\ln\bigl(1+x^2\bigr). \end{aligned}$$

Let us now introduce $h : (0,1] \to \mathbb{R}$, $h(y) = g'(\sqrt{y}) = \ln(1+y^{-1}) - y^{-1}\ln(1+y)$. Then,

$$\begin{aligned} h'(y) &= -y^{-2}\cdot\frac{1}{1+y^{-1}} + y^{-2}\ln(1+y) - y^{-1}\cdot\frac{1}{1+y} \\ &= y^{-2}\ln(1+y) - \frac{2}{y+y^2}. \end{aligned}$$

We claim that $h'(y) \le 0$ for all $y \le 1$. Indeed, from the inequality $\ln(1+y) \le y$ we infer that

$$y^{-2}\ln(1+y) \le y^{-1} = \frac{2}{y+y} \le \frac{2}{y+y^2}.$$

Therefore, $h'(y) \le 0$ for $y \in (0,1]$, so $h(y)$ is non-increasing on this interval. As $h(1) = 0$, this implies that $h(y) \ge 0$ for $y \in (0,1]$, so also $g'(x) \ge 0$ for $x \in (0,1]$. This means that $g(x)$ is non-decreasing on the interval $(0,1]$, so $g(x) \le g(1) = 2\ln 2$. □

3.3. Small cuts

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

Definition 11. Let $G = (V, E)$ be an undirected graph. A partition (V_1, V_2) of V is called a *k-cut of G* if $|E(V_1, V_2)| \le k$.

Lemma 12. *k-Cuts of a graph G can be enumerated with polynomial-time delay.*

Proof. We follow a standard branching strategy. During the branching procedure we maintain two disjoint sets $A_1, A_2 \subseteq V(G)$ that represent partial decisions about the sides of a constructed k-cut (V_1, V_2). We say that a partition (V_1, V_2) *extends* a pair (A_1, A_2) if $A_1 \subseteq V_1$ and $A_2 \subseteq V_2$. Note that for a given pair (A_1, A_2), it can be checked in polynomial time whether there exists some k-cut extending it. We simply run a polynomial-time algorithm computing the minimum edge cut between A_1 and A_2 in G. If this cut contains more than k edges, then there is no k-cut extending (A_1, A_2), and otherwise the found cut actually is a k-cut extending (A_1, A_2).

Let us order $V(G)$ arbitrarily as $v_1, v_2, \ldots, v_{|V(G)|}$. We start the branching procedure with $A_1 = A_2 = \emptyset$ and examine consecutive vertices $v_1, v_2, \ldots, v_{|V(G)|}$. For each consecutive vertex v_i we branch into two subcases: we put v_i either into A_1 or into A_2. In each of the branches, we check whether there exists a k-cut of G that extends the current pair (A_1, A_2). If this is not the case, we immediately terminate the branch since it cannot result in any k-cuts found. Otherwise we pursue the branch further. Once the alignment of all the vertices is decided, we output (A_1, A_2) as the next constructed k-cut; note that the last extension check verifies that the output partition of $V(G)$ is indeed a k-cut. Since we pursue only branches for which we are sure that they can be extended to some feasible k-cut, and all the constructed k-cuts are pairwise different, we have that finding the next k-cut 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.

Lemma 13. *Let K be a cluster graph containing at most p clusters, where $p \le 6k$. 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 first establish a bound on the number

of cuts (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 9. 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 14. *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 13. □

3.4. The algorithm

Proof of Theorem 1. Let (G, p, k) be the given p-Cluster Editing instance. By making use of Theorem 5, 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 Theorem 5 gives the additional $\mathcal{O}(|V(G)| + |E(G)|)$ 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 the 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 14, 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 exactly 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 $(\bigcup_{i=1}^{j} V(C_i), \bigcup_{i=j+1}^{p} V(C_i))$ has to be a k-cut in 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 $V_1' \setminus V_1$ 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 of the form

$$\left(\left(\bigcup_{i=1}^{j} V(C_i), \bigcup_{i=j+1}^{p} V(C_i)\right), j, \mathcal{H}\left(G\left[\bigcup_{i=1}^{j} V(C_i)\right], K\left[\bigcup_{i=1}^{j} V(C_i)\right]\right)\right)$$

constitute 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, since $(\bigcup_{i=1}^{j} V(C_i), \bigcup_{i=j+1}^{p} V(C_i))$ 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 $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + |V(G)| + |E(G)|)$ time. □

4. A multivariate lower bound

This section contains the proof of Theorem 2. The proof consists of four parts. In Section 4.1 we preprocess the input formula Φ to make it more regular. Section 4.2 contains the details of the construction of the graph G. In Section 4.3 we show how to translate a satisfying assignment of Φ into a $6p$-cluster graph G_0 close to G, and we provide the reverse implication in Section 4.4.

4.1. Preprocessing of the formula

We start with a step that regularizes the input formula Φ, while increasing its size only by a constant multiplicative factor. The purpose of this step is to ensure that, when we translate a satisfying assignment of Φ 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 used in the argumentation of the soundness step.

Lemma 15. *There exists a polynomial-time algorithm that, given a 3-CNF formula Φ with n variables and m clauses and an integer p, $p \leq n$, constructs a 3-CNF formula Φ' with n' variables and m' clauses together with a partition of the variable set $\mathrm{Vars}(\Phi')$ into p parts Vars^r, $1 \leq r \leq p$, such that the following properties hold:*

(a) *Φ' is satisfiable iff Φ is;*
(b) *in Φ' every clause contains exactly three literals corresponding to different variables;*
(c) *in Φ' every variable appears exactly three times positively and exactly three times negatively;*
(d) *n' is divisible by p and, for each $1 \leq r \leq p$, $|\mathrm{Vars}^r| = n'/p$ (i.e., the variables are split evenly between the parts Vars^r);*
(e) *if Φ' is satisfiable, then there exists a satisfying assignment of $\mathrm{Vars}(\Phi')$ with the property that in each part Vars^r the numbers of variables set to true and to false are equal;*
(f) *$n' + m' = \mathcal{O}(n+m)$.*

Proof. We modify Φ while preserving satisfiability, consecutively ensuring that properties (b), (c), (d), and (e) are satisfied. Satisfaction of (f) will follow directly from the constructions used.

First, delete every clause that contains two different literals corresponding to the same variable, as they are always satisfied. Remove copies of the same literals inside clauses. Until all the clauses have at least two literals, remove every clause containing one literal, set the value of this literal so that the clause is satisfied and propagate this knowledge to the other clauses. At the end, create a new variable s and for every clause C that has two literals replace it with two clauses $C \vee s$ and $C \vee \neg s$. All these operations preserve satisfiability and at the end all the clauses consist of exactly three different literals corresponding to different variables; hence property (b) is satisfied.

Second, duplicate each clause so that every variable appears an even number of times. Introduce two new variables q, r. Take any variable x, and assume that x appears positively k^+ times and negatively k^- times. If $k^+ < k^-$, introduce clauses $(x \vee q \vee r)$ and $(x \vee \neg q \vee \neg r)$, each $\frac{k^- - k^+}{2}$ times, otherwise introduce clauses $(\neg x \vee q \vee r)$ and $(\neg x \vee \neg q \vee \neg r)$, each $\frac{k^+ - k^-}{2}$ times. These operations preserve satisfiability (as the new clauses can be satisfied by setting q to true and r to false) and, after the operation, every variable appears the same number of time positively as negatively (including the new variables q, r). Note also that the total sum of the numbers k^+, k^- through all the variables is bounded by $\mathcal{O}(m)$, and so we introduce at most $\mathcal{O}(m)$ new clauses to the formula.

Third, copy each clause three times. For each variable x, replace all appearances of the variable x with a cycle of implications in the following way. Assume that x appears $6d$ times (the number of appearances is divisible by six due to the modifications in the previous paragraph and the copying step). Introduce new variables x_i for $1 \leq i \leq 3d$, y_i for $1 \leq i \leq d$ and clauses $(\neg x_i \vee x_{i+1} \vee y_{\lceil i/3 \rceil})$ and $(\neg x_i \vee x_{i+1} \vee \neg y_{\lceil i/3 \rceil})$ for $1 \leq i \leq 3d$ (with $x_{3d+1} = x_1$). Moreover, replace each appearance of the variable x with one of the variables x_i in such a way that each variable x_i is used once in a positive literal and once in a negative one. In this manner each variable x_i and y_i is used exactly three times in a positive literal and three times in a negative one. Moreover, the new clauses form an implication cycle $x_1 \Rightarrow x_2 \Rightarrow \ldots \Rightarrow x_{3d} \Rightarrow x_1$, ensuring that all the variables x_i have equal value in any satisfying assignment of the formula. We have thus ensured that property (c) holds, while property (b) is still satisfied.

Fourth, to make n' divisible by p we first copy the entire formula three times, creating a new set of variables for each copy. In this way we ensure that the number of variables is divisible by three. Then we add new variables in triples to make the number of variables divisible by p; note that since the number of variables is divisible by 3, there exists a number b, $0 \leq b < p$, such that after introducing b triples of variables the total number of variables will be divisible by p. For each triple x, y, z of new variables, we introduce six new clauses: all possible clauses that contain one literal for each variable x, y and z except for $(x \vee y \vee z)$ and $(\neg x \vee \neg y \vee \neg z)$. Note that the new clauses are easily satisfied by setting all new variables to true, while all new variables appear exactly three times positively and three times negatively. Moreover, as initially $p \leq n$, this step increases the size of the formula only by a constant multiplicative factor.

Finally, to achieve (d) and (e) take $\Phi' = \Phi \wedge \overline{\Phi}$, where $\overline{\Phi}$ is a copy of Φ on a disjoint copy of the variable set and with all literals reversed, i.e., positive appearances are replaced by negative ones and vice versa. Of course, if Φ' is satisfiable

then Φ as well, while if Φ is satisfiable, then we can copy the assignment to the copies of variables and reverse it, thus obtaining a satisfying assignment for Φ'. Recall that before this step the number of variables was divisible by p. We can now partition the variable set into p parts, such that whenever we include a variable into one part, we include its copy in the same part as well. In order to prove that the property (e) holds, take any satisfying assignment to Φ', truncate it to $\mathrm{Vars}(\Phi)$ and copy it while reversing onto $\mathrm{Vars}(\overline{\Phi})$. □

4.2. Construction

In this section we show how to compute the graph G and the integer k' from the formula Φ' given by Lemma 15. Recall that we need to have that $k' = \mathcal{O}(k)$, $|V(G)| = \mathcal{O}(\sqrt{pk})$, and that the satisfying assignment to Φ' translates back and forth to a solution of the p-Cluster Editing instance $(G, 6p, k')$ in the sense given in the statement of Theorem 2. As Lemma 15 increases the size of the formula by a constant multiplicative factor, we have that $n', m' = \mathcal{O}(\sqrt{pk})$ and $|\mathrm{Vars}^r| = n'/p = \mathcal{O}(\sqrt{k/p})$ for $1 \le r \le p$. Since each variable of Φ' appears in exactly 6 clauses, and each clause contains 3 literals, it follows that $m' = 2n'$.

Let $L = 1000 \cdot (1 + \frac{n'}{p}) = \mathcal{O}(\sqrt{k/p})$. For each part Vars^r, $1 \le r \le p$, we create six cliques Q^r_α, $1 \le \alpha \le 6$, each of size L.

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 expensive – in the intended solution we have exactly one clique in each cluster. Let $\mathcal{Q}$ be the set of all the vertices of all the cliques Q^r_α.

For every variable $x \in \mathrm{Vars}^r$ 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 $1 \le \alpha \le 6$ and $v \in V(Q^r_\alpha)$, create edges $v w^x_{\alpha-1,\alpha}$ and $v w^x_{\alpha,\alpha+1}$ (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 $w^x_{\alpha,\alpha+1}$, $1 \le \alpha \le 6$ 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.

Let $r(x)$ be the index of the part that contains the variable x, that is, $x \in \mathrm{Vars}^{r(x)}$.

In each clause C we (arbitrarily) enumerate variables: for $1 \le \eta \le 3$, let $\mathrm{var}(C, \eta)$ be the variable in the ηth literal of C, and $\mathrm{sgn}(C, \eta) = 0$ if the ηth literal is negative and $\mathrm{sgn}(C, \eta) = 1$ otherwise. Again, arithmetics over all the indices η in the following will behave cyclically in a natural manner.

For every clause C create nine vertices: $s^C_{\beta,\xi}$ for $1 \le \beta, \xi \le 3$. The edges incident to the vertex $s^C_{\beta,\xi}$ are defined as follows:

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

Consider now a fixed vertex $s^C_{\beta,\xi}$. Observe that the cliques that are made adjacent to $s^C_{\beta,\xi}$ in the construction above are pairwise different, and they have pairwise different subscripts (but may have equal superscripts, i.e., belong to the same part). See Fig. 1 for an illustration.

Let $\mathcal{S}$ be the set of all vertices $s^C_{\beta,\xi}$ for all clauses C. If we seek a $6p$-cluster graph close to the graph G, it is reasonable to put a vertex $s^C_{\beta,\xi}$ in a cluster together with one of the cliques this vertex is attached to. If $s^C_{\beta,\xi}$ is put in a cluster together with one of the vertices $w^{\mathrm{var}(C,\eta)}_{2\beta+2\eta-3,2\beta+2\eta-2}$ for $1 \le \eta \le 3$, we do not need to cut the appropriate edge. The vertices $s^C_{\beta,1}$ verify the assignment encoded by the variable vertices $w^x_{\alpha,\alpha+1}$; the vertices $s^C_{\beta,2}$ and $s^C_{\beta,3}$ help us to make all clusters of equal size (which is helpful in the soundness argument).

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

We now define the budget k' for edge edits. To make the presentation more clear, we split this budget into few summands. Let

$$k_{\mathcal{Q}\text{-}\mathcal{Q}} = 0, \qquad k_{\mathcal{Q}\text{-}\mathcal{WS}} = \left(6n' + 36m'\right)L,$$

$$k^{\mathrm{all}}_{\mathcal{WS}\text{-}\mathcal{WS}} = 6p\binom{\frac{6n'+9m'}{6p}}{2}, \qquad k^{\mathrm{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}} = 6n' + 27m',$$

$$k^{\mathrm{save}}_{\mathcal{W}\text{-}\mathcal{W}} = 3n', \qquad k^{\mathrm{save}}_{\mathcal{W}\text{-}\mathcal{S}} = 9m',$$

and finally

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

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

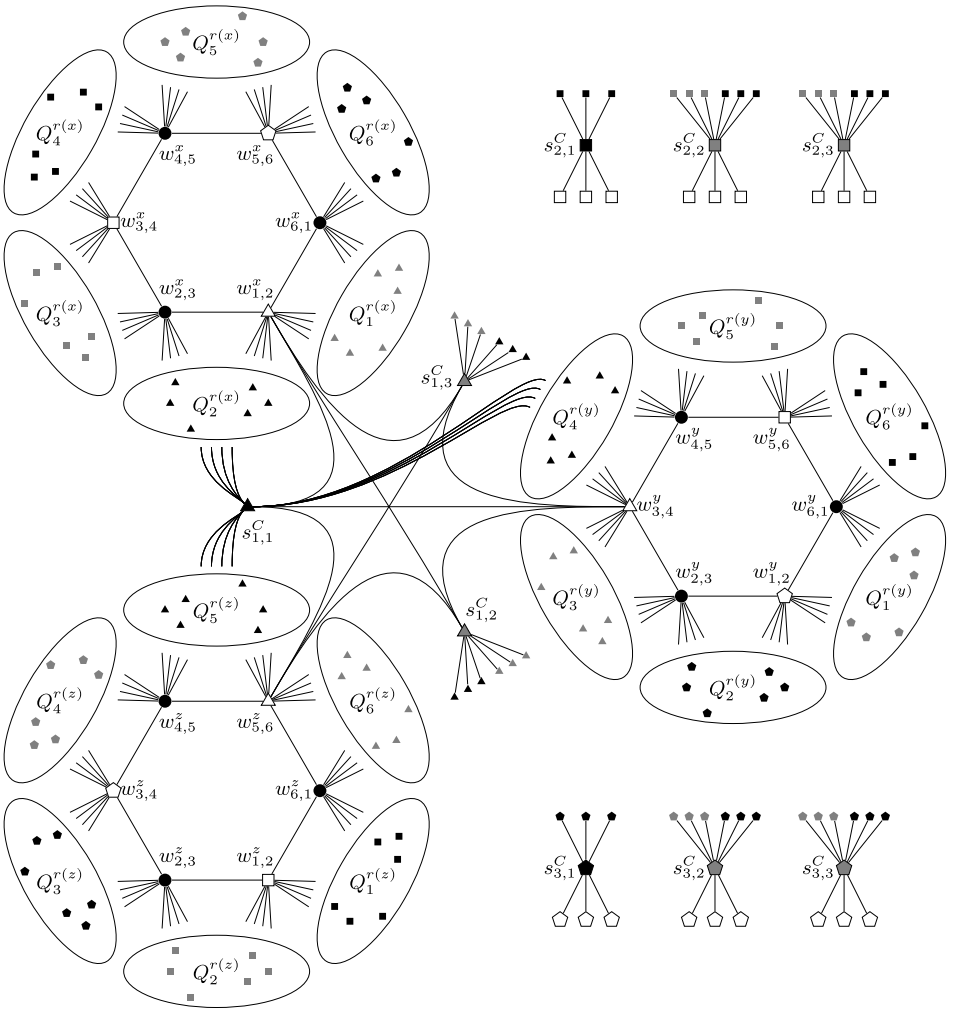

Fig. 1. A part of the graph G created for the clause $C = (\neg x \vee \neg y \vee z)$, with $\mathrm{var}(C,1) = x$, $\mathrm{var}(C,2) = y$ and $\mathrm{var}(C,3) = z$. Note that the parts $r(x)$, $r(y)$ and $r(z)$ may be not be pairwise distinct. However, due to the rotation index β, in any case for a fixed vertex $s^C_{\beta,\xi}$ the cliques this vertex is adjacent to on this figure are pairwise distinct and have pairwise distinct subscripts.

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_α, each clique is contained in its own cluster, and $k_{\mathcal{Q}\text{-}\mathcal{Q}} = 0$. For each $v \in \mathcal{W} \cup \mathcal{S}$, the vertex v is assigned to a cluster with one clique v is adjacent to; $k_{\mathcal{Q}\text{-}\mathcal{WS}}$ accumulates the cost of removal of other edges in $E(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})$. Finally, we count the edits 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 $k^{\text{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}}$). We group the vertices of $\mathcal{W} \cup \mathcal{S}$ into clusters and add edges between vertices in each cluster; the summand $k^{\text{all}}_{\mathcal{WS}\text{-}\mathcal{WS}}$ corresponds to the cost of this operation when all the clusters are of the same size (and the number of edges is minimum possible). Finally, in summands $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$ and $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$ we count how many edges are removed and then added again in this process: $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$ corresponds to saving three edges from each 6-cycle in $E(\mathcal{W}, \mathcal{W})$ and $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$ corresponds to saving one edge in $E(\mathcal{W}, \mathcal{S})$ per each vertex $s^C_{\beta,\xi}$.

4.3. Completeness

We now show how to translate a satisfying assignment of the input formula Φ into a $6p$-cluster graph close to G.

Lemma 16. *If the input formula Φ is satisfiable, then there exists a $6p$-cluster graph G_0 on vertex set $V(G)$ such that $\mathcal{H}(G, G_0) = k'$.*

Proof. Let ϕ' be a satisfying assignment of the formula Φ' as guaranteed by Lemma 15. Recall that in each part Vars^r, the assignment ϕ' sets the same number of variables to true as to false.

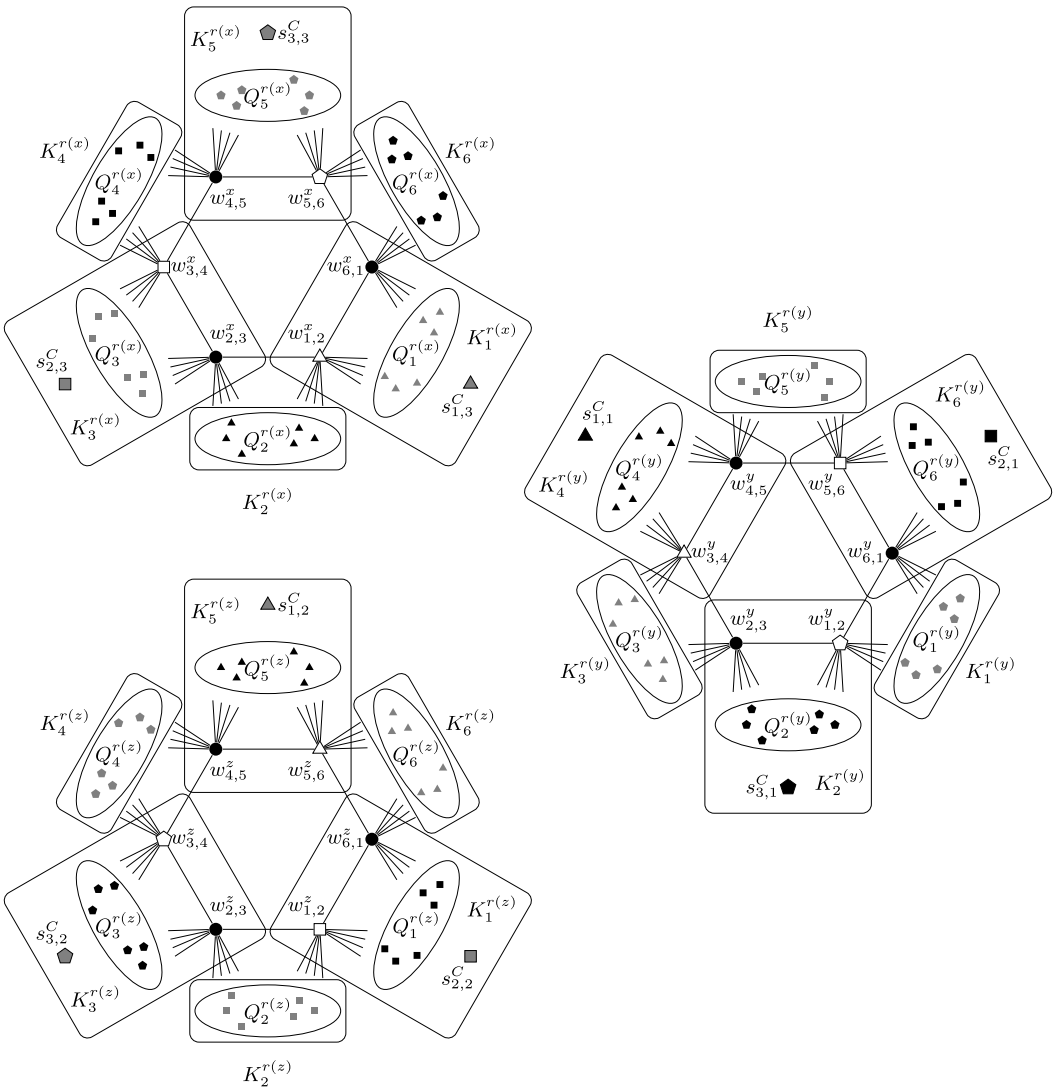

Fig. 2. Parts of clusters for variables x, y and z with $\phi'(x) = 1$, $\phi'(y) = 0$, $\phi'(z) = 1$, and a clause $C = (\neg x \vee \neg y \vee z)$ with $\mathrm{var}(C, 1) = x$, $\mathrm{var}(C, 2) = y$, $\mathrm{var}(C, 3) = z$ and $\eta(C) = 2$ (note that both y and z satisfy C in the assignment ϕ', but y was chosen as a representative).

To simplify the presentation, we identify the range of ϕ' with integers: $\phi'(x) = 0$ if x is evaluated to false in ϕ' and $\phi'(x) = 1$ otherwise. Moreover, for a clause C by $\eta(C)$ we denote the index of an arbitrarily chosen literal that satisfies C in the assignment ϕ'.

We create $6p$ clusters K_α^r, $1 \le r \le p$, $1 \le \alpha \le 6$, as follows:

- $Q_\alpha^r \subseteq K_\alpha^r$ for $1 \le r \le p$, $1 \le \alpha \le 6$;
- for $x \in \mathrm{Vars}(\Phi')$, if $\phi'(x) = 1$ then $w_{6,1}^x, w_{1,2}^x \in K_1^{r(x)}$, $w_{2,3}^x, w_{3,4}^x \in K_3^{r(x)}$, $w_{4,5}^x, w_{5,6}^x \in K_5^{r(x)}$;
- for $x \in \mathrm{Vars}(\Phi')$, if $\phi'(x) = 0$ then $w_{1,2}^x, w_{2,3}^x \in K_2^{r(x)}$, $w_{3,4}^x, w_{4,5}^x \in K_4^{r(x)}$, $w_{5,6}^x, w_{6,1}^x \in K_6^{r(x)}$;
- for each clause C of Φ' and each $1 \le \beta, \xi \le 3$, we assign the vertex $s_{\beta,\xi}^C$ to the cluster $K_{2\beta+2\eta-2-\phi'(\mathrm{var}(C,\eta))}^{r(\mathrm{var}(C,\eta))}$, where $\eta = \eta(C) + \xi - 1$.

Note that in this way $s_{\beta,\xi}^C$ belongs to the same cluster as its neighbor $w_{2\beta+2\eta-3,2\beta+2\eta-2}^{\mathrm{var}(C,\eta)}$, where $\eta = \eta(C) + \xi - 1$. See Fig. 2 for an illustration.

Let us now compute $\mathcal{H}(G, G_0)$. We do not need to add nor delete any edges in $G[\mathcal{Q}]$. We note that each vertex $v \in \mathcal{W} \cup \mathcal{S}$ is assigned to a cluster with one clique Q_α^r it is adjacent to. Indeed, this is non-trivial only for vertices $s_{\beta,1}^C$ for clauses C and $1 \le \beta \le 3$. Note, however, that $s_{\beta,1}^C$ is assigned to a cluster with the clique $Q_{2\beta+2\eta(C)-2-\phi'(\mathrm{var}(C,\eta(C)))}^{r(\mathrm{var}(C,\eta(C)))}$ and is adjacent

to the clique $Q^{r(\mathrm{var}(C,\eta(C)))}_{2\beta+2\eta(C)-2-\mathrm{sgn}(\mathrm{var}(C,\eta(C)))}$. Since literal with variable $\mathrm{var}(C,\eta(C))$ satisfies C in the assignment ϕ', it follows that $\phi'(\mathrm{var}(C,\eta(C))) = \mathrm{sgn}(\mathrm{var}(C,\eta(C)))$, and so $s^C_{\beta,1}$ is assigned to a cluster together with a clique it is adjacent to.

Therefore we need to cut $k_{\mathcal{Q}\text{-}\mathcal{WS}} = (6n' + 36m')L$ edges in $E(\mathcal{Q}, \mathcal{W}\cup\mathcal{S})$: L edges adjacent to each vertex $w^x_{\alpha,\alpha+1}$, $2L$ edges adjacent to each vertex $s^C_{\beta,1}$, and $5L$ edges adjacent to each vertex $s^C_{\beta,2}$ and $s^C_{\beta,3}$. We do not add any new edges between $\mathcal{Q}$ and $\mathcal{W}\cup\mathcal{S}$.

To count the number of edits in $G[\mathcal{W}\cup\mathcal{S}]$, let us first verify that the clusters K^r_α are of equal sizes. Fix a cluster K^r_α, $1 \le \alpha \le 6$, $1 \le r \le p$. K^r_α contains two vertices $w^x_{\alpha-1,\alpha}$ and $w^x_{\alpha,\alpha+1}$ for each variable x with $\phi'(x) = \alpha \bmod 2$. Since ϕ' evaluates the same number of variables in Vars^r to true as to false, we infer that each cluster K^r_α contains exactly n'/p vertices from $\mathcal{W}$, corresponding to $n'/(2p) = |\mathrm{Vars}^r|/2$ variables of Vars^r.

Now we need to verify that each cluster K^r_α contains the same number of vertices from $\mathcal{S}$. Let f be a function that maps the vertices of $\mathcal{S}$ to clusters they are belonging to. Recall that $f(s^C_{\beta,\xi}) = K^{r(\mathrm{var}(C,\eta))}_{2\beta+2\eta-2-\phi'(\mathrm{var}(C,\eta))}$, where $\eta = \eta(C) + \xi - 1$. We claim the following.

Claim 5. *Fix a part r, $1 \le r \le p$, and an index α, $1 \le \alpha \le 6$. Then to each pair (x, C), where x is a variable with $r(x) = r$ present in a clause C, and $\phi'(x) = \alpha \bmod 2$, one can assign a pair of indices (β,ξ) with $1 \le \beta, \xi \le 3$ such that $f(s^C_{\beta,\xi}) = K^r_\alpha$. Moreover, this assignment can be constructed in such a manner that for all pairs (x, C) the corresponding vertices $s^C_{\beta,\xi}$ are pairwise different.*

Proof. Let η be such that $x = \mathrm{var}(C,\eta)$. Define indices (β,ξ) as follows:

$$\xi = \eta - \eta(C) + 1,$$
$$\beta = \frac{\alpha + \phi'(x)}{2} - \eta + 1,$$

where the arithmetic operations are defined cyclically in a natural manner. Note that β is an integer since $\phi'(x) = \alpha \bmod 2$. From these equalities it follows that:

$$\eta = \eta(C) + \xi - 1,$$
$$\alpha = 2\beta + 2\eta - 2 - \phi'\big(\mathrm{var}(C,\eta)\big),$$

and so $f(s^C_{\beta,\xi}) = K^r_\alpha$. We are left with proving that triples (C,β,ξ) are pairwise different for different pairs (x, C). However, note that for different variables x appearing in C we have different indices η, and so the defined indices ξ will be different. □

Observe that for a fixed part r and index α, there are exactly $3|\mathrm{Vars}^r| = 3n'/p$ pairs (x, C) satisfying the assumption in Claim 5: there are $|\mathrm{Vars}^r|/2$ variables in Vars^r that are evaluated to $\alpha \bmod 2$ in ϕ', and each variable appears in 6 different clauses. Thus, Claim 5 ensures that the preimage under f of each cluster is of cardinality at least $3n'/p$. Since there are $6p$ clusters in total, we infer that the union of these preimages is of cardinality at least $18n' = 9m'$ (recall that $m' = 2n'$). However, we have that $|\mathcal{S}| = 9m'$. Consequently, it follows that the preimage under f of each cluster is of size exactly $3n'/p$, so each cluster contains the same number of vertices from $\mathcal{S}$.

We now count the number of edits in $G[\mathcal{W}\cup\mathcal{S}]$ as sketched in the construction section. The subgraph $G[\mathcal{W}\cup\mathcal{S}]$ contains $6n' + 27m'$ edges: one 6-cycle for each variable and three edges incident to each of the nine vertices $s^C_{\beta,\xi}$ for each clause C. Each cluster K^r_α contains n'/p vertices from $\mathcal{W}$ and $\frac{3m'}{2p}$ vertices from $\mathcal{S}$. If we deleted all edges in $G[\mathcal{W}\cup\mathcal{S}]$ and then added all the missing edges in the clusters, we would make $k^{\mathrm{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}} + k^{\mathrm{all}}_{\mathcal{WS}\text{-}\mathcal{WS}}$ edits, due to the clusters being equal-sized. However, in this manner we sometimes delete an edge and then introduce it again; thus, for each edge of $G[\mathcal{W}\cup\mathcal{S}]$ that is contained in one cluster K^r_α, we should subtract 2 in this counting scheme.

For each variable x, exactly three edges of the form $w^x_{\alpha-1,\alpha}w^x_{\alpha,\alpha+1}$ are contained in one cluster; this gives a total of $k^{\mathrm{save}}_{\mathcal{W}\text{-}\mathcal{W}} = 3n'$ saved edges. For each clause C each vertex $s^C_{\beta,\xi}$ is assigned to a cluster with one of the vertices $w^{\mathrm{var}(C,\eta)}_{2\beta+2\eta-3,2\beta+2\eta-2}$, $1 \le \eta \le 3$, thus exactly one of the edges incident to $s^C_{\beta,\xi}$ is contained in one cluster. This sums up to $k^{\mathrm{save}}_{\mathcal{W}\text{-}\mathcal{S}} = 9m'$ saved edges, and we infer that the $6p$-cluster graph G_0 can be obtained from G by exactly $k' = k_{\mathcal{Q}\text{-}\mathcal{Q}} + k_{\mathcal{Q}\text{-}\mathcal{WS}} + k^{\mathrm{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}} + k^{\mathrm{all}}_{\mathcal{WS}\text{-}\mathcal{WS}} - 2k^{\mathrm{save}}_{\mathcal{W}\text{-}\mathcal{W}} - 2k^{\mathrm{save}}_{\mathcal{W}\text{-}\mathcal{S}}$ edits. □

4.4. Soundness

We need the following simple bound on the number of edges of a cluster graph.

Lemma 17. *Let a, b be positive integers and H be a cluster graph with ab vertices and at most a clusters. Then $|E(H)| \ge a\binom{b}{2}$ and equality holds if and only if H is an a-cluster graph and each cluster of H has size exactly b.*

Proof. It suffices to note that if not all clusters of H are of size b, then there is one of size at least $b+1$ and one of size at most $b-1$ or the number of clusters is less than a. Then, moving a vertex from the largest cluster of H to a new or the smallest cluster strictly decreases the number of edges of H. □

We are ready to show how to translate a p'-cluster graph G_0 with $p' \leq 6p$ and $\mathcal{H}(G_0, G) \leq k'$, into a satisfying assignment of the input formula Φ.

Lemma 18. *If there exists a p'-cluster graph G_0 with $V(G) = V(G_0)$, $p' \leq 6p$, and $\mathcal{H}(G, G_0) \leq k'$, then the input formula Φ is satisfiable.*

Proof. By Lemma 6, we may assume that each clique Q_α^r is contained in one cluster in G_0. Let $F = E(G_0) \triangle E(G)$ be the editing set, $|F| \leq k'$.

Before we start, we present some intuition. The cluster graph G_0 may differ from the one constructed in the completeness step in two significant ways, both leading to some savings in the edges incident to $\mathcal{W} \cup \mathcal{S}$ that may not be included in F. First, it may not be true that each cluster contains exactly one clique Q_α^r. However, since the number of cliques is at most $6p$, this may happen only if some clusters contain more than one clique Q_α^r, and we need to add L^2 edges to merge each pair of cliques that belong to the same cluster. Second, a vertex $v \in \mathcal{W} \cup \mathcal{S}$ may not be contained in a cluster together with one of the cliques it is adjacent to. However, as each such vertex needs to be separated from *all* its adjacent cliques (compared to *all but one* in the completeness step), this costs us additional L edges to remove. The large multiplicative constant in the definition of L ensures us that in both these ways we pay more than we save on the edges incident with $\mathcal{W} \cup \mathcal{S}$. We now proceed to the formal argumentation.

We define the following quantities.

$$\begin{aligned}
\ell_{\mathcal{Q}\text{-}\mathcal{Q}} &= \big|F \cap (\mathcal{Q} \times \mathcal{Q})\big|,\\
\ell_{\mathcal{Q}\text{-}\mathcal{WS}} &= \big|F \cap E_G(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})\big|,\\
\ell^{\text{all}}_{\mathcal{WS}\text{-}\mathcal{WS}} &= \big|E\big(G_0[\mathcal{W} \cup \mathcal{S}]\big)\big|,\\
\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}} &= \big|E_G(\mathcal{W}, \mathcal{W}) \cap E_{G_0}(\mathcal{W}, \mathcal{W})\big|,\\
\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}} &= \big|E_G(\mathcal{W}, \mathcal{S}) \cap E_{G_0}(\mathcal{W}, \mathcal{S})\big|.
\end{aligned}$$

Recall that $k^{\text{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}} = |E(G[\mathcal{W} \cup \mathcal{S}])| = 6n' + 27m'$. Similarly as in the completeness proof, we have that

$$|F| \geq \ell_{\mathcal{Q}\text{-}\mathcal{Q}} + \ell_{\mathcal{Q}\text{-}\mathcal{WS}} + \ell^{\text{all}}_{\mathcal{WS}\text{-}\mathcal{WS}} + k^{\text{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}} - 2\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}} - 2\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}.$$

Indeed, $\ell_{\mathcal{Q}\text{-}\mathcal{Q}}$ and $\ell_{\mathcal{Q}\text{-}\mathcal{WS}}$ count (possibly not all) pairs of F that are incident to the vertices of $\mathcal{Q}$. The edges of $F \cap ((\mathcal{W} \cup \mathcal{S}) \times (\mathcal{W} \cup \mathcal{S}))$ are counted in an indirect way: each edge of $G[\mathcal{W} \cup \mathcal{S}]$ is deleted ($k^{\text{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}}$) and each edge of $G_0[\mathcal{W} \cup \mathcal{S}]$ is added ($\ell^{\text{all}}_{\mathcal{WS}\text{-}\mathcal{WS}}$). Then, the edges that are counted twice in this manner are subtracted ($\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$ and $\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$).

We say that a cluster is *crowded* if it contains at least two cliques Q_α^r and *proper* if it contains exactly one clique Q_α^r. A clique Q_α^r that is contained in a crowded (proper) cluster is called a *crowded* (*proper*) clique.

Let a be the number of crowded cliques. Note that $\ell_{\mathcal{Q}\text{-}\mathcal{Q}} - k_{\mathcal{Q}\text{-}\mathcal{Q}} = |F \cap (\mathcal{Q} \times \mathcal{Q})| - 0 \geq aL^2/2$, since each vertex in a crowded clique needs to be connected to at least one other crowded clique.

We say that a vertex $v \in \mathcal{W} \cup \mathcal{S}$ is *attached* to a clique Q_α^r, if it is adjacent to all vertices of the clique in G. Moreover, we say that a vertex $v \in \mathcal{W} \cup \mathcal{S}$ is *alone* if it is contained in a cluster in G_0 that does not contain any clique v is attached to. Let n^{alone} be the number of alone vertices.

Let us now count the number of vertices a fixed clique Q_α^r is attached to. Recall that $|\text{Vars}^r| = n'/p$. For each variable $x \in \text{Vars}^r$ the clique Q_α^r is attached to two vertices $w^x_{\alpha-1,\alpha}$ and $w^x_{\alpha,\alpha+1}$. Moreover, each variable $x \in \text{Vars}^r$ appears in exactly six clauses: three time positively and three times negatively. For each such clause C, Q_α^r is attached to the vertex $s^C_{\beta,2}$ for exactly one choice of the value $1 \leq \beta \leq 3$ and to the vertex $s^C_{\beta,3}$ for exactly one choice of the value $1 \leq \beta \leq 3$. Moreover, if x appears in C positively and α is odd, or if x appears in C negatively and α is even, then Q_α^r is attached to the vertex $s^C_{\beta,1}$ for exactly one choice of the value $1 \leq \beta \leq 3$. We infer that the clique Q_α^r is attached to exactly fifteen vertices from $\mathcal{S}$ for each variable $x \in \text{Vars}^r$. Therefore, there are exactly $17|\text{Vars}^r| = 17n'/p$ vertices of $\mathcal{W} \cup \mathcal{S}$ attached to Q_α^r: $2n'/p$ from $\mathcal{W}$ and $15n'/p$ from $\mathcal{S}$.

Take an arbitrary vertex $v \in \mathcal{W} \cup \mathcal{S}$ and assume that v is attached to b_v cliques, and a_v out of them are crowded. As F needs to contain all edges of G that connect v with cliques that belong to a different cluster than v, we infer that $|F \cap E_G(\{v\}, \mathcal{Q})| \geq (b_v - \max(1, a_v))L$. Moreover, if v is alone, then $|F \cap E_G(\{v\}, \mathcal{Q})| \geq b_v L \geq 1 \cdot L + (b_v - \max(1, a_v))L$. Hence

$$\begin{aligned}
\ell_{\mathcal{Q}\text{-}\mathcal{WS}} &= \big|F \cap E_G(\mathcal{Q}, \mathcal{W} \cup \mathcal{S})\big|\\
&\geq n^{\text{alone}} L + \sum_{v \in \mathcal{W} \cup \mathcal{S}} \big(b_v - \max(1, a_v)\big)L\\
&\geq n^{\text{alone}} L + \sum_{v \in \mathcal{W} \cup \mathcal{S}} (b_v - 1)L - \sum_{v \in \mathcal{W} \cup \mathcal{S}} a_v L.
\end{aligned}$$

Recall that $\sum_{v\in\mathcal{W}\cup\mathcal{S}}(b_v-1)L = k_{\mathcal{Q}\text{-}\mathcal{W}\mathcal{S}}$. Therefore, using the fact that each clique is attached to exactly $17n'/p$ vertices of $\mathcal{W}\cup\mathcal{S}$, we obtain that

$$\begin{aligned}\ell_{\mathcal{Q}\text{-}\mathcal{W}\mathcal{S}} - k_{\mathcal{Q}\text{-}\mathcal{W}\mathcal{S}} &= \big|F\cap E_G(\mathcal{Q},\mathcal{W}\cup\mathcal{S})\big| - k_{\mathcal{Q}\text{-}\mathcal{W}\mathcal{S}}\\ &\geq n^{\text{alone}}L - \sum_{v\in\mathcal{W}\cup\mathcal{S}} a_v L = n^{\text{alone}}L - 17aLn'/p.\end{aligned}$$

In G_0, the vertices of $\mathcal{W}\cup\mathcal{S}$ are split between $p'\leq 6p$ clusters and there are $6n'+9m'$ of them. By Lemma 17, the minimum number of edges of $G_0[\mathcal{W}\cup\mathcal{S}]$ is attained when all clusters are of equal size and the number of clusters is maximum possible. We infer that $\ell^{\text{all}}_{\mathcal{W}\mathcal{S}\text{-}\mathcal{W}\mathcal{S}}\geq k^{\text{all}}_{\mathcal{W}\mathcal{S}\text{-}\mathcal{W}\mathcal{S}}$.

We are left with bounding $\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$ and $\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$. Recall that $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$ counts three edges out of each 6-cycle constructed per variable of Φ', $|k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}| = 3n'$, whereas $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$ counts one edge per each vertex $s^C_{\beta,\xi}\in\mathcal{S}$, $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}} = 9m' = |\mathcal{S}|$.

Consider a crowded cluster K with $c>1$ crowded cliques. We say that K *interferes* with a vertex $v\in\mathcal{W}\cup\mathcal{S}$ if v is attached to a clique in K. As each clique is attached to exactly $17n'/p$ vertices of $\mathcal{W}\cup\mathcal{S}$, $2n'/p$ belonging to $\mathcal{W}$ and $15n'/p$ to $\mathcal{S}$, in total at most $2an'/p$ vertices of $\mathcal{W}$ and at most $15an'/p$ vertices of $\mathcal{S}$ interfere with a crowded cluster.

Fix a variable $x\in\text{Vars}(\Phi')$. If none of the vertices $w^x_{\alpha,\alpha+1}\in\mathcal{W}$ interferes with any crowded cluster K, then all the cliques $Q^{r(x)}_{\alpha'}$, $1\leq\alpha'\leq 6$, are proper cliques, each contained in a different cluster in G_0. Moreover, if additionally no vertex $w^x_{\alpha,\alpha+1}$, $1\leq\alpha\leq 6$, is alone, then in the 6-cycle constructed for the variable x at most three edges are not in F. On the other hand, if some of the vertices $w^x_{\alpha,\alpha+1}\in\mathcal{W}$ interfere with a crowded cluster K, or at least one of them is alone, it may happen that all six edges of this 6-cycle are contained in one cluster of G_0. The total number of 6-cycles that contain either alone vertices or vertices interfering with crowded clusters is bounded by $n^{\text{alone}}+an'/p$, as every clique is attached to exactly n'/p 6-cycles. In $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$ we counted three edges per a 6-cycle, while in $\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$ we counted at most three edges per every 6-cycles except 6-cycles that either contain alone vertices or vertices attached to crowded cliques, for which we counted at most six edges. Hence, we infer that

$$\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}} - k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}} \leq 3\big(n^{\text{alone}}+an'/p\big).$$

We claim that if a vertex $s^C_{\beta,\xi}\in\mathcal{S}$ (i) is not alone, and (ii) is not attached to a crowded clique, and (iii) is not adjacent to any alone vertex in $\mathcal{W}$, then at most one edge from $E(\{s^C_{\beta,\xi}\},\mathcal{W})$ may not be in F. Recall that $s^C_{\beta,\xi}$ has exactly three neighbors in $\mathcal{W}$, each of them attached to exactly two cliques and all these six cliques are pairwise distinct; moreover, $s^C_{\beta,\xi}$ is attached only to these six cliques, if $\beta=2,3$, or only to three out of these six, if $\beta=1$. Observe that (i) and (ii) imply that $s^C_{\beta,\xi}$ is in the same cluster as exactly one of the six cliques attached to his neighbors in $\mathcal{W}$, so if it was in the same cluster as two of his neighbors in $\mathcal{W}$, then at least one of them would be alone, contradicting (iii). Therefore, if conditions (i), (ii), and (iii) are satisfied, then at most one edge adjacent to $s^C_{\beta,\xi}$ may be not contained in F. However, if at least one of (i), (ii) or (iii) is not satisfied, then all three edges incident to $s^C_{\beta,\xi}$ may be contained in one cluster. As each vertex in $\mathcal{W}$ is adjacent to at most 18 vertices in $\mathcal{S}$ (at most 3 per every clause in which the variable is present), there are at most $18n^{\text{alone}}$ vertices $s^C_{\beta,\xi}$ that are alone or adjacent to an alone vertex in $\mathcal{W}$. Note also that the number of vertices of $\mathcal{S}$ interfering with crowded clusters is bounded by $15an'/p$, as each of a crowded cliques has exactly $15n'/p$ vertices of $\mathcal{S}$ attached. Thus, we are able to bound the number of vertices of $\mathcal{S}$ for which (i), (ii) or (iii) does not hold by $18n^{\text{alone}}+15an'/p$. As in $k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$ we counted one edge per every vertex of $\mathcal{S}$, while in $\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$ we counted at most one edge per every vertex of $\mathcal{S}$ except for vertices not satisfying (i), (ii), or (iii), for which we counted at most three edges, we infer that

$$\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}} - k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}} \leq 2\big(18n^{\text{alone}}+15an'/p\big).$$

Summing up all the bounds:

$$\begin{aligned}|F|-k' &\geq (\ell_{\mathcal{Q}\text{-}\mathcal{Q}}-k_{\mathcal{Q}\text{-}\mathcal{Q}}) + (\ell_{\mathcal{Q}\text{-}\mathcal{W}\mathcal{S}}-k_{\mathcal{Q}\text{-}\mathcal{W}\mathcal{S}}) + \big(\ell^{\text{all}}_{\mathcal{W}\mathcal{S}\text{-}\mathcal{W}\mathcal{S}}-k^{\text{all}}_{\mathcal{W}\mathcal{S}\text{-}\mathcal{W}\mathcal{S}}\big) - 2\big(\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}-k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}\big)\\ &\quad - 2\big(\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}-k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}\big)\\ &\geq aL^2/2 + n^{\text{alone}}L - 17aLn'/p + 0 - 6\big(n^{\text{alone}}+an'/p\big) - 4\big(18n^{\text{alone}}+15an'/p\big)\\ &= a\cdot\big(L^2/2 - 17Ln'/p - 66n'/p\big) + n^{\text{alone}}\cdot(L-78)\\ &\geq a + n^{\text{alone}}.\end{aligned}$$

To see that the last inequality holds, observe that both the terms $L^2/2-17Ln'/p-66n'/p$ and $L-78$ are at least 1. This follows from the choice of the value of L, $L=1000\cdot(1+\frac{n'}{p})$; note that in particular $L\geq 1000$.

We infer that $a=0$, that is, each clique Q^r_α is contained in a different cluster of G_0, and each cluster of G_0 contains exactly one such clique. Moreover, $n^{\text{alone}}=0$, that is, each vertex $v\in\mathcal{W}\cup\mathcal{S}$ is contained in a cluster with at least one

clique v is attached to; as all cliques are proper, v is contained in a cluster with exactly one clique v is attached to and $\ell_{\mathcal{Q}\text{-}\mathcal{WS}} = k_{\mathcal{Q}\text{-}\mathcal{WS}}$.

Recall that $|F \cap ((\mathcal{W} \cup \mathcal{S}) \times (\mathcal{W} \cup \mathcal{S}))| = \ell^{\text{all}}_{\mathcal{WS}\text{-}\mathcal{WS}} + k^{\text{exist}}_{\mathcal{WS}\text{-}\mathcal{WS}} - 2\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}} - 2\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$. As each clique is now proper and no vertex is alone, for each variable x at most three edges out of the 6-cycle $w^x_{\alpha,\alpha+1}$, $1 \le \alpha \le 6$, are not in F, that is, $\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}} \le k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{W}}$. Moreover, for each vertex $s^C_{\beta,\xi} \in \mathcal{S}$, the three neighbors of $s^C_{\beta,\xi}$ are contained in different clusters and at most one edge incident to $s^C_{\beta,\xi}$ is not in F, that is, $\ell^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}} \le k^{\text{save}}_{\mathcal{W}\text{-}\mathcal{S}}$. As $|F| \le k'$ and $\ell^{\text{all}}_{\mathcal{WS}\text{-}\mathcal{WS}} \ge k^{\text{all}}_{\mathcal{WS}\text{-}\mathcal{WS}}$, these inequalities are tight: exactly three edges out of each 6-cycle are not in F, and exactly one edge adjacent to a vertex in $\mathcal{S}$ is not in F.

Consider an assignment ϕ' of $\text{Vars}(\Phi')$ that assigns $\phi'(x) = 1$ if the vertices $w^x_{\alpha,\alpha+1}$, $1 \le \alpha \le 6$ are contained in clusters with cliques $Q^{r(x)}_1$, $Q^{r(x)}_3$, and $Q^{r(x)}_5$ (i.e., the edges $w^x_{6,1}w^x_{1,2}$, $w^x_{2,3}w^x_{3,4}$ and $w^x_{4,5}w^x_{5,6}$ are not in F), and $\phi'(x) = 0$ otherwise (i.e., if the vertices $w^x_{\alpha,\alpha+1}$, $1 \le \alpha \le 6$ are contained in clusters with cliques $Q^{r(x)}_2$, $Q^{r(x)}_4$ and $Q^{r(x)}_6$) – a direct check shows that these are the only ways to save 3 edges inside a 6-cycle. We claim that ϕ' satisfies Φ', which implies that Φ is also satisfiable. Consider a clause C. The vertex $s^C_{1,1}$ is contained in a cluster with one of the three cliques it is attached to (as $n^{\text{alone}} = 0$), say $Q^r_{\alpha'}$, and with one of the three vertices of $\mathcal{W}$ it is adjacent to, say $w^x_{\alpha,\alpha+1}$. Therefore $r(x) = r$, $w^x_{\alpha,\alpha+1}$ is contained in the same cluster as $Q^r_{\alpha'}$, and it follows that the literal in C that contains x satisfies C in the assignment ϕ'. □

5. Conclusion and open questions

In this work we have given an algorithm that solves p-Cluster Editing in time $\mathcal{O}(2^{\mathcal{O}(\sqrt{pk})} + |V(G)| + |E(G)|)$ 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 $p \ge 6$ (Theorem 4). It does not seem unreasonable that, for example, the 2-Cluster Editing problem, already NP-complete [39], may have enough structure to allow an algorithm with running time $2^{o(\sqrt{k})} \cdot |V(G)|^{\mathcal{O}(1)}$. 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 [29], whereas the general problem is APX-hard [15]. 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 [19,20,27]. 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 $p = \sqrt{k}$.

Acknowledgments

We thank Christian Komusiewicz for pointing us to the recent results on Cluster Editing [7,35]. We also thank an anonymous referee for suggesting the current, elegant proof of Lemma 10. Moreover, we thank Pål Grønås Drange, M.S. Ramanujan and Saket Saurabh for helpful discussions.

References

[1] Nir Ailon, Moses Charikar, Alantha Newman, Aggregating inconsistent information: ranking and clustering, J. ACM 55 (5) (2008), Article No. 23, 27 pp.
[2] Noga Alon, Daniel Lokshtanov, Saket Saurabh, Fast FAST, in: Proceedings of the 36th International Colloquium on Automata, Languages and Programming, ICALP 2009, in: Lect. Notes Comput. Sci., vol. 5555, Springer, 2009, pp. 49–58.
[3] Noga Alon, Konstantin Makarychev, Yury Makarychev, Assaf Naor, Quadratic forms on graphs, in: Proceedings of the 37th ACM Symposium on Theory of Computing, STOC 2005, ACM, 2005, pp. 486–493.
[4] Sanjeev Arora, Eli Berger, Elad Hazan, Guy Kindler, Muli Safra, On non-approximability for quadratic programs, in: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2005, IEEE Computer Society, 2005, pp. 206–215.
[5] Nikhil Bansal, Avrim Blum, Shuchi Chawla, Correlation clustering, Mach. Learn. 56 (2004) 89–113.
[6] Amir Ben-Dor, Ron Shamir, Zohar Yakhini, Clustering gene expression patterns, J. Comput. Biol. 6 (3–4) (1999) 281–297.
[7] Sebastian Böcker, A golden ratio parameterized algorithm for cluster editing, J. Discrete Algorithms 16 (2012) 79–89.
[8] Sebastian Böcker, Jan Baumbach, Cluster Editing, in: CiE, in: Lect. Notes Comput. Sci., vol. 7921, Springer, 2013, pp. 33–44.
[9] Sebastian Böcker, Sebastian Briesemeister, Quang Bao, Anh Bui, Anke Truß, A fixed-parameter approach for weighted Cluster Editing, in: Proceedings of the 6th Asia-Pacific Bioinformatics Conference, APBC 2008, in: Adv. Bioinform. Comput. Biol., vol. 6, 2008, pp. 211–220.
[10] Sebastian Böcker, Sebastian Briesemeister, Gunnar W. Klau, Exact algorithms for Cluster Editing: evaluation and experiments, Algorithmica 60 (2) (2011) 316–334.
[11] Sebastian Böcker, Peter Damaschke, Even faster parameterized Cluster Deletion and Cluster Editing, Inf. Process. Lett. 111 (14) (2011) 717–721.
[12] Hans L. Bodlaender, Michael R. Fellows, Pinar Heggernes, Federico Mancini, Charis Papadopoulos, Frances A. Rosamond, Clustering with partial information, Theor. Comput. Sci. 411 (7–9) (2010) 1202–1211.
[13] Yixin Cao, Jianer Chen, Cluster editing: kernelization based on edge cuts, in: IPEC, in: Lect. Notes Comput. Sci., vol. 6478, Springer, 2010, pp. 60–71.

[14] Yixin Cao, Jianer Chen, Cluster editing: kernelization based on edge cuts, Algorithmica 64 (1) (2012) 152–169.
[15] Moses Charikar, Venkatesan Guruswami, Anthony Wirth, Clustering with qualitative information, J. Comput. Syst. Sci. 71 (3) (2005) 360–383.
[16] Moses Charikar, Anthony Wirth, Maximizing quadratic programs: extending Grothendieck's inequality, in: Proceedings of the 45th Symposium on Foundations of Computer Science, FOCS 2004, IEEE Computer Society, 2004, pp. 54–60.
[17] Jianer Chen, Jie Meng, A 2k kernel for the Cluster Editing problem, J. Comput. Syst. Sci. 78 (1) (2012) 211–220.
[18] Peter Damaschke, Fixed-parameter enumerability of Cluster Editing and related problems, Theory Comput. Syst. 46 (2) (2010) 261–283.
[19] Erik D. Demaine, Fedor V. Fomin, MohammadTaghi Hajiaghayi, Dimitrios M. Thilikos, Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs, J. ACM 52 (6) (2005) 866–893.
[20] Erik D. Demaine, MohammadTaghi Hajiaghayi, Bidimensionality: new connections between FPT algorithms and PTASs, in: Proceedings of the 16th Symposium on Discrete Algorithms, SODA 2005, 2005, pp. 590–601.
[21] Rodney G. Downey, Michael R. Fellows, Parameterized Complexity, Springer-Verlag, New York, 1999.
[22] Michael R. Fellows, Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, Johannes Uhlmann, Graph-based data clustering with overlaps, Discrete Optim. 8 (1) (2011) 2–17.
[23] Jörg Flum, Martin Grohe, Parameterized Complexity Theory, Texts Theor. Comp. Sci. EATCS Ser., Springer-Verlag, Berlin, 2006.
[24] Fedor V. Fomin, Dieter Kratsch, Exact Exponential Algorithms, Texts Theor. Comp. Sci. EATCS Ser., Springer, 2010.
[25] Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, Yngve Villanger, Subexponential fixed-parameter tractability of Cluster Editing, CoRR, abs/1112.4419, 2011.
[26] Fedor V. Fomin, Stefan Kratsch, Marcin Pilipczuk, Michał Pilipczuk, Yngve Villanger, Tight bounds for parameterized complexity of cluster editing, in: STACS, in: LIPIcs, vol. 20, Schloss Dagstuhl – Leibniz-Zentrum fuer Informatik, 2013, pp. 32–43.
[27] Fedor V. Fomin, Daniel Lokshtanov, Venkatesh Raman, Saket Saurabh, Bidimensionality and EPTAS, in: Proceedings of the 22nd Symposium on Discrete Algorithms, SODA 2011, SIAM, 2011, pp. 748–759.
[28] Fedor V. Fomin, Yngve Villanger, Subexponential parameterized algorithm for Minimum Fill-in, in: Proceedings of the 23rd Symposium on Discrete Algorithms, SODA 2012, SIAM, 2012, pp. 1737–1746.
[29] Ioannis Giotis, Venkatesan Guruswami, Correlation clustering with a fixed number of clusters, in: Proceedings of the 17th Symposium on Discrete Algorithms, SODA 2006, ACM Press, 2006, pp. 1167–1176.
[30] Jens Gramm, Jiong Guo, Falk Hüffner, Rolf Niedermeier, Graph-modeled data clustering: exact algorithms for clique generation, Theory Comput. Syst. 38 (4) (2005) 373–392.
[31] Jiong Guo, A more effective linear kernelization for Cluster Editing, Theor. Comput. Sci. 410 (8–10) (2009) 718–726.
[32] Jiong Guo, Iyad A. Kanj, Christian Komusiewicz, Johannes Uhlmann, Editing graphs into disjoint unions of dense clusters, Algorithmica 61 (4) (2011) 949–970.
[33] Jiong Guo, Christian Komusiewicz, Rolf Niedermeier, Johannes Uhlmann, A more relaxed model for graph-based data clustering: s-plex Cluster Editing, SIAM J. Discrete Math. 24 (4) (2010) 1662–1683.
[34] Russell Impagliazzo, Ramamohan Paturi, Francis Zane, Which problems have strongly exponential complexity?, J. Comput. Syst. Sci. 63 (4) (2001) 512–530.
[35] Christian Komusiewicz, Johannes Uhlmann, Cluster Editing with locally bounded modifications, Discrete Appl. Math. 160 (15) (2012) 2259–2270.
[36] Daniel Lokshtanov, Dániel Marx, Saket Saurabh, Lower bounds based on the Exponential Time Hypothesis, Bull. Eur. Assoc. Theor. Comput. Sci. 105 (2011) 41–72.
[37] Dániel Marx, What's next? Future directions in parameterized complexity, in: The Multivariate Algorithmic Revolution and Beyond, in: Lect. Notes Comput. Sci., vol. 7370, Springer, 2012, pp. 469–496.
[38] Fábio Protti, Maise Dantas da Silva, Jayme Luiz Szwarcfiter, Applying modular decomposition to parameterized cluster editing problems, Theory Comput. Syst. 44 (1) (2009) 91–104.
[39] Ron Shamir, Roded Sharan, Dekel Tsur, Cluster graph modification problems, Discrete Appl. Math. 144 (1–2) (2004) 173–182.