

Matrix Editing Meets Fair Clustering: Parameterized Algorithms and Complexity

Robert Ganian, Hung P. Hoang, Simon Wietheger

Algorithms and Complexity Group, TU Wien, Austria
{rganian, phoang, swietheger}@ac.tuwien.ac.at

Abstract

We study the computational problem of computing a fair means clustering of discrete vectors, which admits an equivalent formulation as editing a colored matrix into one with few distinct color-balanced rows by changing at most k values. While NP-hard in both the fairness-oblivious and the fair settings, the problem is well-known to admit a fixed-parameter algorithm in the former “vanilla” setting. As our first contribution, we exclude an analogous algorithm even for highly restricted fair means clustering instances. We then proceed to obtain a full complexity landscape of the problem, and establish tractability results which capture three means of circumventing our obtained lower bound: placing additional constraints on the problem instances, fixed-parameter approximation, or using an alternative parameterization targeting tree-like matrices.

1 Introduction

In a typical matrix modification problem, we are given a matrix M and are tasked with modifying it into some matrix M' satisfying a specified desirable property. Matrix modification problems arise in a broad range of research contexts directly related to artificial intelligence and machine learning, prominently including recommender systems and data recovery (Candès and Plan 2010; Candès and Recht 2012; Elhamifar and Vidal 2013) but also occurring in, e.g., Markov inference (Roth and Yih 2005) and computational social choice (Bredereck et al. 2014). Matrix completion is perhaps the most classical example of matrix modification: there, certain entries in the provided matrix M are marked as “missing” and the task is to complete the missing entries. A second classical example—one which will be the focus of our interests here—is matrix editing, where we are allowed to alter at most k entries of a (complete) matrix M in order to achieve the sought-after property.

The vast majority of matrix completion and matrix editing problems are known to be NP-hard, leading to the investigation of these problems using the more refined *parameterized complexity* paradigm (Cygan et al. 2015). There, the general aim is to circumvent the intractability of problems by designing algorithms with running times which are not exponential in the whole input size, but only exponential in

some well-defined integer *parameters* of the input. From a complexity-theoretic perspective, we ask for which natural parameters p one can obtain an algorithm solving the problem in time $f(p) \cdot n^{O(1)}$, where f is a computable function and n the input size; such algorithms are called *fixed-parameter* and form a weaker (but still desirable) baseline of tractability than polynomial-time algorithms.

The parameterized complexity of matrix completion was first investigated by Ganian, Kanj, Ordyniak and Szeider (2018), who targeted the two fundamental cases where M' must adhere to an input-specified bound r on either the rank, or the number of distinct rows. Subsequent works in the completion setting then considered a variety of different constraints on the output matrix M' (Eiben et al. 2021, 2023a; Ganian et al. 2022; Koana, Froese, and Niedermeier 2020, 2023). For matrix editing, Fomin, Golovach and Panolan (2020) studied the parameterized complexity of the two problems analogous to those considered in the completion setting (Ganian et al. 2018) and as their main positive result obtained a fixed-parameter algorithm for the task of editing a binary matrix to achieve at most r distinct rows, parameterized by the budget k on the number of altered entries (i.e., edits).¹ This task is particularly interesting, as it precisely corresponds to the classical BINARY MEANS CLUSTERING problem (Kleinberg, Papadimitriou, and Raghavan 2004; Ostrovsky and Rabani 2002)—a discrete counterpart to the means clustering that is frequently used on real-valued data in machine learning (Charikar et al. 2023; Marom and Feldman 2019; Zhang, Lange, and Xu 2020). Intuitively, the reason the editing and clustering tasks coincide is that each time we edit a row \vec{v} to its final value \vec{w} , the number of edits is equal to the cost of placing a data point \vec{v} into a cluster centered at \vec{w} ; see Figure 1 (*Top*) for an illustration.

In this article, we investigate the computational complexity of the same task of editing a matrix to achieve at most r distinct rows, but in the presence of a *fairness constraint*. The reason for considering fairness in this setting is directly tied to the clustering perspective, where requiring each of the clusters to be “fair” is equivalent to ensuring that the (at most r) distinct rows in M' are “fair”. Fair clustering has

¹We remark that there, the bound r is chosen to be on the number of columns as opposed to rows; however, the role of columns and rows is entirely symmetric.

become an increasingly important research topic in recent years (Amagata 2024; Backurs et al. 2019; Bandyapadhyay, Fomin, and Simonov 2024; Dickerson et al. 2023), starting from the pioneering paper of Chierichetti, Kumar, Lattanzi and Vassilvitskii (Chierichetti et al. 2017). The fairness constraint we adopt here is the same as in the latter foundational work: each row is equipped with a specified color (representing an aspect of that data point that should be proportionately represented in clusters) and each cluster must admit a partitioning into *fairlets*, which are minimum sets of colored elements exhibiting the same color ratio as M . (For example, for two colors with 1:1 ratio, a fairlet contains one element of each color—see Fig. 1 (Bottom).) We remark that while this “canonical” fairness constraint has been used in several related works (Ahmadian et al. 2020; Bandyapadhyay, Fomin, and Simonov 2024; Casel et al. 2023), we also discuss possible extensions of our results to other fairness notions in Section 7.

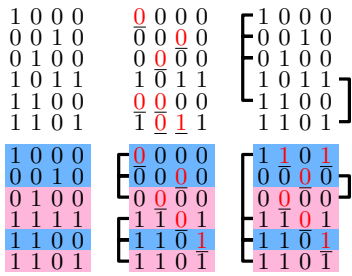


Figure 1: **(Top)** A matrix (left) received 7 edits (center), reducing the number of distinct rows from 6 to 2. Equivalently, the rows are partitioned into clusters (right) with centers 0000 and 1011, respectively. Hamming distances between rows and their respective center equal the number of edits in the middle. **(Bottom)** Example of matrix editing with fairness colors blue and pink. The center matrix has 2 distinct rows but is not fair. The right matrix is fair, with one cluster consisting of a single fairlet (a blue row and a pink row) and the other cluster consisting of two fairlets.

In this work, we consider both the binary and higher-domain settings. We denote our general problem of interest FAIR DISCRETE MEANS CLUSTERING (or FDMC) over some input-specified domain (of distinct entries); this matches the fair version of the matrix editing task where M' must have at most r distinct rows. 2-FDMC is then the restriction to the binary case and is equivalent to BINARY MEANS CLUSTERING under the aforementioned fairness constraint. Formal definitions are provided in Section 2.

Contributions. Given the aforementioned fixed-parameter algorithm of Fomin, Golovach and Panolan (2020) for BINARY MEANS CLUSTERING parameterized by k , a first natural question that arises in our study is whether one can achieve an analogous result when we require the obtained clustering to be fair. As our first contribution, we provide a non-trivial reduction which rules this out under well-established complexity assumptions:

Theorem 1. *2-FDMC is W[1]-hard when parameterized by the fairlet size \tilde{c} plus the budget k , even if M already achieves the target number r of distinct rows.*

One may notice that Theorem 1 excludes fixed-parameter algorithms not only when parameterized by k alone, but even if the parameter includes \tilde{c} , the fairlet size and—as a direct consequence—also the number of colors. Essentially, our reduction shows that the problem becomes intractable w.r.t. k if all the clusters must include a balanced combination of a small number of colors. However, what is perhaps even more remarkable is that the lower bound holds even if M already has at most r distinct rows, i.e., it also applies to instances which are trivial in the “vanilla” setting without fairness.

In the remainder of the paper, we provide tractability results to circumvent this strong lower bound via three different avenues: additional constraints, approximation, and alternative parameterizations.

Additional Constraints. In Theorem 2, we show that FDMC is, in fact, fixed-parameter tractable w.r.t. the number of edits when dealing with instances where the fairlet sizes are “sufficiently large”.

Theorem 2. *When restricted to instances with $\tilde{c} > k$, FDMC is fixed-parameter tractable w.r.t. k .*

Theorem 2 is fairly surprising, as it shows that the effect of the fairlet size on the problem’s complexity is “non-uniform”: instances with fairlet size 1 precisely correspond to the vanilla settings and are hence also fixed-parameter tractable w.r.t. k . Moreover, we also obtain a fixed-parameter algorithm for the problem when parameterized by the budget plus the bound r on the number of clusters.

Theorem 3. *FDMC is fixed-parameter tractable with respect to $k + r$.*

These results allow us to piece together the full complexity landscape of FDMC when parameterized by every combination of k , r , \tilde{c} , the number c of colors and the domain, as we discuss at the end of Section 4. Nevertheless, the proofs of these two theorems are comparatively simple and can hence be seen as a gentle introduction to the reasoning we will employ for the results which form the bulk of our algorithmic contributions—specifically Theorems 4 and 5.

Approximation. Theorems 2 and 3 allow us to circumvent the aforementioned lower bound if certain conditions are met; however, a more generally applicable approach would be to ask for a fixed-parameter algorithm parameterized by k alone that can compute a fair clustering which is at least approximately optimal (in the number k of edits). Fixed-parameter approximation algorithms have found applications for a number of other clustering problems to date (Bandyapadhyay, Fomin, and Simonov 2024; Goyal and Jaiswal 2023; Zhang et al. 2024).

In terms of approximation, the “vanilla” BINARY MEANS CLUSTERING is known to admit a randomized approximation (Ostrovsky and Rabani 2002) and also a deterministic approximation (Fomin et al. 2020). However, both of these algorithms require the number of clusters to be fixed in order to

run in polynomial time, and moreover none of the techniques developed in the previous works can be directly applied to solve our problem of interest here.

Instead, we develop a new approach utilizing a matching-based decomposition of the *edit graph*, which is a hypothetical structure capturing the modifications carried out by an optimal solution. By using this decomposition to establish the existence of a near-optimal and “well-structured” clustering, we obtain the following constant-factor approximation:

Theorem 4. *FDMC admits a $(5 - 3/\tilde{c})$ -approximate fixed-parameter algorithm with respect to k .*

Alternative Parameterizations. For our final contribution, we show that one can in fact have an exact fixed-parameter algorithm at least for 2-FDMC under a different parameterization than the number k of edits. Towards this, we consider a structural measure of the input matrix \mathbf{M} , thus yielding exact algorithms—even for instances requiring a large number of edits—whose performance scales with how “well-structured” \mathbf{M} is. Our structural measure of choice here is the *treewidth* t of \mathbf{M} , which has been successfully employed for other problems on binary matrices (Eiben et al. 2023b; Ganian et al. 2022) but not yet in the editing setting. Essentially, t measures how tree-like the interactions are between rows which share the underrepresented value (say 1) on the same coordinate, and is obtained by measuring the *treewidth* of the so-called *primal graph* of \mathbf{M} (Ganian et al. 2022). By developing a complex dynamic programming subroutine that not only carefully aggregates information from the previously processed parts of the input, but also anticipates the properties of the remainder of the instance, we obtain:

Theorem 5. *2-FDMC is fixed-parameter tractable with respect to the treewidth of \mathbf{M} .*

We remark that Theorem 5 also yields, as a special case, an alternative parameterization that can be used to solve the “vanilla” clustering problem studied, e.g., by Fomin, Golovach and Panolan (Fomin, Golovach, and Panolan 2020).

2 Preliminaries

For a positive integer i , we write $[i]$ for the set $\{1, 2, \dots, i\}$. For an $m \times n$ matrix \mathbf{M} (i.e., a matrix with m rows and n columns over some arbitrary domain), and for $i \in [m], j \in [n]$, $\mathbf{M}[i, j]$ denotes the entry in the i -th row and j -th column of \mathbf{M} . We write $\mathbf{M}[i, \star]$ for the row-vector $(\mathbf{M}[i, 1], \mathbf{M}[i, 2], \dots, \mathbf{M}[i, n])$, and $\mathbf{M}[\star, j]$ for the column-vector $(\mathbf{M}[1, j], \mathbf{M}[2, j], \dots, \mathbf{M}[m, j])$. We call an n -dimensional vector a *type* and refer to $\mathbf{M}[i, \star]$ as the type of row i . We let $\text{dr}(\mathbf{M})$ denote the number of distinct types (i.e., the number of distinct rows) of \mathbf{M} . We denote by $p(\mathbf{M})$ the number of distinct entries in \mathbf{M} . For two types τ_1, τ_2 , let $\text{Hamm}(\tau_1, \tau_2)$ be their Hamming distance. For two $m \times n$ matrices \mathbf{M}, \mathbf{M}' , we write $\|\mathbf{M} - \mathbf{M}'\|_{\neq 0}$ to denote the number of entries in which they differ.

We refer to the maximal sets of pairwise identical rows in a matrix as *clusters*. For an m -dimensional vector γ , and for $i \in [m]$, $\gamma[i]$ denotes the i -th entry of the vector. For

$c, m \in \mathbb{N}$, a vector $\gamma \in [c]^m$ is a (c, m) -coloring. The input for our problem will formally include an $m \times n$ matrix \mathbf{M} and a (c, m) -coloring γ ; we say that the i -th row of \mathbf{M} has color $\gamma[i]$. We call \mathbf{M} *fair* (w.r.t. γ) if all its clusters are fair, that is, they each witness the same color distribution as γ . Formally, for each color $i \in [c]$, each fair cluster S contains precisely $\frac{|\gamma|_i}{m} \cdot |S|$ rows of color i , where $|\gamma|_i$ denotes the number of entries of value i in γ . We call a type τ \mathbf{M} -*fair* if the cluster of this type in \mathbf{M} is empty or fair, and \mathbf{M} -*unfair* otherwise. For a (c, m) -coloring γ , let $\tilde{c} = m / \text{gcd}(|\gamma|_1, \dots, |\gamma|_c)$ denote the minimum size of any fair set of rows, where gcd denotes the greatest common divisor. A *fairlet* is a fair cluster of size \tilde{c} , and we also refer to \tilde{c} as the *fairlet size*. We are now ready to define our problem of interest.²

Fair Discrete Means Cluster Editing (FDMC)

Input: $m \times n$ matrix \mathbf{M} , (c, m) -coloring γ , positive integers k and r

Task: Find a $m \times n$ matrix \mathbf{M}' that is fair for γ such that $\|\mathbf{M} - \mathbf{M}'\|_{\neq 0} \leq k$ and $\text{dr}(\mathbf{M}') \leq r$.

We refer to such a matrix \mathbf{M}' as a *solution* to the instance. We let 2-FDMC denote the problem in the binary domain, that is, $p(\mathbf{M}) \leq 2$. Note that for $c = 1$, every $m \times n$ matrix \mathbf{M}' is fair and hence in this case FDMC reduces to the classical Matrix Editing problem without the fairness constraint.

Given a matrix \mathbf{M}' and an instance $\mathcal{I} = (\mathbf{M}, \gamma, k, r)$ of FDMC, we define the *edit graph* $G_{\mathcal{I}}^{\mathbf{M}'}$ as the following edge-colored edge-labeled edge-weighted directed multigraph. The vertex set of $G_{\mathcal{I}}^{\mathbf{M}'}$ is the set of types occurring in \mathbf{M} or \mathbf{M}' . For each row index $t \in [m]$, there is exactly one edge of $G_{\mathcal{I}}^{\mathbf{M}'}$ from $\mathbf{M}[t, \star]$ to $\mathbf{M}'[t, \star]$; this edge has label $t \in [m]$, color $\gamma[t] \in [c]$, and weight $\text{Hamm}(\mathbf{M}[t, \star], \mathbf{M}'[t, \star])$. We drop the subscript from $G_{\mathcal{I}}^{\mathbf{M}'}$ when the instance is clear from context. An illustration of an edit graph is provided in Fig. 2.

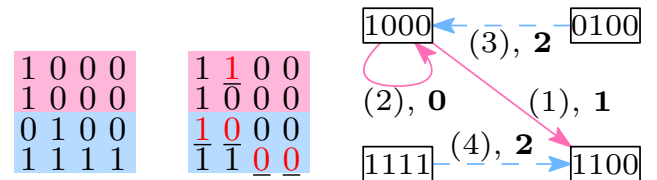


Figure 2: Example of an edit graph (right) describing the changes from a matrix \mathbf{M} (left) to a matrix \mathbf{M}' (center). Weights are printed in **bold** and labels are printed in (brackets). Solid and dashed edges represent changes in the two different fairness colors.

We call $G^{\mathbf{M}'}$ *fair* if its edge set witnesses the same color distribution as γ . It is easy to see that $G^{\mathbf{M}'}$ is fair if and only if \mathbf{M}' is fair. We say a type *survives* in $G^{\mathbf{M}'}$, if it has an incoming edge in $G^{\mathbf{M}'}$. Therefore, \mathbf{M}' is a solution of \mathcal{I} if

²For purely complexity-theoretic reasons, here we formalize the problem in its decision variant. However, all algorithmic results obtained in this article are constructive and can also solve the corresponding optimization task.

and only if $G^{M'}$ is fair, at most r vertices survive in $G^{M'}$, and the total edge weight of $G^{M'}$ is at most k . Analogous to the fairness of clusters, we say that a set of edges is *fair* if it witnesses the same color distribution as γ .

To simplify some proofs, we define the *reduced edit graph* $R(G^{M'})$ as the graph obtained from $G^{M'}$ by removing self-loops and edge labels. (Note that the reduced edit graph can still have multi-edges.) For a reduced edit graph H , there may be many matrices M' such that $R(G^{M'}) = H$; however, all of these are equivalent up to permutation of rows, and in particular either all or none are solutions.

Parameterizations. We refer to the literature (Cygan et al. 2015) for a formalization of parameterized complexity theory beyond the basic overview provided in Section 1, including the notions of *parameterized reductions*, XP and W[1]-*hardness*. Note that the latter rules out fixed-parameter tractability under standard complexity assumptions.

If the domain of M is $\{0, 1\}$, the *primal graph* $G_P(M)$ consists of vertices v_1, \dots, v_m and $v_h v_i \in E(G_P(M))$ if and only if there is a column j such that $M[h, j] \neq 0$ and $M[i, j] \neq 0$. In other words, we can construct $G_P(M)$ by first adding a vertex for each row and then, for each column of M , adding a clique among the vertices corresponding to the rows with a nonzero entry in that column. The *treewidth* $\text{tw}(G)$ of a graph G is a well-established measure of how “tree-like” it is; for instance, trees have a treewidth of 1 while an n -vertex complete graph has treewidth $n - 1$. For a definition of treewidth, including the notions of *nice tree decompositions* and *bags* which are used in the formal proof of Theorem 5, we refer to the literature (Cygan et al. 2015, Subsection 7.2). We let the *treewidth* of M denote the treewidth of its primal graph, i.e., $\text{tw}(M) = \text{tw}(G_P(M))$.

3 Fixed-Parameter Intractability of 2-FDMC

We first note that the NP-hardness of BINARY MEANS CLUSTERING directly transfers to FDMC:

Fact 1 ((Feige 2014)). 2-FDMC is NP-hard even if $c = \tilde{c} = 1$ and $r = 2$.

As our first contribution, we establish that—in contrast to its fairness-oblivious variant—our problem of interest does not admit a fixed-parameter algorithm under standard complexity assumptions.

Theorem 1. 2-FDMC is W[1]-hard when parameterized by the fairlet size \tilde{c} plus the budget k , even if M already achieves the target number r of distinct rows.

Proof Sketch. We reduce from the MULTI-COLORED CLIQUE problem which asks, given a properly vertex-colored graph $G = (V = \{v_1, \dots, v_{|V|}\}, E = \{e_1, \dots, e_{|E|}\})$ with q colors $[q]$, whether G contains a clique of size q . Such a clique must contain precisely one vertex of each color, and the problem is well-known to be W[1]-hard w.r.t. q (Cygan et al. 2015). For each vertex v , let $\text{col}(v)$ refer to its color. Given an instance of the MULTI-COLORED CLIQUE problem with $q \geq 4$, we construct a 2-FDMC instance $(M, \gamma, \text{dr}(M), k)$ as follows. We begin by carefully choosing a budget $k \in \mathcal{O}(q^5)$, and produce a matrix M with

$m \in \mathcal{O}(q^8 + q^6(|V| + |E|))$ rows and $\mathcal{O}(q^4 + |V|)$ columns along with a $(2q, m)$ -coloring γ by following the description provided below.

We identify the colors of the MULTI-COLORED CLIQUE instance with the first q row colors for 2-FDMC. Next we list the types used in M . For every color $i \in [q]$, we have a set of $(q - 1)$ types \mathcal{V}_{it} for $t \in [q] \setminus \{i\}$. For every pairs of colors $i, j \in [q], i < j$, we have a set of $(2q - 2)$ types \mathcal{E}_{ijt} for $t \in [2q] \setminus \{i, j\}$. For each vertex v_a for $a \in [|V|]$ with color i , we have one type V_a . Lastly, for each edge $v_a v_b$ for some $a, b \in [|V|], a < b$ with $\text{col}(v_a) = i, \text{col}(v_b) = j$, we have one type E_{ab} . We carefully design the types such that the pairwise Hamming distances are as listed in Table 1.

	$\mathcal{V}_{it'}$	$\mathcal{E}_{ijt'}$	$V_{a'}$	$E_{a'b'}$
\mathcal{V}_{it}	4	$y + 4$	3	$y + 4$
\mathcal{E}_{ijt}	—	4	$y + 3$	4
V_a	—	—	2	$y + 1$
E_{ab}	—	—	—	2

Table 1: Minimum Hamming distances between distinct types, for a carefully selected $y \in \Theta(q^3)$. These are only achieved if $\{\text{col}(a), \text{col}(b)\} = \{i, j\}, \text{col}(a) = \text{col}(a')$, and $\text{col}(b) = \text{col}(b')$; distances between other pairs of types are too large to be relevant for a hypothetical solution.

We then create rows of each type and color such that each of the types V_a and E_{ab} is M -fair; at the same time, each type \mathcal{V}_{it} has an “excess” of one row of color t and each type \mathcal{E}_{ijt} has a “deficit” of one row of color t . By the choices of distances between types and the ranges for t , we can show that the only way to make every cluster fair within the prescribed budget k is to do the following. Each \mathcal{V}_{it} sends its excess row to some type $V_{h(i)}$ such that $\text{col}(h(i)) = i$ (representing a vertex in the clique). Further, for each pair of colors $i < j$ in $[q]$, $E_{h(i)h(j)}$ (representing an edge in the clique) sends a row of color j to $V_{h(i)}$, a row of color i to $V_{h(j)}$, and each remaining row to some \mathcal{E}_{ijt} . Note that for this to yield a solution, $E_{h(i)h(j)}$ has to exist for every pair of colors i, j in $[q]$; that is, $v_{h(1)}, \dots, v_{h(q)}$ must form a (multi-colored) clique. \square

4 Fixed-Parameter Algorithms for FDMC via Additional Constraints

Our aim in this section is to identify constraints under which we can circumvent the lower bounds in Section 3. By the end of the section, we will in fact have obtained a comprehensive classification of the problem’s complexity under the considered parameterizations.

We begin by noting that FDMC admits a straightforward $\mathcal{O}((mnp(M))^k)$ -time algorithm—indeed, one can exhaustively branch over precisely which cells are edited into which value of M .

Observation 1. FDMC is in XP when parameterized by k .

Next, we identify the first constraint that yields fixed-parameter tractability: in particular, a fairlet size that is larger than the parameter k .

Theorem 2. *When restricted to instances with $\tilde{c} > k$, FDMC is fixed-parameter tractable w.r.t. k .*

Proof. Every fair cluster contains at least \tilde{c} rows. Hence, due to the fairness constraint, when $\tilde{c} > k$ we cannot create any new type in \mathbf{M}' . Further, for an \mathbf{M} -fair type, in order for it to be \mathbf{M}' -fair, we have to remove or add either zero or at least \tilde{c} rows, requiring at least \tilde{c} edits. Therefore, as $\tilde{c} > k$, no row can be edited from or into an \mathbf{M} -fair type.

With k edits, the clusters of at most $2k$ types change between an input matrix \mathbf{M} and a solution matrix \mathbf{M}' . Thus, if more than $2k$ types are \mathbf{M} -unfair, we can correctly reject. For every \mathbf{M} -unfair type τ with a cluster S , note that there are at most two sizes in $\{\tilde{c}i \mid i \in \mathbb{N}, |S| - k \leq \tilde{c}i \leq |S| + k\}$ that the cluster of type τ can have in a solution \mathbf{M}' . We test each of the at most 2^{2k} branches for all choices across all \mathbf{M} -unfair types. In particular, we attempt to construct a fair reduced edit graph such that there is no new type, all \mathbf{M} -fair types are isolated (i.e., they are incident to only self-loops in the ordinary edit graph), and the total edge weight is at most k . We do so by having colored “half-edges” at each \mathbf{M} -unfair type, such that the number of incoming (or outgoing) half-edges of each color is the number of rows of that color to be added to (or removed from) its cluster to obtain a fair cluster of the branched size. Then we match each outgoing half-edge with an incoming half-edge of the same color and assign the combined edge the Hamming distance between the corresponding types as weight. By fixing an arbitrary order of the outgoing half-edges, we can view a matching of these half-edges as a permutation of the incoming half-edges—yielding at most $\mathcal{O}(k!)$ such matchings and allowing us to exhaustively branch over these. After that, for each graph, we compute its total edge weight, which can be done in $\mathcal{O}(kn)$ time. Clearly, the FDMC instance has a solution if and only if at least one of the branches succeeds. In total, this algorithm runs in time $\mathcal{O}(2^{2k} k! \cdot kn + m)$. \square

Next, we show that the problem is fixed-parameter tractable with respect to $k + r$. For this, we first establish that we can assume that every YES-instance has at least one well-structured solution, which follows by observing that the center for any cluster can be determined by column-wise majority votes.

Lemma 1. *Every YES-instance $(\mathbf{M}, \gamma, k, r)$ of FDMC is witnessed by a matrix \mathbf{M}^* such that for every cluster $S = \{s_1, \dots, s_{|S|}\}$ in \mathbf{M}^* and for each $j \in [n]$, $\mathbf{M}^*[s_1, j]$ corresponds to the winner of a majority vote of $\{\mathbf{M}[s_i, j] \mid i \in [|S|]\}$, breaking ties arbitrarily.*

Theorem 3. *FDMC is fixed-parameter tractable with respect to $k + r$.*

Proof. Since we can remove at most k clusters with k edits, if there are more than $k + r$ types in \mathbf{M} , we correctly reject. If $\tilde{c} > k$, we use the algorithm in Theorem 2. Otherwise, we have $c \leq \tilde{c} \leq k$.

Consider a modified reduced edit graph where there are $k/2$ new types that are undetermined (we will determine them later). The number of new types is at most $k/2$, because we can assume that a cluster with a new type is formed by rows

of at least two types in \mathbf{M} (otherwise it would be cheaper to keep the previous type). Since $c < k$, there are at most $\mathcal{O}((2^{\binom{3k/2+r}{2}} k)^k)$ such graphs. We disregard graphs with outgoing edges of new types. For each graph, we then fix the new types according to the majority vote rule described in Lemma 1. Finally, we check whether the resulting graph is indeed a fair reduced edit graph and whether the total edge weight is at most k .

It is straightforward to see that for a YES-instance, there exists a graph that passes all the checks above, while for a NO-instance, each graph will fail at least one check. We then output accordingly. The running time of the algorithm is at most $(k + r)^{\mathcal{O}(k)} \cdot (n + m)^{\mathcal{O}(1)}$, and the theorem follows. \square

With Theorem 3, we have an essentially full picture of the parameterized complexity of FDMC w.r.t. $k, r, p(\mathbf{M}), c$ and \tilde{c} . Indeed, the problem is fixed-parameter tractable w.r.t. every superset of $\{k, r\}$. All remaining considered parameterizations which are supersets of $\{k\}$ yield XP-tractability and W[1]-hardness. Finally, FDMC is paraNP-hard under every considered parameterization not covered by the first two cases.

5 Parameterized Approximation

In this section, we discuss the parameterized approximability of FDMC. To this end, we say that FDMC admits an α -approximate fixed-parameter algorithm if there is a fixed-parameter algorithm that for every instance $\mathcal{I} = (\mathbf{M}, \gamma, r, k)$ of FDMC either correctly identifies that \mathcal{I} is a NO-instance, or outputs a fair $m \times n$ matrix \mathbf{M}' such that $\text{dr}(\mathbf{M}') \leq r$ and $\|\mathbf{M} - \mathbf{M}'\|_{\neq 0} \leq \alpha k$. We start with the following key technical result that our main theorem for this section hinges upon.

Lemma 2. *For every YES-instance $(\mathbf{M}, \gamma, r, k)$ of FDMC, there is a fair matrix \mathbf{M}' such that (i) at most r vertices survive in $G^{\mathbf{M}'}$; (ii) all these surviving vertices are types of \mathbf{M} ; (iii) each \mathbf{M} -fair type either has no out-neighbor or has no in-neighbor and at most one out-neighbor in $R(G^{\mathbf{M}'})$; and (iv) the total edge weight of $G^{\mathbf{M}'}$ is at most $(5 - 3/\tilde{c})k$.*

Proof Sketch. Let \mathbf{M}^* be a solution of the YES-instance $(\mathbf{M}, \gamma, r, k)$, and recall that $G := G^{\mathbf{M}^*}$ may contain self-loops. By the redirection of some edges and a simple application of the triangle inequality, we can assume that for every \mathbf{M} -fair type τ , in $R(G)$, τ has no out-edges or no in-edges. For the construction of the partitions and auxiliary graphs described below, see Fig. 3 for an illustration.

As G is fair, for each type τ we can partition the set of in-edges to τ (including self-loops) in G into fair sets of size \tilde{c} . Let P_τ^+ denote such a partition. For each \mathbf{M} -fair type τ , additionally consider an arbitrary partition P_τ^- of the out-edges of τ in G into fair sets of size \tilde{c} ; here, P_τ^- is guaranteed to exist since τ is \mathbf{M} -fair and has no out-edges or no in-edges in $R(G)$. As the set of all out-edges from all \mathbf{M} -fair types is fair, the set of all out-edges of all \mathbf{M} -unfair types is fair as well. Thus there exists a partition P_u^- of the latter set into fair sets of size \tilde{c} .

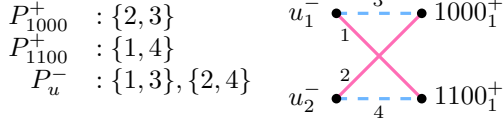


Figure 3: For the edit graph in Fig. 2, the partitions of edges into fair sets of size $\tilde{c} = 2$ are described on the left. (Numbers refer to labels of the edges in Fig. 2.) Note that there are no M-fair types, and hence no partitions of the form P_τ^- in this example. The right side depicts the union of two auxiliary graphs for the two color classes, one with the pink solid edges and one with the blue dashed edges.

For every color $i \in [c]$, create an auxiliary undirected bipartite graph H_i from G as follows. For each type τ , create $|P_\tau^+|$ vertices $\tau_1^+, \tau_2^+, \dots$, called (+)-nodes. For each M-fair type τ , create $|P_\tau^-|$ vertices $\tau_1^-, \tau_2^-, \dots$, called fair (-)-nodes. Further, we create $|P_u^-|$ unfair (-)-nodes u_1^-, u_2^-, \dots . Consider an edge (τ, ρ) of color i and some label $t \in [m]$ in G , i.e., (τ, ρ) intuitively represents the change in the t^{th} row of \mathbf{M} . Suppose this edge belongs to the j^{th} part of P_ρ^+ and the λ^{th} part of either P_τ^- or P_u^- , depending on whether τ is M-fair or not. Then we add to H_i an edge of label t and color i between ρ_j^+ and either τ_λ^- or u_λ^- , whichever is applicable. Proceed this way for all edges of color i .

Let c_i be the number of occurrences of the color i in a fairlet with respect to γ . Observe that H_i as constructed above is bipartite with all the (+)-nodes as one part and all the (-)-nodes as the other. Further, every vertex in H_i has degree c_i . Such a graph can be decomposed into c_i perfect matchings (König 1916). Doing this for all $i \in [c]$, we obtain $\sum_{i \in [c]} c_i = \tilde{c}$ matchings in the multigraph H , defined as the edge-disjoint union of all H_i . We label these matchings $M_1, \dots, M_{\tilde{c}}$ in the increasing order of their total edge weights, breaking ties arbitrarily. Note that for each row index $t \in [m]$ exactly one edge e of H has label t , this edge has color $\gamma[t]$ and belongs to exactly one perfect matching. We assign the weight of the edge of label t in G to e .

We now modify G in three phases. In Phase 1, our aim is to make every fair (-)-node have exactly one neighbor in H , namely its neighbor in M_1 . We do this by repeatedly identifying an alternating path in the union of M_1 and another matching M_j . Then we modify the edges in the matching M_j in the path; most of them become identical to the edges in M_1 , except for potentially one edge incident to an unfair (-)-node, whose weight may increase. We translate these changes in H into G to obtain a graph G_1 . This means that in G_1 , the set of edges from an M-fair type to another type is fair.

In Phase 2, we modify G_1 to obtain G_2 : For every M-fair type τ , we redirect all its out-edges to the type τ_{\min} closest to τ , among all surviving types in G (τ_{\min} may be τ itself). Then each M-fair type has only one out-neighbor in G_2 . In Phase 3, we aim to remove the new types (i.e., types that are not in \mathbf{M}) from G_2 . Let τ be a surviving new type in

G_2 . Note that τ only has in-edges. Let $\tau^{+\text{opt}}$ be the closest in-neighbor of τ in G . We then redirect all in-edges of τ to $\tau^{+\text{opt}}$. Exhaustively performing the procedure above until there is no surviving new type, we obtain an edit graph G_3 .

We then show that G_3 satisfies conditions (i)–(iv) of the lemma. For (iv), note that due to Phase 2, the total weight on out-edges from M-fair types may only increase in the last phase (by a factor of 2). For the out-edges from M-unfair types, we bound the total weight increase via a careful charging scheme and the fact that M_1 has the smallest total edge weight among $M_1, \dots, M_{\tilde{c}}$. \square

Theorem 4. *FDMC admits a $(5 - 3/\tilde{c})$ -approximate fixed-parameter algorithm with respect to k .*

Proof. Consider an instance $(\mathbf{M}, \gamma, r, k)$ of FDMC. If $\tilde{c} \geq k$, compute an exact solution using Theorem 2. Otherwise, if it is a YES-instance then there exists a matrix \mathbf{M}' as described in Lemma 2. In particular, since the total edge weight of $G^{\mathbf{M}'}$ is at most $(5 - 3/\tilde{c})k$, there are at most $2(5 - 3/\tilde{c})k$ non-isolated vertices in $R(G^{\mathbf{M}'})$. Note that all M-unfair types have to be non-isolated in $R(G^{\mathbf{M}'})$.

We now define a colorful variant of the problem as follows: Given an FDMC instance $(\mathbf{M}, \gamma, r, k)$ with an assignment of every M-fair type to a code³ in $[2(5 - 3/\tilde{c})k]$, the goal is to find a fair reduced edit graph G with total edge weight at most $(5 - 3/\tilde{c})k$ and at most r surviving types such that all M-fair types of non-isolated vertices have distinct codes, or to decide that no such graph exists.

Given a fixed-parameter algorithm with respect to k for this colorful variant, we get a $(5 - 3/\tilde{c})$ -approximate fixed-parameter algorithm with respect to k for FDMC, using the *color coding* technique. In particular, it is well-established that for at most m types and $2(5 - 3/\tilde{c})k \leq 10k$ codes one can enumerate a set of encodings of the types (an assignment of each type to a code) in time at most $k^{\mathcal{O}(k)}m$ such that for every set T of types with $|T| \leq 10k$ there is at least one encoding in which all codes in T are distinct (Cygan et al. 2015, Subsections 5.2 and 5.6). Thus, given an instance of FDMC, we can iterate through all enumerated encodings of the M-fair types and, for each of them, solve the respective instance of the colorful variant. As argued above, for every YES-instance of FDMC at least one of the tested encodings will yield a YES-instance of the colorful variant and output a corresponding reduced edit graph. This reduced edit graph then implies the existence of a computable fair $m \times n$ matrix \mathbf{M}' such that $\text{dr}(\mathbf{M}') \leq r$ and $\|\mathbf{M} - \mathbf{M}'\|_{\neq 0} \leq (5 - 3/\tilde{c})k$.

It remains to provide a fixed-parameter algorithm with respect to k to solve a given colorful instance $(\mathbf{M}, \gamma, r, k)$ with a code assignment. We define a *template* as a directed multigraph $H = (V, E)$, where V contains all M-unfair types and some other unlabeled vertices. Each edge of H has a weight of a positive integer and a color in $[\tilde{c}]$. The weight of an edge between two M-unfair types must be the Hamming distance between the two types, and the total edge weight must be at most $(5 - 3/\tilde{c})k$. Each vertex is incident to at

³In the *color coding* technique used here, these codes are usually referred to as *colors* (Alon, Yuster, and Zwick 1995). We employ the term *code* to avoid confusion with the colors in FDMC instances.

least one edge and is assigned a unique code in $[2(5 - 3/\tilde{c})k]$. Each unlabeled vertex has either at most one out-neighbor and no in-neighbor or no out-neighbors.

By the definition above, a template has at most $2(5 - 3/\tilde{c})k$ vertices and $(5 - 3/\tilde{c})k$ edges. Since we assume $\tilde{c} < k$, and since the number of codes is at most $10k$, the number of templates is upper bounded by $k^{\mathcal{O}(k^4)}$. For each template, we test for the existence of an assignment of a type that is \mathbf{M} -fair and not new to each unlabeled vertex, so that the resulting graph can then be *extended* to a reduced unlabeled edit graph on \mathbf{M} and γ (by adding suitable isolated vertices to the resulting graph). In particular, for each unlabeled vertex v of code z we need to identify an \mathbf{M} -fair type τ of code z such that (i) for every out-edge of v there is a distinct row of the same color in the cluster of type τ in \mathbf{M} and (ii) for every edge (v, τ') or (τ', v) , the weight on the edge equals $\text{Hamm}(\tau, \tau')$ (where τ' might be in an \mathbf{M} -unfair type or an \mathbf{M} -fair type assigned to some unlabeled vertex).

For every unlabeled vertex, we discard all \mathbf{M} -fair types of the respective code that do not fulfil property (i) or do not satisfy property (ii) for at least one \mathbf{M} -unfair type. By the definition of the template, the induced subgraph on the unlabeled vertices is a union of pairwise vertex-disjoint stars, each of which has the edges oriented towards the center. We test whether there is a suitable assignment of the remaining \mathbf{M} -fair types to unlabeled vertices independently for every star. For every remaining \mathbf{M} -fair type τ with the code of the center vertex, test whether for each leaf of the star there is at least one remaining \mathbf{M} -fair type with respective code and whose Hamming distance to τ matches the weight of the respective edge. If this is successful for all unlabeled vertices, the resulting graph witnesses a YES-instance of the colorful problem variant. Otherwise, we correctly decide that $(\mathbf{M}, \gamma, r, k)$ is a NO-instance of FDMC. This process can be completed in polynomial time for each template graph, yielding an overall running time bound of $k^{\mathcal{O}(k^4)} \cdot (n + m)^{\mathcal{O}(1)}$.

It remains to show that for every YES-instance for the colorful variant of FDMC, at least one branch succeeds. In this case, there is a matrix \mathbf{M}' as described in Lemma 2. Removing all isolated vertices in $G^{\mathbf{M}'}$ and the labels of all \mathbf{M} -fair types yields a valid template and the proof follows. \square

6 A Treewidth-Based Fixed-Parameter Algorithm

As our final contribution, we provide an alternative route towards fixed-parameter tractability for 2-FDMC—specifically, by utilizing the structure of the input matrix \mathbf{M} rather than the budget k . Here, the restriction to binary matrices is necessary in order to facilitate a suitable definition of $\text{tw}(\mathbf{M})$; that being said, the binary setting has also been extensively studied in the literature (Fomin, Golovach, and Panolan 2020; Kleinberg, Papadimitriou, and Raghavan 2004; Ostrovsky and Rabani 2002).

We first show that instances with a “large” fairlet size are trivial when parameterized by treewidth.

Lemma 3. *A 2-FDMC instance $(\mathbf{M}, \gamma, r, k)$ with $2\text{tw}(\mathbf{M}) + 2 \leq \tilde{c}$ is a YES-instance if and only if k is at least the number of non-zero entries in \mathbf{M} .*

Proof. If k is at least this large, changing every non-zero entry to zero will witness a YES-instance by creating a single fair cluster containing all rows. For the other direction, let $w = \text{tw}(\mathbf{M}) + 1$ and assume a YES-instance is witnessed by a matrix \mathbf{M}' that satisfies Lemma 1. Note that, due to the fairness requirement, every cluster in \mathbf{M}' contains at least $\tilde{c} \geq 2w$ rows. For every column j there are at most w rows i with $\mathbf{M}[i, j] \neq 0$ as all rows with a 1 in column j would form a clique and thus cannot exceed the width w of the tree decomposition. Hence, in every column at least half of the rows in S have value 0, so by Lemma 1 and breaking ties in favor of 0, every cluster has type $(0, \dots, 0)$. Thus, we have $k \geq \|\mathbf{M} - \mathbf{M}'\|_{\neq 0}$, which equals the number of non-zero entries in \mathbf{M} . \square

Theorem 5. *2-FDMC is fixed-parameter tractable with respect to the treewidth of \mathbf{M} .*

Proof Sketch. Let $w = \text{tw}(\mathbf{M}) + 1$ and consider a nice tree decomposition \mathbf{T} of $G_P(\mathbf{M})$ with treewidth at most $\text{tw}(\mathbf{M})$. If $\tilde{c} \geq 2w$ we solve the instance in linear time by using Lemma 3.

Otherwise, we have $\tilde{c} < 2w$ and employ a treewidth dynamic program which, intuitively, works on the following principles. At each node of \mathbf{T} , we keep track of which *states* (configurations of relevant values) are *candidates* (i.e., whether they describe a valid sub-solution). The theorem then follows as we ensure that the number of states at each node is upper bounded by $f(w)(nm)^{\mathcal{O}(1)}$ for a computable function f , the candidates at each node can be efficiently identified given the candidates of its child nodes, and any candidate at the root describes a valid solution matrix.

Here, a state describes how a potential sub-solution partitions the rows (i.e., vertices in the primal graph) in the bag into clusters, the number of distinct types in the sub-solution, and the total cost of all columns which are *processed* for this bag. Crucially, we show that for each column, there is precisely one node where a vertex representing a row with value 1 in that column is forgotten and all remaining rows with a 1 in that column are in the bag. At this step, using Lemma 1, we can already *process* the column, that is, compute the total number of required edits in that column based on each possible partition of the vertices in the bag, as this partition reveals in which clusters there will be a majority of 1s in that column (as all rows outside the bag have a 0 there).

To ensure fairness, states further include some additional information. Intuitively, whenever a new row r of color i is added to a bag, there are three possibilities. Firstly, r may be part of a cluster that is of type $(0, \dots, 0)$ or that includes rows from previous bags and does not overlap the current one. With the above observation, we know that r does not share any 1s in any column with the previous rows in the cluster, so we set all such new rows aside. Secondly, r may be part of a cluster that overlaps the current bag. In this case, we need to ensure that the cluster still has space for color i . In order to check this, for each cluster in a hypothetical solution overlapping the current bag, the state tracks its final size and how many rows of which color in that cluster were already encountered in the bag of its descendants. This way, we know whether “adding” a new row to an existing cluster

would make it irreparably unfair (since from the final size we know how many rows of each color it should have). We show that one can assume the final size of such clusters to be at most $2w$ since for larger clusters all 1s would be set to 0s by Lemma 3, so the corresponding rows would be set aside. Lastly, r may be part of a new cluster that has no rows in any of the previous or the current bags and is of type other than $(0, \dots, 0)$. To ensure that we do not open up too many clusters which cannot be filled later, the state also keeps track of the total (final) size of all clusters which have received at least one row from the bag or its descendants (at most m).

In the end, for a candidate, as the set of all rows is fair and the total final size of clusters that we added vertices to is at most m , there are sufficiently many rows of each color set aside to fill up all the remaining spots in the clusters. The other set-aside rows form a fair set as well and are placed in a cluster with type $(0, \dots, 0)$, yielding a fair solution.

As the number of distinct states at each bag can be shown to lie in $w^{\mathcal{O}(w^2)} krm$, the proof follows. \square

7 Concluding Remarks

We remark that while our investigation concentrates on the well-established fairlet-based notion of fairness, many of our results can be lifted to different or more general variants. For example, if we assume that the instance is equipped with prescribed ranges for the proportion of each color in the clusters, then the lower bounds in Section 3 carry over immediately to the arising more general problem. Moreover, analogous results to those obtained in Section 4 can be obtained using similar proof ideas; on the other hand, it is not immediately clear whether or how the more involved algorithmic ideas in Sections 5 and 6 would generalize to such a setting.

Future work can also focus on optimizing the running time bounds of the algorithms and the lower bounds arising from the reductions in order to pinpoint the exact fine-grained complexity of the problem under the Exponential Time Hypothesis and/or its strong variant (Impagliazzo, Paturi, and Zane 2001).

Another avenue that could be explored in the future is whether the results of Section 6 can be generalized to higher-domain instances. Finally, while this work provides a foundational analysis of the problem’s complexity, it would be interesting to see how and whether the obtained insights can be used in more applied settings.

Acknowledgements

The authors were supported by the Austrian Science Foundation (FWF, project 10.55776/Y1329). The first author was additionally supported by FWF project 10.55776/COE12, and the second author by FWF project 10.55776/ESP1136425. We would also like to thank Frank Sommer for suggesting the decomposition of regular bipartite graphs into edge-disjoint perfect matchings that inspired the auxiliary graph in Lemma 2.

References

Ahmadian, S.; Epasto, A.; Knittel, M.; Kumar, R.; Mahdian, M.; Moseley, B.; Pham, P.; Vassilvitskii, S.; and Wang,

Y. 2020. Fair Hierarchical Clustering. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Alon, N.; Yuster, R.; and Zwick, U. 1995. Color-Coding. *J. ACM*, 42(4): 844–856.

Amagata, D. 2024. Fair k-center Clustering with Outliers. In Dasgupta, S.; Mandt, S.; and Li, Y., eds., *International Conference on Artificial Intelligence and Statistics, 2-4 May 2024, Palau de Congressos, Valencia, Spain*, volume 238 of *Proceedings of Machine Learning Research*, 10–18. PMLR.

Backurs, A.; Indyk, P.; Onak, K.; Schieber, B.; Vakilian, A.; and Wagner, T. 2019. Scalable Fair Clustering. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, 405–413. PMLR.

Bandyapadhyay, S.; Fomin, F. V.; and Simonov, K. 2024. On coresets for fair clustering in metric and Euclidean spaces and their applications. *J. Comput. Syst. Sci.*, 142: 103506.

Bredereck, R.; Chen, J.; Hartung, S.; Kratsch, S.; Niedermeier, R.; Suchý, O.; and Woeginger, G. J. 2014. A Multivariate Complexity Analysis of Lobbying in Multiple Referenda. *J. Artif. Intell. Res.*, 50: 409–446.

Candès, E. J.; and Plan, Y. 2010. Matrix Completion With Noise. *Proc. IEEE*, 98(6): 925–936.

Candès, E. J.; and Recht, B. 2012. Exact matrix completion via convex optimization. *Commun. ACM*, 55(6): 111–119.

Casel, K.; Friedrich, T.; Schirneck, M.; and Wietheger, S. 2023. Fair Correlation Clustering in Forests. In Talwar, K., ed., *4th Symposium on Foundations of Responsible Computing, FORC 2023, June 7-9, 2023, Stanford University, California, USA*, volume 256 of *LIPICs*, 9:1–9:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

Charikar, M.; Henzinger, M.; Hu, L.; Vötsch, M.; and Waingarten, E. 2023. Simple, Scalable and Effective Clustering via One-Dimensional Projections. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10-16, 2023*.

Chierichetti, F.; Kumar, R.; Lattanzi, S.; and Vassilvitskii, S. 2017. Fair Clustering Through Fairlets. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5029–5037.

Cygan, M.; Fomin, F. V.; Kowalik, L. u.; Lokshтанov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized algorithms*. Springer, Cham. ISBN 978-3-319-21274-6; 978-3-319-21275-3.

Dickerson, J. P.; Esmaili, S. A.; Morgenstern, J. H.; and Zhang, C. J. 2023. Doubly Constrained Fair Clustering. In

- Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Eiben, E.; Ganian, R.; Kanj, I.; Ordyniak, S.; and Szeider, S. 2021. The Parameterized Complexity of Clustering Incomplete Data. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, 7296–7304. AAAI Press.
- Eiben, E.; Ganian, R.; Kanj, I.; Ordyniak, S.; and Szeider, S. 2023a. On the parameterized complexity of clustering problems for incomplete data. *J. Comput. Syst. Sci.*, 134: 1–19.
- Eiben, E.; Ganian, R.; Kanj, I. A.; Ordyniak, S.; and Szeider, S. 2023b. The Computational Complexity of Concise Hypersphere Classification. In Krause, A.; Brunskill, E.; Cho, K.; Engelhardt, B.; Sabato, S.; and Scarlett, J., eds., *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, 9060–9070. PMLR.
- Elhamifar, E.; and Vidal, R. 2013. Sparse Subspace Clustering: Algorithm, Theory, and Applications. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(11): 2765–2781.
- Feige, U. 2014. NP-hardness of hypercube 2-segmentation. *CoRR*, abs/1411.0821.
- Fomin, F. V.; Golovach, P. A.; Lokshantov, D.; Panolan, F.; and Saurabh, S. 2020. Approximation Schemes for Low-rank Binary Matrix Approximation Problems. *ACM Trans. Algorithms*, 16(1): 12:1–12:39.
- Fomin, F. V.; Golovach, P. A.; and Panolan, F. 2020. Parameterized low-rank binary matrix approximation. *Data Min. Knowl. Discov.*, 34(2): 478–532.
- Ganian, R.; Hamm, T.; Korchemna, V.; Okrasa, K.; and Simonov, K. 2022. The Complexity of k-Means Clustering when Little is Known. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 6960–6987. PMLR.
- Ganian, R.; Kanj, I. A.; Ordyniak, S.; and Szeider, S. 2018. Parameterized Algorithms for the Matrix Completion Problem. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 1642–1651. PMLR.
- Goyal, D.; and Jaiswal, R. 2023. Tight FPT Approximation for Socially Fair Clustering. *Inf. Process. Lett.*, 182: 106383.
- Impagliazzo, R.; Paturi, R.; and Zane, F. 2001. Which Problems Have Strongly Exponential Complexity? *J. Comput. Syst. Sci.*, 63(4): 512–530.
- Kleinberg, J.; Papadimitriou, C.; and Raghavan, P. 2004. Segmentation problems. *Journal of the ACM (JACM)*, 51(2): 263–280.
- Koana, T.; Froese, V.; and Niedermeier, R. 2020. Parameterized Algorithms for Matrix Completion with Radius Constraints. In Gørtz, I. L.; and Weimann, O., eds., *31st Annual Symposium on Combinatorial Pattern Matching, CPM 2020, June 17-19, 2020, Copenhagen, Denmark*, volume 161 of *LIPICs*, 20:1–20:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.
- Koana, T.; Froese, V.; and Niedermeier, R. 2023. The complexity of binary matrix completion under diameter constraints. *J. Comput. Syst. Sci.*, 132: 45–67.
- König, D. 1916. Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre. *Math. Ann.*, 77(4): 453–465.
- Marom, Y.; and Feldman, D. 2019. k-Means Clustering of Lines for Big Data. In Wallach, H. M.; Larochelle, H.; Beygelzimer, A.; d’Alché-Buc, F.; Fox, E. B.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 12797–12806.
- Ostrovsky, R.; and Rabani, Y. 2002. Polynomial-time approximation schemes for geometric min-sum median clustering. *J. ACM*, 49(2): 139–156.
- Roth, D.; and Yih, W. 2005. Integer linear programming inference for conditional random fields. In Raedt, L. D.; and Wrobel, S., eds., *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*, volume 119 of *ACM International Conference Proceeding Series*, 736–743. ACM.
- Zhang, Z.; Chen, X.; Liu, L.; Chen, J.; Huang, J.; and Feng, Q. 2024. Parameterized Approximation Schemes for Fair-Range Clustering. In Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J. M.; and Zhang, C., eds., *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Zhang, Z.; Lange, K.; and Xu, J. 2020. Simple and Scalable Sparse k-means Clustering via Feature Ranking. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.