Modification-Fair Cluster Editing

Vincent Froese, Leon Kellerhals, Rolf Niedermeier

Technische Universität Berlin, Faculty IV, Algorithmics and Computational Complexity, Berlin, Germany.
{vincent.froese, leon.kellerhals}@tu-berlin.de

Abstract

The classic CLUSTER EDITING problem (also known as CORRELATION CLUSTERING) asks to transform a given graph into a disjoint union of cliques (clusters) by a small number of edge modifications. When applied to vertex-colored graphs (the colors representing subgroups), standard algorithms for the NP-hard CLUSTER EDITING problem may yield solutions that are biased towards subgroups of data (e.g., demographic groups), measured in the number of modifications incident to the members of the subgroups. We propose a modification fairness constraint which ensures that the number of edits incident to each subgroup is proportional to its size. To start with, we study MODIFICATION-FAIR CLUSTER EDITING for graphs with two vertex colors. We show that the problem is NP-hard even if one may only *insert* edges *within* a subgroup; note that in the classic "non-fair" setting, this case is trivially polynomial-time solvable. However, in the more general *editing* form, the modification-fair variant remains fixed-parameter tractable with respect to the number of edge edits. We complement these and further theoretical results with an empirical analysis of our model on real-world social networks where we find that the price of modification-fairness is surprisingly low, that is, the cost of optimal modification-fair differs from the cost of optimal "non-fair" solutions only by a small percentage.

Introduction

Imagine two competing parties, the reds and the blues. The party members are nodes in a (social) network. The goal is to cluster the nodes into a set of disjoint cliques by a small number of edge modifications, that is, edge deletions or insertions. As the parties are competing against each other, they are very careful about the other party having to pay the relatively same "modification cost". That is, the average number of edits involving red vertices as edge endpoints should be the same as the average number involving blue vertices, see Figure 1 for an example. This yields a colored version of the well-studied NP-hard CLUSTER EDITING problem, where now the criterion of having relatively same modification cost becomes a *process-oriented fairness* concept which we are going to introduce in this paper.

In recent years, fairness in algorithmic problems has become a profoundly studied topic, particularly so in machine
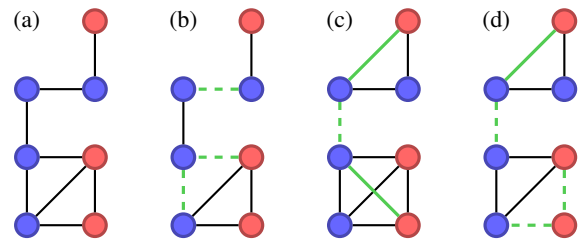
Figure 1: An exemplary graph G with blue and red vertices (a) and three modifications of G to a cluster graph (b)–(d). Added edges are marked green, deleted edges are green and dashed. (b) A modification of minimum size with five edits incident to the blues and one edit incident to the reds. So $\Delta_{\text{ed}} = |{}^5\!/_4 - {}^1\!/_3| = {}^{11}\!/_{12}$, see Eq. (1). (c) A more fair modification of minimum size ($\Delta_{\text{ed}} = {}^1\!/_2$). (d) A more fair (but larger) modification ($\Delta_{\text{ed}} = {}^1\!/_3$).

learning and related areas. Clustering problems are fundamental topics of unsupervised learning and optimization in general. In this work, we focus on graph-based data clustering, and therein on one of the most basic and best studied problems, CLUSTER EDITING. Our main conceptual contribution is to introduce a fairness concept that is not modeling the "fairness of the output" of the algorithm (in our case a disjoint set of cliques, the clusters), but rather the fairness of the generation process. In our case, this means each party should pay (proportionally to its size) roughly the same modification cost. We perform both a theoretical (algorithms and complexity) as well as an empirical study. In a nutshell, we show that our new problem MODIFICATION-FAIR CLUSTER EDITING seems computationally harder (but not hopelessly hard) than CLUSTER EDITING, but our experimental studies also indicate that the "price of fairness" (that is, how much more edge edits are needed compared to the classic, "non-fair" case) is relatively low if one does not go for perfect fairness.

Related work. For a thorough review on fairness in the context of machine learning we refer to the survey by Mehrabi et al. (2021). Closest to our work in terms of the underlying clustering problem are studies on fair CORRELATION CLUSTERING (Ahmadian et al. 2020b; Friggstad and Mousavi 2021). Both works, following the *disparate im-*

pact doctrine, focus on an output-oriented fairness, that is, the fairness is defined by looking at the resulting clusters enforcing a predefined fraction of each group within each cluster. This kind of fairness, while prudent in some scenarios, may be inapt in other contexts, e.g. political districting. Facing the NP-hardness of the problem, both works mainly study polynomial-time approximation algorithms (while we focus on exact solvability).

Chierichetti et al. (2017) were the first to study fairness in the context of clustering, studying k-median and k-center problems. Herein, the works by Abbasi, Bhaskara, and Venkatasubramanian (2021) and Ghadiri, Samadi, and Vempala (2021) are closest to our scenario as they, too, seek similar costs for every party. There are numerous further recent works studying fairness for clustering problems (Ahmadian et al. 2020a; Bandyapadhyay, Fomin, and Simonov 2021; Bandyapadhyay et al. 2021; Chakrabarty and Negahbani 2021; Mahabadi and Vakilian 2020; Vakilian and Yalçıner 2021). For a general account on classic CLUSTER EDITING, we refer to the survey of Böcker and Baumbach (2013) and note that the "PACE implementation challenge 2021" (Kellerhals et al. 2021) was dedicated to CLUSTER EDITING.

Our contributions. We introduce MODIFICATION-FAIR CLUSTER EDITING, reflecting a process-oriented fairness criterion in graph-based data clustering: instead of looking at the outcome, we consider the modification process that yields the clustering. Here we demand that the average number of modifications at a vertex is balanced among the groups (we focus on two groups). We parameterize our fairness constraint by the difference between these averages. Thus, our modification fairness can be seen as aiming at equal average *error* or *distortion* for both groups (similar in spirit as for socially fair k-means (Ghadiri, Samadi, and Vempala 2021) or fair PCA (Samadi et al. 2018)). For a formal definition of MODIFICATION-FAIR CLUSTER EDITING, we refer to the next section. On the theoretical side, we show that MODIFICATION-FAIR CLUSTER EDITING remains NP-hard even if only edge insertions are allowed (which is trivially polynomial-time solvable in the classic case). Moreover, we show the NP-hardness of very restricted cases of the general editing version (e.g., all but one vertex having the same color) and provide conditional running time lower bounds. On the positive side, we identify a polynomial-time solvable special case (disallowing to edit edges whose both endpoints have the same color) and show fixed-parameter tractability with respect to the number of edge modifications. We also introduce a useful cluster transformation problem (a simple to define algorithmic problem with a strong numerical flavor) which may be of independent interest; we employ it in our NP-hardness proofs. On the empirical side, we demonstrate that while typically computationally hard(er) to find, "fair solutions" typically seem not much more expensive than conventional ones.

Preliminaries

We assume familiarity with basic concepts of algorithms, complexity, and graph theory. A clique is a completely connected graph and by clique size we refer to its number of vertices. It is well-known that a clique cannot contain a P_3 (a path on three vertices) as an induced subgraph. A cluster graph is a disjoint union of cliques.

We investigate the well-studied CLUSTER EDITING problem (also called CORRELATION CLUSTERING) under fairness constraints regarding edge modifications. Here, we are given a graph $G = (V, E)$ with vertices $V = R \uplus B$ colored either *red* or *blue*. An *edge modification set* is a subset $S \subseteq \binom{V}{2}$, that is, a subset of the size-two vertex subsets. We define $\#\mathrm{ed}_S(v) := |\{e \in S \mid v \in e\}|$ to be the number of edge modifications incident to a vertex v (that is, the degree of v in the modification graph (V, S)). Then

$$\Delta_{\mathrm{ed}}(S) := \left| \frac{\sum_{v \in R} \#\mathrm{ed}_S(v)}{|R|} - \frac{\sum_{v \in B} \#\mathrm{ed}_S(v)}{|B|} \right| \quad (1)$$

is the absolute difference of the average numbers of modifications at a red vertex and a blue vertex.

MODIFICATION-FAIR CLUSTER EDITING

Input: An undirected graph $G = (V, E)$ with vertices colored either red or blue, $k \in \mathbb{N}$, and $\delta \in \mathbb{Q}$.

Question: Is there an edge modification set $S \subseteq \binom{V}{2}$ with $|S| \leq k$ and $\Delta_{\mathrm{ed}}(S) \leq \delta$ that transforms G into a cluster graph?

We immediately observe some simple upper bounds on Δ_{ed}.

Observation 1 ($\star$[1]). *For every edge modification set S, the following upper bounds hold: (i) $\Delta_{ed}(S) \leq |S|$, (ii) $\Delta_{ed}(S) \leq |V| - 1$, (iii) $\Delta_{ed}(S) \leq \frac{2|S|}{\min(|R|,|B|)}$.*

By Observation 1, if $\delta \geq \min(k, |V| - 1, 2k/\min(|R|,|B|))$, then MODIFICATION-FAIR CLUSTER EDITING is simply CLUSTER EDITING.

Finally, we recall some basic (parameterized) complexity concepts. A parameterized problem is *fixed-parameter tractable* if there exists an algorithm solving any instance (x, p) (x is in the input instance and p is some parameter—in our case it will be the number k of edge modifications) in $f(p) \cdot |x|^{\mathcal{O}(1)}$ time, where f is a computable function solely depending on p. The class XP contains all parameterized problems which can be solved in polynomial time if the parameter p is a constant, that is, in $f(p) \cdot |x|^{g(p)}$ time. The Exponential Time Hypothesis (ETH) claims that there exists a constant $c > 0$ such that the 3-SAT problem cannot be solved in $\mathcal{O}(2^{cn})$ time. It is used to prove conditional running time lower bounds, for example, it is known that one cannot find a clique clique of size s in an n-vertex graph in $\rho(s) \cdot n^{o(s)}$ time for any function ρ, unless the ETH fails (Chen et al. 2006).

Modification Fairness: Complexity

We explore the algorithmic complexity of MODIFICATION-FAIR CLUSTER EDITING and compare it to the standard

---

[1]Proofs of results marked with $\star$ are deferred to the full version of the paper, available at arXiv (http://arxiv.org/abs/2112.03183).

CLUSTER EDITING problem and its restrictions which either only allow edge deletions (CLUSTER DELETION) or insertions (CLUSTER COMPLETION). We find NP-hardness, ETH-based lower running time bounds, and tractability results (a polynomial-time solvable special case and fixed-parameter tractability of the general problem).

First, we show that even very special cases of MODIFICATION-FAIR CLUSTER EDITING remain NP-hard, the corresponding polynomial-time many-one reductions also leading to ETH-based running time lower bounds.

Theorem 2. MODIFICATION-FAIR CLUSTER EDITING *and* MODIFICATION-FAIR CLUSTER DELETION *are NP-hard for arbitrary* $\delta \geq 0$ *and neither solvable in* $2^{o(k)} \cdot |V|^{\mathcal{O}(1)}$ *time, in* $2^{o(|V|)}$ *time, nor in* $2^{o(|E|)}$ *time unless the Exponential Time Hypothesis fails. This also holds*

1. *if only mono-colored edge modifications are allowed or*
2. *if there is only one red vertex.*

Proof. Both cases use similar reductions, based on NP-hardness results in the "non-fair setting".

Case 1. Komusiewicz and Uhlmann (2012, Theorem 1) showed that CLUSTER EDITING is NP-hard in the case that every solution has size at least k and can be assumed to only delete edges. They also showed that this case is not solvable in $2^{o(k)} \cdot |V|^{\mathcal{O}(1)}$, $2^{o(|V|)}$, or $2^{o(|E|)}$ time assuming the ETH (Komusiewicz and Uhlmann 2012, Theorem 2). We reduce from this version of classic CLUSTER EDITING.

Let $(G = (V, E), k)$ be such an instance and let $\delta \geq 0$. We define an instance (G', k', δ) of MODIFICATION-FAIR CLUSTER EDITING as follows: The graph G' contains a copy of G where all vertices are colored blue. Additionally, G' contains $3k$ red vertices which form k disjoint P_3s. Moreover, we add $\max(|V|, 3k) - 3k$ isolated red vertices and $\max(|V|, 3k) - |V|$ isolated blue vertices to G' such that the number of red vertices equals the number of blue vertices. Finally, we set $k' := 2k$.

Clearly, the reduction can be performed in polynomial time. For the correctness, assume first that (G, k) is a *yes*-instance, that is, G can be made a cluster graph by deleting exactly k edges. Then, deleting the corresponding k edges in G' and also one arbitrary edge of each red P_3 in G' clearly yields a solution S of size $2k = k'$ with $\Delta_{\text{ed}}(S) = 0 \leq \delta$ (as G' contains the same number of red and blue vertices).

Conversely, let (G', k', δ) be a *yes*-instance. Note that every solution deletes at least one edge of each of the k red P_3s in G'. Hence, at most k edge deletions are performed to make the copy of G in G' a cluster graph.

For the ETH-based running time lower bounds, note that $k' \in \mathcal{O}(k)$, $|V(G')| \in \mathcal{O}(\max(|V|, k))$, and $|E(G')| \in \mathcal{O}(\max(|E|, k))$. Hence, the claimed lower bounds follow.

Case 2. As in the proof for Case 1, we reduce from CLUSTER EDITING with a given instance $(G = (V, E), k)$ obtained from the reduction by Komusiewicz and Uhlmann (2012, Theorem 1). We make additional use of the fact that their construction yields an instance where $|V| < 2k$.

Our instance (G', k', δ) is defined as follows. The graph G' contains a blue copy of G and one red vertex r which is connected to an arbitrary vertex x of degree six

from G (which always exists for nontrivial graphs G). We further add $2k - |V| + 1$ many isolated blue vertices such that overall G' contains $2k + 1$ blue vertices. We set $k' := k + 1$.

Clearly, the reduction can be performed in polynomial time; the correctness is shown next. The ETH-based lower bounds follow in complete analogy to Case 1. If (G, k) is a *yes*-instance with solution S, then $S' := S \cup \{\{r, x\}\}$ yields a solution for G' of size $k + 1 = k'$ with $\Delta_{\text{ed}}(S') = |\frac{1}{1} - \frac{2k+1}{2k+1}| = 0 \leq \delta$.

Conversely, if (G', k', δ) is a *yes*-instance with solution S', then $\{\{r, x\}\} \in S'$. If $\{r, x\}$ were not in S', then the six other edges incident to x would have to be deleted, which leaves a budget of $k - 5$ edge deletions for the remaining graph. However, by construction, the remaining graph has a budget lower bound of $k - 4$ (Komusiewicz and Uhlmann 2012, Theorem 1). Thus, $\{r, x\} \in S'$ and G has a solution of size at most k. $\quad\square$

Remark. The graph in the reduction of Komusiewicz and Uhlmann (2012, Theorem 1) has maximum degree six. Moreover, they also show that the maximum number of edge deletions at any vertex in a solution is four (Komusiewicz and Uhlmann 2012, Corollary 1). Hence, Case 1 of Theorem 2 also holds for input graphs of maximum degree six and at most four local edge deletions.

Surprisingly, CLUSTER COMPLETION, which is trivially solvable in polynomial time, becomes NP-hard when enforcing fairness.

Theorem 3. MODIFICATION-FAIR CLUSTER COMPLETION *is NP-hard for every constant* $\delta \geq 0$ *(also if only mono-colored edges are inserted).*

The proof is based on a polynomial-time many-one reduction from the following problem, which we will prove to be NP-hard in the following section (Theorem 6).

CLUSTER TRANSFORMATION BY EDGE ADDITION

Input: A cluster graph G and an integer k.
Question: Can G be transformed into another cluster graph by adding exactly k edges?

Proof of Theorem 3. Let $(G = (V, E), k)$ be an instance of CLUSTER TRANSFORMATION BY EDGE ADDITION. Moreover, we clearly can assume that $k \leq \binom{|V|}{2} - |E|$ since otherwise we have a trivial *no*-instance.

Let $\delta \geq 0$ be an arbitrary constant. We construct an instance (G', k', δ) as follows. The graph G' contains a blue copy of G together with $|V|$ many red vertices which form an arbitrary connected graph with $\binom{|V|}{2} - k - \lfloor \frac{|V|}{2} \delta \rfloor$ edges. To see that this is actually possible, note that $\binom{|V|}{2} - k - \lfloor \frac{|V|}{2} \delta \rfloor \geq |V|$ (for large enough $|V|$) since $k \in o(|V|^2)$ in our reduction (see Lemma 7). We set $k' := 2k + \lfloor \frac{|V|}{2} \delta \rfloor$.

Now assume that (G, k) is a *yes*-instance. Then, adding the corresponding k edges to the blue copy of G in G' and the $k + \lfloor \frac{|V|}{2} \delta \rfloor$ missing edges to the red subgraph yields a cluster graph. This set S' of added edges satisfies

$$\Delta_{\text{ed}}(S') = \frac{2(k + \lfloor \frac{|V|}{2} \delta \rfloor)}{|V|} - \frac{2k}{|V|} \leq \delta.$$

Conversely, let (G', k', δ) be a *yes*-instance. Then, the solution S' clearly contains the $k + \lfloor \frac{|V|}{2}\delta \rfloor$ missing edges of the red subgraph of G'. Let $k_b \leq k$ denote the number of added edges between blue vertices in S'. Then,

$$\Delta_{\mathrm{ed}}(S') = \frac{2(k + \lfloor \frac{|V|}{2}\delta \rfloor) + k - k_b}{|V|} - \frac{2k_b + k - k_b}{|V|}$$
$$= \frac{2k}{|V|} + \frac{2}{|V|}\lfloor \frac{|V|}{2}\delta \rfloor - \frac{2k_b}{|V|} > \delta + \frac{2(k - k_b - 1)}{|V|}.$$

Since $\Delta_{\mathrm{ed}}(S') \leq \delta$, we have $k_b = k$, so S' contains k edges in the blue copy of G in G', and (G, k) is a *yes*-instance. $\square$

We observe from the intractability results so far that the hardness of MODIFICATION-FAIR CLUSTER EDITING is rooted in finding the right mono-colored edge modifications. Indeed, when allowing only bicolored edge modifications, the problem becomes polynomial-time solvable.

Theorem 4. MODIFICATION-FAIR CLUSTER EDITING *is polynomial-time solvable when no mono-colored edge modifications are allowed.*

Proof. Let $(G = (R \uplus B, E), k, \delta)$ be an input instance. Then, $G[R]$ and $G[B]$ have to be cluster graphs, otherwise we have a *no*-instance. Also, since mono-colored edge modifications are forbidden, we have $\Delta_{\mathrm{ed}}(S) = |S| \cdot |\frac{1}{|R|} - \frac{1}{|B|}|$ for any solution S. Thus, we may assume that $k \leq \delta / |\frac{1}{|R|} - \frac{1}{|B|}|$.

Let $R_1, \ldots, R_r$ and $B_1, \ldots, B_b$ be the vertex sets of the clusters in $G[R]$, respectively $G[B]$. Since mono-colored edge modifications are forbidden, a solution can never merge two blue or two red clusters into one. Thus, any solution either isolates a cluster, or merges it with exactly one cluster of the other color. This can be modeled as a matching in a complete bipartite graph H with vertices $v_1, \ldots, v_r$ and $w_1, \ldots, w_b$, where a matching edge indicates which clusters are merged. Clearly, every cluster editing solution corresponds to a matching and vice versa. Let $E' \subseteq E$ be the edges between R and B and let $E_{ij} \subseteq E'$ denote the edges between R_i and B_j. For a given matching M in H, the size of the corresponding solution can then be written as

$$|E'| - \sum_{\{v_i, w_j\} \in M} |E_{ij}| + \sum_{\{v_i, w_j\} \in M} (|R_i||B_j| - |E_{ij}|).$$

Hence, the solution size is minimized by a matching that maximizes $\sum_{\{v_i, w_j\} \in M} (2|E_{ij}| - |R_i||B_j|)$. Thus, we can solve the problem by computing a maximum-weight matching in H with edge weights $w(\{v_i, w_j\}) := (2|E_{ij}| - |R_i||B_j|)$ in polynomial time (Lovász and Plummer 2009). $\square$

We next show fixed-parameter tractability for the number k of edge modifications. Our approach is as follows. We first run the well-known P_3-branching algorithm (Cai 1996) to obtain a cluster graph. As the resulting solution need not be modification-fair, we may need to do further edge modifications. For this, we first apply polynomial-time data reduction rules which shrink the graph size to a polynomial in k, and then brute-force on the reduced graph.

Theorem 5. MODIFICATION-FAIR CLUSTER EDITING *is solvable in* $2^{\mathcal{O}(k \log k)} \cdot (|V| + |E|)$ *time.*

Proof. Let $(G = (V, E), k, \delta)$ be an instance of MODIFICATION-FAIR CLUSTER EDITING. We first apply the standard P_3-branching algorithm for CLUSTER EDITING to enumerate all minimal cluster edge modification sets S of size at most k in $\mathcal{O}(3^k(|V| + |E|))$ time (Cai 1996). For each S, we check whether $\Delta_{\mathrm{ed}}(S) \leq \delta$. If not, then we try to extend S to a fair edge modification set. Clearly, each fair edge modification set of size at most k contains at least one of the enumerated edge modification sets. Note that in order to check later that our modification set is fair, we store the original numbers $|B|$ and $|R|$ of blue and red vertices in G.

For each S, we first apply the following three data reduction rules to the cluster graph G' obtained from S.

1. If there is a clique with more than $k + 1$ vertices, then delete it.
2. If there are more than $2k$ isolated vertices of the same color which have not been touched by S, then delete one of them.
3. Let $2 \leq s \leq k + 1$ and $0 \leq t \leq s$. If there are more than k cliques with s vertices, t of which are blue, and none of them are touched by S, then delete one of them.

Note that we keep all cliques with at most $k + 1$ in G' which contain an endpoint of an edge in S. Clearly, there are at most $2|S|$ such cliques.

For the correctness, note that if a clique with $\ell \geq 2$ vertices is modified, then this requires at least $\ell - 1$ edge modifications. Hence, Rule 1 is correct. Clearly, k edge modifications can touch at most $2k$ vertices of any color, so Rule 2 is correct. Rule 3 is correct as we cannot touch more than k cliques of size at least two.

For exhaustive application of the data reduction rules, we run a depth-first search and count the number of cliques with the same numbers of blue and red vertices. As we added at most k edges to obtain G', we can apply the rules in $\mathcal{O}(|V| + |E| + k)$ time. After exhaustive application, the remaining graph comprises $\mathcal{O}(k^2)$ vertices contained in cliques touched by S and $\mathcal{O}(k^3)$ vertices not touched by S.

Let $W \subseteq V$ be the vertices remaining after exhaustive application of the above data reduction rules. We now try all possible extensions $S' \subseteq \binom{W}{2} \setminus S$ of size at most $k - |S|$ and check whether the set $S^* := S \cup S'$ transforms G into a cluster graph and is fair, that is, $\Delta_{\mathrm{ed}}(S^*) \leq \delta$. There are $\mathcal{O}(k^{6k})$ such extensions; the checking can be done in $\mathcal{O}(|V| + |E| + k)$ time each. The overall running time is thus $2^{\mathcal{O}(k \log k)} \cdot (|V| + |E|)$. $\square$

We remark that the above theorem implies that MODIFICATION-FAIR CLUSTER EDITING can be solved in linear time when only constantly many edges need to be modified.

Transforming Cluster Graphs

In this section, we prove the following, most technical theorem of our work. We believe the considered problem CLUSTER TRANSFORMATION BY EDGE ADDITION (defined in

the previous section) to be of independent interest, also making connections to the area of scheduling.

Theorem 6. CLUSTER TRANSFORMATION BY EDGE ADDITION *is NP-hard.*

We devise a polynomial-time many-one reduction from the NUMERICAL 3D MATCHING problem introduced and proven to be strongly NP-hard by Garey and Johnson (1975).

NUMERICAL 3D MATCHING

Input: Positive integers t, $a_1, a_2, \ldots, a_n$, $b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_n$.

Question: Are there bijections $\alpha, \beta, \gamma \colon [n] \to [n]$ such that for each $i \in [n]$, $a_{\alpha(i)} + b_{\beta(i)} + c_{\gamma(i)} = t$?

On a high level, our reduction works as follows. We add a *small* clique for every a_i, a *medium* clique for every b_i, and a *large* clique for every c_i. By appropriate choice of our budget k, we can ensure that every clique in the resulting cluster graph $G + S$ is the result of merging one small, one medium, and one large clique. We finally show that if each cluster consists of cliques corresponding to elements a_i, b_j, and c_ℓ such that their sum is equal to the target t, then the number of required edge additions is minimized. That is, if there is a cluster that does not hit this target, then the budget k is overspent.

Construction. Let $I = (t, a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n, c_1, c_2, \ldots, c_n)$, $n \geq 3$, be an instance of NUMERICAL 3D MATCHING. As NUMERICAL 3D MATCHING is strongly NP-hard, we may assume that for all $i \in [n]$, $a_i, b_i, c_i \leq n^d$ for some constant $d > 0$. We further assume that $t > a_i, b_i, c_i$ for all $i \in [n]$ and that $\sum_{i=1}^{n}(a_i + b_i + c_i) = n \cdot t$, as otherwise I is a trivial *no*-instance.

We construct an instance $I' = (G, k)$ of CLUSTER TRANSFORMATION BY EDGE ADDITION as follows. Let $A := n^{2d}$, let $B := n^{3d}$, and let $C := n^{7d}$. For $i \in [n]$, we set $a'_i := a_i + A$, $b'_i := b_i + B$, $c'_i := c_i + C$, and add three cliques of order a'_i, b'_i, and c'_i, respectively, to G. We refer to these cliques by their order a'_i, b'_i, c'_i and call them *small*, *medium-sized*, and *large*, respectively. Finally, we set $t' := t + A + B + C$ and

$$k := n\binom{t'}{2} - |E(G)| = n\binom{t'}{2} - \sum_{i=1}^{n}\left(\binom{a'_i}{2} + \binom{b'_i}{2} + \binom{c'_i}{2}\right).$$

First, we provide a lower and an upper bound on k, which will be used later. The bounds are obtained by observing that

$$k = \sum_{i=1}^{n}\left(\binom{t}{2} - \binom{a_i}{2} - \binom{b_i}{2} - \binom{c_i}{2} + AB + BC + AC \right.$$
$$\left. + (t - a_i)A + (t - b_i)B + (t - c_i)C\right), \tag{2}$$

as shown in the proof of Lemma 7 in the full version of the paper.

Lemma 7 (★). *For $n \geq 3$ and $d \geq 1$, one has $n^{10d+1} \leq k \leq 2n^{10d+1}$.*

Lemma 7 implies that we cannot merge two large cliques.

Lemma 8 (★). *If I' is a* yes-*instance, then in a solution no two large cliques can be merged.*

At the same time, we cannot reach our budget unless we merge every medium-sized clique with a large one.

Lemma 9 (★). *If I' is a yes-instance, then in any solution graph, for every $i \in [n]$, the clique b'_i is merged with exactly one clique $c'_{\varphi(i)}$, where $\varphi(i) \in [n]$.*

Now we know that a significant part of the budget is spent on merging medium-sized cliques with large cliques. Nevertheless, we cannot meet our budget unless we spend the remaining budget on merging small cliques with a large clique as well.

Lemma 10. *If I' is a* yes-*instance, then in any solution graph, for every $i \in [n]$, the clique a'_i is merged with exactly one clique $c'_{\chi(i)}$, where $\chi(i) \in [n]$.*

Proof. Due to Lemma 9, every solution contains at least

$$\sum_{i=1}^{n} b'_i c'_{\varphi(i)} = \sum_{i=1}^{n} BC + b_i C + c_{\varphi(i)}(b_i + B)$$

edges, where $c'_{\varphi(i)}$ is the large clique that is merged with b'_i. Hence, our remaining budget is $k' := k - \sum_{i=1}^{n} b'_i c'_{\ell_i} =$

$$\sum_{i=1}^{n}\left(\binom{t}{2} - \binom{a_i}{2} - \binom{b_i}{2} - \binom{c_i}{2} + AB + BC + AC\right.$$
$$+ (t - a_i)A + (t - b_i)B + (t - c_i)C$$
$$\left. - \left(BC + b_i C + c_{\varphi(i)}b_i + c_{\varphi(i)}B\right)\right)$$
$$= \sum_{i=1}^{n}\left(\binom{t}{2} - \binom{a_i}{2} - \binom{b_i}{2} - \binom{c_i}{2} + AB + AC\right) \tag{3}$$
$$+ \sum_{i=1}^{n}\left((t - a_i)A + (t - b_i - c_{\varphi(i)})B + (t - c_i - b_i)C\right),$$

due to (2). As $\binom{a_i}{2} + \binom{b_i}{2} + \binom{c_i}{2} < 3n^{2d} < AB$, and the second sum in (3) must be nonnegative (otherwise t is too small, contradicting our assumption in the construction), we obtain that $k' \geq nAC = n^{9d+1}$. Assume towards a contradiction that there is one small clique that is not merged with a large clique. Then, the maximum number of edge additions is achieved by merging $n - 1$ small cliques with all medium-sized cliques and one large clique, and leaving the remaining small clique and $n - 1$ large cliques untouched. The number of edge additions provided by this is at most

$$\sum_{i=1}^{n}\left(b'_i c'_{\varphi(i)} + \binom{n}{2}(B + n^d)^2\right.$$
$$+ n(n-1)(A + n^d)(B + n^d)$$
$$\left. + (n-1)(A + n^d)(C + n^d) + \binom{n}{2}(A^2 + 2An^d + n^{2d})\right)$$
$$\leq \sum_{i=1}^{n} b'_i c'_{\varphi(i)} + n^2(n^{6d} + 2n^{4d} + n^{2d}) + n^2(n^{5d} + n^{4d}$$
$$+ n^{3d} + n^{2d}) + (n-1)(n^{9d} + n^{8d} + n^{3d} + n^{2d})$$
$$+ n^2(n^{4d} + 2n^{3d} + n^{2d}) < \sum_{i=1}^{n} b'_i c'_{\varphi(i)} + n^{9d+1} \leq k,$$

a contradiction to I' being a *yes*-instance. $\square$

6635

Combining Lemmas 9 and 10, we obtain the following.

Lemma 11 (★). *If I' is a* yes-*instance, then every solution graph consists of exactly n cliques.*

With Lemma 11 at hand, we can show that the budget k is exactly met if and only if each resulting clique contains t' vertices.

Lemma 12. *If I' is a* yes-*instance, then every clique in a solution graph G' is of size t'.*

Proof. By Lemma 11, G' consists of n cliques. Let their sizes be $s_1, s_2, \ldots, s_n$. Note that $\sum_{i=1}^n s_i = nt'$, since otherwise the NUMERICAL 3D MATCHING instance I is a *no*-instance. Further, the number $|E(G')|$ of edges in G' is

$$\sum_{i=1}^n \binom{s_i}{2} = \frac{1}{2}\sum_{i=1}^n s_i^2 - \frac{1}{2}\sum_{i=1}^n s_i = \frac{1}{2}\sum_{i=1}^n s_i^2 - \frac{1}{2}nt'.$$

Note that $|E(G')| = |E(G)| + k = n\binom{t'}{2}$. Hence, $\sum_{i=1}^n s_i^2 = nt'^2$ holds. By the Cauchy-Schwarz inequality, we have

$$n \cdot \sum_{i=1}^n s_i^2 = \left(\sum_{i=1}^n 1^2\right)\left(\sum_{i=1}^n s_i^2\right) \geq \left(\sum_{i=1}^n 1 \cdot s_i\right)^2$$
$$= (nt')^2 = n \cdot nt'^2,$$

that is, $\sum_{i=1}^n s_i^2 \geq nt'^2$ holds for all $s_1, \ldots, s_n$ with $\sum_{i=1}^n s_i = nt'$. Note that equality is known to hold only for $s_1 = s_2 = \cdots = s_n = t'$. $\square$

Lemma 13 (★). *If I' is a* yes-*instance, then every clique in a solution graph G' consists of a small, a medium-sized, and a large clique.*

We now have everything at hand to prove Theorem 6.

Proof of Theorem 6. We use our construction to build an instance I' of CLUSTER TRANSFORMATION BY EDGE ADDITION from a given instance I of NUMERICAL 3D MATCHING in polynomial time. Let α, β, γ be a solution for instance I. Creating n clusters by merging the cliques $a'_{\alpha(i)}$, $b'_{\beta(i)}, c'_{\gamma(i)}$ for each $i \in [n]$ yields a solution graph G' with

$$|E(G')| = \sum_{i=1}^n \binom{a'_{\alpha(i)} + b'_{\beta(i)} + c'_{\gamma(i)}}{2}$$
$$= \sum_{i=1}^n \binom{t + A + B + C}{2} = \sum_{i=1}^n \binom{t'}{2}$$

edges, created by adding $|E(G')| - |E(G)| = k$ edges.

Let S be a solution for instance I' and let $G' := G\Delta S$ be the corresponding solution graph. By Lemma 11, G' consists of n clusters of orders $s_1, s_2, \ldots, s_n$. By Lemma 13, there are $\alpha, \beta, \gamma \colon [n] \to [n]$ such that, for every $i \in [n]$, we have $s_i = a'_{\alpha(i)} + b'_{\beta(i)} + c'_{\gamma(i)} = a_{\alpha(i)} + b_{\beta(i)} + c_{\gamma(i)} + A + B + C = t + A + B + C$. Hence, α, β, γ is a solution for our instance I of NUMERICAL 3D MATCHING. $\square$

Set	n	m	k_{opt}	t_{opt}
1	16.1	40.3	14.1	0.0186
2	34.4	128.9	50.0	0.1579
3	69.2	292.0	140.3	13.991
4	114.1	611.2	339.1	2150.6

Table 1: Averages of the number of vertices (n), the number of edges (m), and the solution size (k_{opt}) and running time (t_{opt}) in seconds required to solve standard ("non-fair") CLUSTER EDITING on the graphs in sets 1 through 4.

Experiments

In the spirit of Böcker, Briesemeister, and Klau (2011) who studied classic CLUSTER EDITING, we refrain from using our FPT algorithm (Theorem 5 is rather a classification result) but instead rely on mathematical programming to investigate our model of modification fairness. We evaluate our model on the SNAP Facebook data set (Leskovec and Krevl 2014) which lists for each person (vertex) their gender (color). As this data was gathered from Facebook before 2012, the data on gender is binary. The dataset contains ten graphs, for each of which we sampled four subgraphs of different sizes. We grouped the subgraphs of similar sizes into sets. See Table 1 for the average graph sizes in each set.

We computed optimal solutions for MODIFICATION-FAIR CLUSTER EDITING using an integer linear programming (ILP) formulation of our problem fed into the commercial solver Gurobi 8.1.1. The ILP formulation is based on the standard formulation for CLUSTER EDITING (Grötschel and Wakabayashi 1989), wherein one has a binary variable $x_{i,j}$ for every $\{i, j\} \in \binom{V}{2}$, indicating whether or not the solution graph contains the edge $\{i, j\}$, and three constraints for every vertex triple, which ensure that the triple does not induce a P_3. The formulation can be easily extended to a formulation for MODIFICATION-FAIR CLUSTER EDITING by adding a constraint that ensures that the upper bound δ on our fairness measure Δ_{ed} holds. In order to make the results within the datasets comparable, we introduce a normalized fairness measure $\Delta_{\mathrm{norm}}(S) := \Delta_{\mathrm{ed}}(S)/(2^{|S|}/\min(|R|,|B|))$. Herein, we use the upper bound on Δ_{ed} from Observation 1, Part (iii). Analogously let $\delta_{\mathrm{norm}} := \delta/(2^{k_{\mathrm{opt}}}/\min(|R|,|B|))$, wherein k_{opt} is the minimum size of a "non-fair" cluster editing set. Hence, if $\delta_{\mathrm{norm}} = 1$, our instance is an instance of standard CLUSTER EDITING, see the discussion after Observation 1.

The experiments were run on machines with an Intel Xeon W-2125 4-core 8-thread CPU clocked at 4.0 GHz and 256GB of RAM, running Ubuntu 18.04. All material to reproduce the results is publicly available.[1]

We first evaluate the modification fairness of standard CLUSTER EDITING. Figure 2 shows the number n of vertices and the modification fairness for each of our instances when run with $\delta_{\mathrm{norm}} = 1$. For small graphs, the modification fairness is rather low, with $\Delta_{\mathrm{norm}} \geq 0.1$ for 35% of the graphs in Sets 1 and 2. For larger graphs, however, even
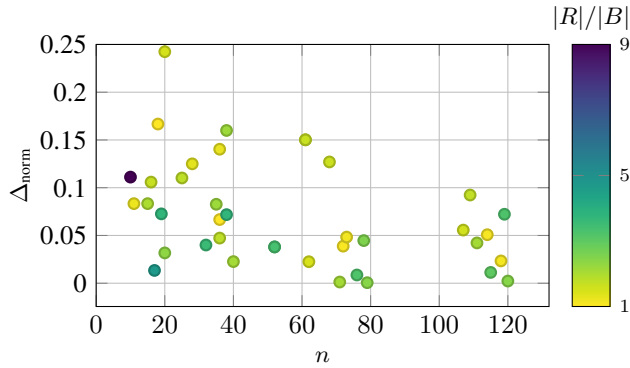
---

[1]https://git.tu-berlin.de/akt-public/mod-fair-ce

Figure 2: *How fair is the "non-fair" variant?* Normed modification fairness Δ_{norm} (y-axis) of the optimal solution for standard CLUSTER EDITING compared to the number n of vertices (x-axis) and the ratio $|R|/|B|$ of red versus blue vertices (color). The ratio is normed such that $|R| \geq |B|$.
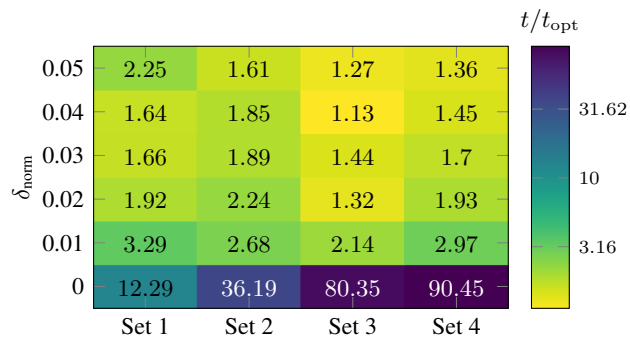

Figure 3: *How much extra time do we need to be fair?* Each heatmap cell contains the average ratio of the running time needed to compute a solution with fairness δ_{norm} and the "non-fair" running time for every set of graphs, see Table 1.

without imposing fairness constraints the solution is already very fair, the mean value of Δ_{norm} being 0.05 for graphs in Set 4. Figure 2 further shows that our tested graphs do not allow for a statement whether the initial modification fairness correlates with the ratio between red and blue vertices.

We next evaluate the *price of fairness*, that is, how much solution size and running time increase when requiring fair solutions. We made runs for $\delta_{\mathrm{norm}} \in \{0, 0.01, 0.02, \ldots, 0.05\}$. We set a time limit of 100 times the time required for running standard CLUSTER EDITING on the same instance and return the best feasible solution found up to this point. The running times and solutions sizes for standard CLUSTER EDITING are gathered in Table 1.

Figure 3 shows that requiring perfect fairness results in prohibitively high running time. Allowing a little bit of slack in the fairness however yields significantly lower increments in running time. For $\delta_{\mathrm{norm}} = 0.01$ and graphs of Set 4, $3/4$ of the runs were slower by a factor of at most 2.81. For $\delta_{\mathrm{norm}} = 0.02$ the factor was at most 1.48 for $3/4$ of the graphs.

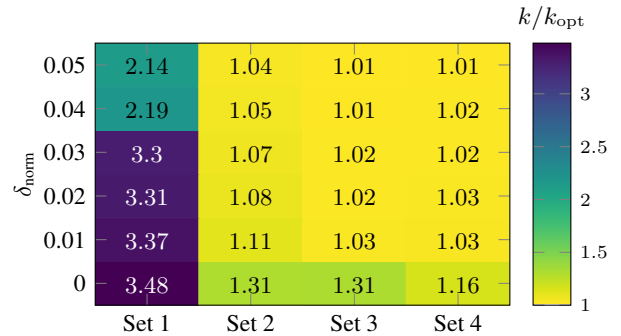Figure 4 shows that, except for very small graphs, the so-

Figure 4: *How much extra edits do we need to be fair?* Each heatmap cell contains the average ratio of the minimum size of a solution with fairness of δ_{norm} and the minimum size of a "non-fair" solution for every set of graphs, see Table 1.

lution size need not increase much to meet modification fairness requirements. For 80% of the graphs in Sets 2, 3 and 4, the solution size increases by at most 33%, 10%, and 8% when δ_{norm} is 0, 0.01, and 0.02, respectively. A likely reason for the higher increment for Set 1 graphs is that a single edge has a higher impact on the modification fairness, i.e., balancing out the modifications requires proportionally more edits.

Conclusion

Undoubtedly, fairness is an elusive, multi-faceted concept bearing many (future) challenges. With our work, we hope to have provided a first step towards process-oriented fairness in graph-based data clustering. Focusing on our newly introduced problem MODIFICATION-FAIR CLUSTER EDITING, there are many research challenges. For instance, in Theorem 5 we showed that MODIFICATION-FAIR CLUSTER EDITING is fixed-parameter tractable for the parameter number k of edge modifications. The corresponding exponential factor is $2^{\mathcal{O}(k \log k)}$—can we improve on this or is there an ETH-based lower bound excluding $2^{o(k \log k)}$?[2]

Arguably, our definition of modification fairness is not the only one that could be studied; e.g., instead of studying the average, one could also study the maximum number of edits per group member. Generally, we believe that some of our results could be adapted to such related concepts. One should also extend our studies to more than two colors. While we focused on exact algorithms, the study of approximation algorithms makes sense as well.

Finally, the fairness investigations could be extended to generalizations of CLUSTER EDITING such as HIERARCHICAL TREE CLUSTERING (Guo et al. 2010a), s-PLEX CLUSTER EDITING (Guo et al. 2010b), but also to other combinatorial problems. This was already done for SHORTEST PATH (Bentert, Kellerhals, and Niedermeier 2022) and MATROID INTERSECTION and BIPARTITE MATCHING (Chierichetti et al. 2019).

---

[2]We remark that for classic CLUSTER EDITING there is a tight bound $2^{\Theta(k)}$ (Komusiewicz and Uhlmann 2012).

Acknowledgements

References

Abbasi, M.; Bhaskara, A.; and Venkatasubramanian, S. 2021. Fair Clustering via Equitable Group Representations. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, 504–514. ACM.

Ahmadian, S.; Epasto, A.; Knittel, M.; Kumar, R.; Mahdian, M.; Moseley, B.; Pham, P.; Vassilvitskii, S.; and Wang, Y. 2020a. Fair Hierarchical Clustering. In *Proceedings of the 33nd Annual Coference on Advances in Neural Information Processing Systems (NeurIPS '20)*.

Ahmadian, S.; Epasto, A.; Kumar, R.; and Mahdian, M. 2020b. Fair Correlation Clustering. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS '20)*, volume 108, 4195–4205. PMLR.

Bandyapadhyay, S.; Fomin, F. V.; Golovach, P. A.; Purohit, N.; and Simonov, K. 2021. FPT Approximation for Fair Minimum-Load Clustering. *CoRR*, abs/2107.09481.

Bandyapadhyay, S.; Fomin, F. V.; and Simonov, K. 2021. On Coresets for Fair Clustering in Metric and Euclidean Spaces and Their Applications. In *Proceedings of the 48th International Colloquium on Automata, Languages, and Programming (ICALP '21)*, 23:1–23:15.

Bentert, M.; Kellerhals, L.; and Niedermeier, R. 2022. Finding Balance-Fair Short Paths in Graphs. *CoRR*, abs/2203.17132.

Böcker, S.; and Baumbach, J. 2013. Cluster Editing. In *9th Conference on Computability in Europe, CiE 2013*, volume 7921 of *LNCS*, 33–44. Springer.

Böcker, S.; Briesemeister, S.; and Klau, G. W. 2011. Exact Algorithms for Cluster Editing: Evaluation and Experiments. *Algorithmica*, 60(2): 316–334.

Cai, L. 1996. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4): 171–176.

Chakrabarty, D.; and Negahbani, M. 2021. Better Algorithms for Individually Fair k-Clustering. In *Proceedings of the 35th Conference on Advances in Neural Information Processing Systems (NeurIPS '21)*.

Chen, J.; Huang, X.; Kanj, I. A.; and Xia, G. 2006. Strong computational lower bounds via parameterized complexity. *Journal of Computer and System Sciences*, 72(8): 1346–1367.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair Clustering Through Fairlets. In *Proceedings of the 31st Conference on Advances in Neural Information Processing Systems (NIPS '17)*, 5029–5037.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2019. Matroids, Matchings, and Fairness. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics (AISTATS '19)*, volume 89, 2212–2220. PMLR.

Friggstad, Z.; and Mousavi, R. 2021. Fair Correlation Clustering with Global and Local Guarantees. In *Proceedings of the 17th International Symposium on Algorithms and Data Structures (WADS '21)*, 414–427. Springer.

Garey, M. R.; and Johnson, D. S. 1975. Complexity results for multiprocessor scheduling under resource constraints. *SIAM Journal on Computing*, 4: 397–411.

Ghadiri, M.; Samadi, S.; and Vempala, S. S. 2021. Socially Fair k-Means Clustering. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT '21)*, 438–448. ACM.

Grötschel, M.; and Wakabayashi, Y. 1989. A cutting plane algorithm for a clustering problem. *Mathematical Programming*, 45(1-3): 59–96.

Guo, J.; Hartung, S.; Komusiewicz, C.; Niedermeier, R.; and Uhlmann, J. 2010a. Exact Algorithms and Experiments for Hierarchical Tree Clustering. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010*, 457–462. AAAI Press.

Guo, J.; Komusiewicz, C.; Niedermeier, R.; and Uhlmann, J. 2010b. A More Relaxed Model for Graph-Based Data Clustering: s-Plex Cluster Editing. *SIAM J. Discret. Math.*, 24(4): 1662–1683.

Kellerhals, L.; Koana, T.; Nichterlein, A.; and Zschoche, P. 2021. The PACE 2021 Parameterized Algorithms and Computational Experiments Challenge: Cluster Editing. In *Proceedings of the 16th International Symposium on Parameterized and Exact Computation (IPEC '21)*, volume 214 of *LIPIcs*, 26:1–26:18.

Komusiewicz, C.; and Uhlmann, J. 2012. Cluster editing with locally bounded modifications. *Discrete Applied Mathematics*, 160(15): 2259–2270.

Leskovec, J.; and Krevl, A. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

Lovász, L.; and Plummer, M. D. 2009. *Matching Theory*. AMS Chelsea Publishing.

Mahabadi, S.; and Vakilian, A. 2020. Individual Fairness for k-Clustering. In *Proceedings of the 37th International Conference on Machine Learning (ICML '20)*, volume 119, 6586–6596.

Mehrabi, N.; Morstatter, F.; Saxena, N.; Lerman, K.; and Galstyan, A. 2021. A Survey on Bias and Fairness in Machine Learning. *ACM Computing Surveys*, 54(6): 115:1–115:35.

Samadi, S.; Tantipongpipat, U. T.; Morgenstern, J.; Singh, M.; and Vempala, S. S. 2018. The Price of Fair PCA: One Extra dimension. In *Proceedings of the 31st Annual Coference on Advances in Neural Information Processing Systems (NeurIPS '18)*, 10999–11010.

Vakilian, A.; and Yalçıner, M. 2021. Improved Approximation Algorithms for Individually Fair Clustering. *CoRR*, abs/2106.14043.