A refined complexity analysis of fair districting over graphs

Niclas Boehmer[1] · Tomohiro Koana[1] · Rolf Niedermeier[1]

Abstract

We study the NP-hard FAIR CONNECTED DISTRICTING problem recently proposed by Stoica et al. [AAMAS 2020]: Partition a vertex-colored graph into k connected components (subsequently referred to as districts) so that in every district the most frequent color occurs at most a given number of times more often than the second most frequent color. FAIR CONNECTED DISTRICTING is motivated by various real-world scenarios where agents of different types, which are one-to-one represented by nodes in a network, have to be partitioned into disjoint districts. Herein, one strives for "fair districts" without any type being in a dominating majority in any of the districts. This is to e.g. prevent segregation or political domination of some political party. We conduct a fine-grained analysis of the (parameterized) computational complexity of FAIR CONNECTED DISTRICTING. In particular, we prove that it is polynomial-time solvable on paths, cycles, stars, and caterpillars, but already becomes NP-hard on trees. Motivated by the latter negative result, we perform a parameterized complexity analysis with respect to various graph parameters including treewidth, and problem-specific parameters, including, the numbers of colors and districts. We obtain a rich and diverse, close to complete picture of the corresponding parameterized complexity landscape (that is, a classification along the complexity classes FPT, XP, W[1]-hard, and para-NP-hard).

Keywords Parameterized Algorithmics · Electoral Districting · Fairness · Social Choice on Social Networks

---

✉ Niclas Boehmer
niclas.boehmer@tu-berlin.de

Tomohiro Koana
tomohiro.koana@tu-berlin.de

1 Algorithmics and Computational Complexity, Technische Universität Berlin, Berlin 10623, Berlin, Germany

1 Introduction

Stoica et al. [48] recently introduced graph-based problems on fair (re)districting, employing "margin of victory" as the measure of fair representation. In their work, they performed theoretical and empirical studies; the latter clearly supporting the practical relevance of these problems. The main contribution of their work is certainly with respect to modeling and performing promising empirical studies (based on greedy heuristics). In this paper, we instead focus on the theoretical aspects, significantly extending their findings in this direction.

Dividing agents into groups is a ubiquitous task. Electoral districting is one of the prime examples: Voters are partitioned into voting districts, each electing its own representative.[1] Another example emerges in education; in many countries, children are assigned to schools based on their residency. In such scenarios, the agents (in the settings above, voters or school children) are often placed on a (social or geographical) network. When assigning them to districts, it is natural to require that every district should be connected in the network and meet some further criteria.

In districting, there are various objectives. What we study here can be interpreted as a "benevolent" counterpart of the well-studied gerrymandering scenario in voting theory. For gerrymandering, every voter is characterized by their projected vote in the upcoming election. The goal is then to find a partition of the voters into connected districts such that some designated alternative gains the majority in as many districts as possible. Following Stoica et al. [48], we consider an opposite objective. That is, we assume that some central authority wishes to partition the agents, which are of different types, into connected districts that are *fair*, where a district is deemed fair if the margin of victory in the district is smaller than a given bound. The *margin of victory* of a district is the minimum number of agents whose deletion results in a tie between the two most frequent types in the district. When it comes to school districts, sociodemographic attributes such as race, gender, and religion may be modeled. Here, a low margin of victory would be desirable because a majority of students sharing certain attributes may result in significant funding disparities between schools, as claimed by EdBuild [19] (see Stoica et al. [48] for a more extensive discussion). In electoral districting where agents' types can represent their projected vote or ethnicity, a low margin of victory may foster competition among politicians, thereby motivating elected officials to do a great job. To illustrate that districts that are dominated by a certain ethnicity are a serious problem in particular in developing countries, we quote the two Noble price winners Banerjee and Duflo [4, pp. 251–252]

> There is reason to be concerned that voting [in developing countries] is often based on ethnic loyalties, which means that the candidate from the largest ethnic group often wins, whatever his intrinsic merit. [...] [I]f voters choose based on ethnicity rather than on merit, the quality of candidates representing the majority group will suffer: These candidates don't need to make much of an effort because the fact that they are from the "right" caste or ethnic group is sufficient to ensure that they are elected.

Banerjee and Duflo [5] even found evidence that in the 1980 s and 1990 s in North India elected official that belong to the (clearly) dominating caste group were significantly more

---

[1] One prominent example of electoral districting are the congressional districts in the United States: The US is divided into 435 congressional districts each electing one member of the House of Representatives. How these districts are drawn is the topic of an ongoing debate [13, 21, 27, 37].

likely to be corrupt. This illustrates the practical importance of creating districts with a low margin of victory (in terms of ethnicity). The importance of this problem is further stressed by recent finding of Zhao et al. [50], who provided evidence that the 2021 Georgia Congressional Districting Plan was designed in a way to have a high margin of victory in each district (thereby making elections in each district non-competitive and non-responsive to change in the preferences of voters).

In our work, we build upon the studies of Stoica et al. [48] to search for tractable special cases of fair districting over graphs. We focus on the FAIR CONNECTED DISTRICTING (FCD) problem (a natural special case of Stoica et al.'s FAIR CONNECTED REGROUPING problem). The input of FCD consists of a graph $G = (V, E)$ in which every vertex is assigned a color from a set C, and integers k, ℓ, $s_{\min}$, and $s_{\max}$. The question is whether the vertex set of G can be partitioned into k connected districts, each containing between $s_{\min}$ and $s_{\max}$ vertices, whose margin of victory is at most ℓ. The difference to FAIR CONNECTED REGROUPING is that FCD does not impose any constraints to which districts an agent can be assigned.[2]

It is easy to see that FCD generalizes the known NP-hard PERFECTLY BALANCED CONNECTED PARTITION problem [14, 18], which asks for a partition of a graph into two connected components of the same size (see Proposition 1). This motivates a parameterized complexity analysis and the study of restrictions of the underlying graph in order to identify tractable special cases. Specifically, we analyze the computational complexity of FCD on specific graph classes and the parameterized complexity of FCD with respect to several problem-specific parameters (such as $|C|$ and k) as well as several parameters measuring structural properties of the underlying graph (such as its treewidth or vertex cover number).

1.1 Related work

1.1.1 Relation to model studied by Stoica et al. [48]

Stoica et al. [48] introduced FAIR CONNECTED REGROUPING, which is a generalization of our FCD problem. FAIR CONNECTED REGROUPING differs from FCD in that, in FAIR CONNECTED REGROUPING, one is additionally given a function that specifies for each vertex to which district it can belong. They proved that FAIR CONNECTED REGROUPING is NP-hard even for only two colors and two districts. Moreover, Stoica et al. [48] considered two special cases of FAIR CONNECTED REGROUPING: FAIR REGROUPING (omitting connectivity constraints) and FAIR REGROUPING_X (omitting connectivity constraints and any restriction to which districts vertices can belong). They proved that FAIR REGROUPING is NP-hard for three colors but in XP with respect to the number of districts. Turning to FAIR REGROUPING_X, they showed that the problem is in XP with respect to the number of colors or districts, but left open the general complexity. Overall, our problem is a special case of FAIR CONNECTED REGROUPING, incomparable to FAIR REGROUPING, and a generalization of FAIR REGROUPING_X. Thus, there are no direct implications of the studies of Stoica et al. for our problem.

---

[2] We mention that Stoica et al. [48] use a slightly different yet in most cases equivalent definition of margin of victory. See Sect. 1.1.1 for a detailed discussion. Clearly, there are also other fairness measures for assessing the fairness of districts different from the margin of victory, e.g., the difference between the occurrences of the most and least frequent color. While these are natural as well, our definition is particularly appealing if each district may be only associated with its most frequent color if its margin of victory is too high. Notably, margin of victory is also a popular concept in other domains such as group identification [9], tournament solutions [12], and voting theory [17, 49].

Moreover, our problem differs from the ones studied by Stoica et al. [48] in a slightly different definition of margin of victory. While we look at the number of vertices that need to be deleted to have a tied most frequent color, they examine the number of vertices that need to change their color such that the most frequent color changes. We chose our definition in order to be able to distinguish the case of two tied most frequent colors from the case where one color appears once more than the others (which both have margin of victory one in the model of Stoica et al. [48]). In all other cases, if m is the margin of victory in our definition, then the margin of victory in the definition of Stoica et al. [48] is $\lfloor \frac{m}{2} \rfloor$. However, all our algorithmic results can be extended to the definition of Stoica et al. [48]. We expect that hardness results similar to ours can be obtained for their definition as well.

1.1.2 Further related work

Following up on the work of Stoica et al. [48], Boehmer and Koana [10] analyzed the FAIR REGROUPING and the FAIR REGROUPING_X problem in more detail. Among others, they proved that FAIR REGROUPING_X without size constraints is polynomial-time solvable while the NP-hard FAIR REGROUPING problem without size constraints is polynomial-time solvable for two colors and fixed-parameter tractable with respect to the number of districts. In addition to the margin of victory, they also considered the maximum difference between the occurrences of two colors in a district as an additional fairness notion. Again, their results have no implications on the complexity of our FCD problem.

FCD is relevant in district-based elections, where voters are partitioned into districts and each district elects its own representative. Several papers have studied how to assign voters to districts so as to "fairly" reflect the political choices of voters [3, 34, 35, 43, 44]. Well-studied in this context is gerrymandering, which can be regarded as a "malicious" counterpart to our problem. In gerrymandering, the task is to partition a set of voters into districts obeying certain conditions such that a designated alternative wins in as many districts as possible. An intuitive strategy to solve this problem, which is not necessarily optimal [45], is to maximize the number of districts where the designated alternative wins only by a small margin (this is somewhat related to our problem.) Initially, gerrymandering has been predominantly studied from the perspective of social and political science [23, 28, 40]. More recently, different variants of gerrymandering have been considered from an algorithmic perspective [20, 38]. Notably, the study of gerrymandering over graphs, which is analogous to our problem, has recently gained significant interest [6, 15, 26, 29]. In particular, as done here for FCD, Bentert et al. [6], Gupta et al. [26], and Ito et al. [29] analyzed the complexity of gerrymandering on paths, cycles, and trees and studied the influence of the number of candidates/colors and the number of districts. A similar model for graph-based redistribution scenarios and political districting has been studied under the name "network-based vertex dissolution" [7].

Partitioning agents of different types into balanced groups is conceptually closely related to studies of social segregation. In computer science, social segregation is, for instance, quite extensively studied in the context of Schelling's segregation model [47]: While initially the work in this context was mostly concerned with the theoretical analysis of segregation patterns [8, 11], recently Schelling's model has been approached from a game-theoretic perspective [1, 33].

1.2 Contribution

Motivated by the NP-hardness of FCD (see Proposition 1), we conduct a parameterized complexity analysis of FCD and study restrictions of the underlying graph in order to identify tractable special cases. We investigate the influence of problem-specific parameters (the number $|C|$ of colors, the number k of districts, and the margin of victory ℓ) and the structure of the underlying graph on the computational complexity of FCD.

The motivation for our refined complexity analysis is two-fold. First, analyzing special graph classes and types of graphs offers a better understanding of the complexity of FCD: Path, stars, and cycles form the building blocks of more complex graphs and thus understanding the problem on these graphs is vital for a further analysis.[3] Moreover, real-world networks will usually have some structure. One specific use case for our algorithms might be sampling algorithms for "fair" districting plans. These sampling algorithms often work by pre-aggregating groups of voters, creating a spanning tree on the merged space, and then continuing by further merging adjacent groups [2, 16, 42]. Here, our analysis of FCD on trees and tree-like graphs might be useful. Second, considering the analysis of the influence of the number $|C|$ of colors and the number k of districts, in most applications we can think of these two parameters are substantially smaller than the number of vertices, motivating to check whether FCD becomes tractable if one (or both) of these parameters are small. For instance, in their experiments, Stoica et al. [48] partitioned 50, 000 voters into 10 voting districts and 41 834 schoolchildren into 61 school districts with $|C| = 7$.

We show that FCD is NP-hard even if $|C| = k = 2$ and $\ell = 0$ but polynomial-time solvable on paths, cycles, stars, and caterpillars (for stars, our algorithm even runs in linear time). Subsequently, we extend our polynomial-time algorithms for paths and cycles to a polynomial-time algorithm for all graphs with a constant max leaf number (mln), which are basically graphs that consist of a constant number of paths and cycles (where the two endpoints of each path and one point from each cycle can be arbitrarily connected).

Remarkably, in our most involved hardness reduction, we show that FCD already becomes NP-hard and even W[1]-hard with respect to $|C| + k$ on trees. However, when the number of colors or the number of districts is constant, FCD on trees becomes polynomial-time solvable. In fact, we show that these results hold for some tree-like graphs as well. Herein, the tree-likeness of a graph is measured by one of three parameters, namely, the treewidth (tw), the feedback edge number (fen), and the feedback vertex number (fvn). More precisely, as our most involved algorithmic results, we establish polynomial-time solvability of FCD when the number of colors and the treewidth are constant. We achieve this with a dynamic programming approach on the tree decomposition of the given graph empowered by some structural observations on FCD. Moreover, we observe that there is a simple polynomial-time algorithm on graphs with a constant feedback edge number when there are a constant number of districts. On the other hand, we prove that FCD is NP-hard for two districts even on graphs with fvn = 1 (and tw = 2). Lastly, we show that FCD is polynomial-time solvable on graphs with a constant vertex cover number (vcn) and fixed-parameter tractable with respect to the vertex cover number and the number of colors. A

---

[3] This also motivated a study of these graph classes for gerrymandering over graphs [6, 26, 29]. Moreover, initially, it is not at all clear that FCD is polynomial-time solvable on these graphs, as, for instance, gerrymandering over graphs is NP-hard even on paths [6, Theorem 1] That we establish that FCD is polynomial-time solvable on paths also proves that FCD is sometimes easier than the corresponding gerrymandering over graphs problem.
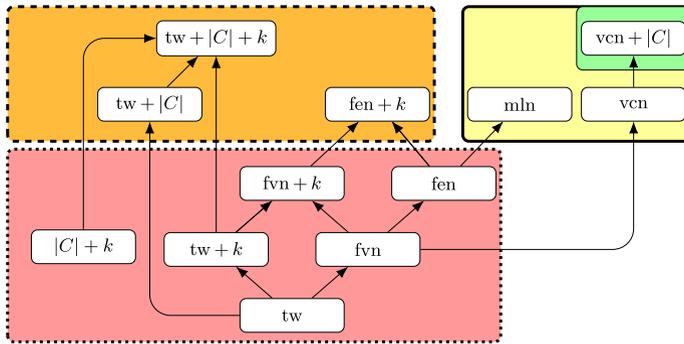
Fig. 1 Overview of our parameterized complexity results. Each box represents one parameterization of FCD. An arc from parameter p to another parameter p' indicates that p is upper-bounded by some function of p'. For parameters in the red area (dotted), we prove that FCD is NP-hard even if the parameter is a constant. For parameters in the orange area (dashed), we prove W[1]-hardness and present an XP-algorithm. For parameters in the yellow area (solid thick), we have an XP-algorithm but W[1]-hardness is unknown. The green area (solid) indicates fixed-parameter tractability (Color figure online)

summary of our parameterized results can be found in Fig. 1. Notably, all our hardness results also hold without size constraints.

In our studies, we identify several sharp complexity dichotomies. For instance, FCD is polynomial-time solvable on trees with diameter at most three but NP-hard and W[1]-hard with respect to $|C| + k$ on trees with diameter four. Similarly, FCD is NP-hard and W[1]-hard with respect to $|C| + k$ on graphs with pathwidth at least two but polynomial-time solvable on pathwidth-one graphs.

To summarize, we show that FCD without size constraints is NP-hard even in very restricted settings, e.g., on trees or if $|C| = k = 2$ and $\ell = 0$. To make the problem tractable, one possibility is to significantly restrict the input graph, e.g., to consist of a constant number of paths and cycles, or to combine structural parameters of the given graph with the number $|C|$ of colors or the number k of districts. For small $|C|$ and k, the tractability of FCD extends to certain tree-like graphs and graphs with a small vertex cover number. In contrast to the parameters $|C|$ and k, which have a strong influence on the complexity of FCD, the bound ℓ on the margin of victory has only little impact as all hardness results already hold for $\ell = 0$ and all our algorithmic results hold for arbitrary ℓ.

1.2.1 Organization

The remainder of this paper is structured as follows. In Sect. 2, we formally introduce FCD and the graph parameters we examine. In Sect. 3, we present some preliminary results on FCD mostly concerning NP-hardness. In Sect. 4, we then consider FCD on very special graph classes such as paths and cycles as well as on graphs that can be partitioned into a bounded number of those. In Sect. 5, we shift our attention to trees and graphs that are tree-like. Lastly, in Sect. 6, we analyze the influence of the vertex cover number on the complexity of FCD. We defer the proofs of two technical results (marked with ★) to the appendix.

2 Preliminaries

For $a, b \in \mathbb{N}$, let $[a, b]$ denote $\{a, a + 1, \ldots, b - 1, b\}$ and let $[b]$ denote $[1, b]$. Throughout the paper, all graphs $G = (V, E)$ are undirected and have no self-loops or multi-edges. By convention, we will use $n := |V|$. Given a graph $G = (V, E)$ and a vertex set $V' \subseteq V$, let $G[V']$ be the graph G induced by the vertices from V'.

2.1 Fair connected districting

Let $C = \{c_1, \ldots, c_{|C|}\}$ be the set of colors. We assume that each vertex $v \in V$ of the given graph has a color $c \in C$ determined by a given coloring function $\mathrm{col} : V \to C$. For a vertex set $V' \subseteq V$, let $\chi_c(V')$ denote the number of vertices of color c in V'. Moreover, let $\chi(V') \in \mathbb{N}^{|C|}$ denote the vector in which the ith entry contains the number of vertices of color c_i in V', i.e, $\chi_i(V') = \chi_{c_i}(V')$. For a vector $\mathbf{x} = (x_1, \ldots, x_t) \in \mathbb{N}^t$, let $i^*_\mathbf{x}$ denote an index with the largest entry in $\mathbf{x}$, i.e., $i^*_\mathbf{x} \in \arg\max_{i \in [t]} x_i$. The *margin of victory* $\mathrm{MOV}(\mathbf{x})$ of a vector $\mathbf{x}$ is defined as the difference between the largest and second largest entry in $\mathbf{x}$, i.e., $\mathrm{MOV}(\mathbf{x}) := x_{i^*_\mathbf{x}} - \max_{i \in [t] \setminus \{i^*_\mathbf{x}\}} x_i$. If $t = 1$, then we set $\mathrm{MOV}(\mathbf{x}) := x_1$. Accordingly, we define the margin of victory $\mathrm{MOV}(V')$ of a vertex set $V' \subseteq V$ of colored vertices as $\mathrm{MOV}(V') := \mathrm{MOV}(\chi(V'))$. For $\ell \in \mathbb{N}$, we call a vertex set V' ℓ-*fair* if $\mathrm{MOV}(V') \leq \ell$. We use the term *district* to refer to a vertex set $V' \subseteq V$. We say that a color $c_i \in C$ is the jth most frequent color in a district V' if $\chi_i(V')$ is the jth largest entry in $\chi(V')$ (we break ties arbitrarily unless stated otherwise). We now present our central problem[4]:

FAIR CONNECTED DISTRICTING (FCD)

Input: A graph $G = (V, E)$, a set C of colors, a function $\mathrm{col} : V \to C$ assigning each vertex one color from C, a number $k \leq |V|$ of districts, a maximum margin of victory ℓ, and two integers $s_{\max} \geq s_{\min} \geq 1$.

Question: Does there exist a partition of the vertices into k districts $(V_1, \ldots, V_k)$ such that, for all $i \in [k]$, V_i is ℓ-fair, $|V_i| \in [s_{\min}, s_{\max}]$, and $G[V_i]$ is connected?

2.2 Graph parameters

We define several graph parameters for an undirected graph $G = (V, E)$. The *diameter* of G is the maximum shortest distance between any pair of vertices. The *max leaf number* $\mathrm{mln}\,(G)$ (for a connected graph G) is the maximum number of leaves over all spanning trees of G. A vertex set $V' \subseteq V$ is a *vertex cover* of G if $G[V \setminus V']$ has no edge. A vertex

---

[4] Note that we chose to require that all districts are non-empty to be consistent with the closely related literature on gerrymandering over graphs [15, 29]. Moreover, our variant can easily be used to decide whether there exists a partitioning into at most k non-empty districts by simply iterating over all $i \in [k]$. For the reverse direction, this is not easily possible. For instance, consider a clique with two blue vertices and one green and one red vertex and $\ell = 0$. Then, a partitioning into two non-empty ℓ-fair districts is possible but not into any other number of non-empty districts.

set $V' \subseteq V$ is a *feedback vertex set* if $G[V \setminus V']$ is a forest. Analogously, an edge set $E' \subseteq E$ is a *feedback edge set* if $(V, E \setminus E')$ is a forest. The *vertex cover number* vcn (G), *feedback vertex number* fvn (G), and *feedback edge number* fen (G) is the size of a smallest vertex cover, feedback vertex set, and feedback edge set, respectively. A *tree decomposition* of a graph $G = (V, E)$ is a pair $\left(T, \{B_x\}_{x \in V_T}\right)$, where $T = (V_T, E_T)$ is a rooted tree and $B_x \subseteq V$ for each $x \in V_T$ such that

(i) $\bigcup_{x \in V_T} B_x = V$,
(ii)   for each edge $\{u, v\} \in E$, there is an $x \in V_T$ with $u, v \in B_x$, and
(iii)   for each $v \in V$, the set of nodes $x \in V_T$ with $v \in B_x$ induces a connected subtree in T.

The *width* of $(T, \{B_x\}_{x \in V_T})$ is $\max_{x \in V_T} |B_x| - 1$. The *treewidth* tw (G) of G is the minimum width of all tree decompositions of G. The *pathwidth* pw (G) is defined analogously with the additional constraint that T is a path.

If G is clear from context, then we simply omit it and write mln, vcn, fvn, fen, tw, and pw, respectively. For all our parameterized algorithms with respect to one of these parameters, we assume that the corresponding structure is given as part of the input. For instance, in our algorithm parameterized by tw, we assume that we are given a tree decomposition of width tw. However, note that in all cases, the worst-case running time of our algorithms would not change if we compute the structure using known parameterized algorithms.

2.3 Parameterized complexity theory

A *parameterized problem L* consists of a problem instance $\mathcal{I}$ and a parameter value $k \in \mathbb{N}$. Then L lies in XP with respect to k if there exists an algorithm deciding L in $|\mathcal{I}|^{f(k)}$ time for some computable function f. Furthermore, L is called *fixed-parameter tractable* with respect to k if there exists an algorithm deciding L in $f(k)|\mathcal{I}|^{\mathcal{O}(1)}$ time for a computable function f. The corresponding complexity class is called FPT. There is a hierarchy of complexity classes for parameterized problems: FPT$\subseteq$ W[1] $\subseteq$ W[2] $\subseteq$ XP, where it is commonly believed that all inclusions are strict. Thus, if L is shown to be W[1]-hard, then the common belief is that it is not fixed-parameter tractable. For instance, computing a clique of size at least k in a graph is known to be W[1]-hard with respect to the parameter k. One can show that L is W[t]-hard, $t \geq 1$, by a *parameterized reduction* from a known W[t]-hard parameterized problem L'. A parameterized reduction from L' to L is a function that maps an instance $(\mathcal{I}', k')$ of L' to an instance $(\mathcal{I}, k)$ of L such that $(\mathcal{I}', k')$ is a yes-instance for L' if and only if $(\mathcal{I}, k)$ is a yes-instance for L. Moreover, we require that k is bounded in a function of k' and that the transformation takes at most $f(k')|\mathcal{I}|^{\mathcal{O}(1)}$ time for some computable function f. Finally, we say that a parameterized problem is *para-NP-hard* if it is NP-hard even for constant parameter values.

3 Basic results on FCD

In this section, we make some basic observations on the computational complexity of FCD. First, we prove that FCD is para-NP-hard with respect to the combination $|C| + k + \ell$ of all three problem-specific parameters. This strong general hardness result motivates the study of various restricted graph classes in subsequent sections.

The NP-hardness of FCD easily follows from the fact that it generalizes the Perfectly Balanced Connected Partition problem, which is NP-hard on bipartite graphs [18]:

Perfectly Balanced Connected Partition

Input:　A graph $G = (V, E)$.

Question:　Is there a partition (V_1, V_2) of the vertex set V such that $G[V_1]$ and $G[V_2]$ are connected and $|V_1| = |V_2|$?

Consider the following polynomial-time many-one reduction: Given an instance $G = (V, E)$ of Perfectly Balanced Connected Partition, construct an equivalent instance of FCD by coloring all vertices in G in the same color, setting $k = 2$, $\ell = |V|/2$, $s_{\min} = 1$, and $s_{\max} = \infty$. Moreover, it is also possible to prove the NP-hardness for the case with $\ell = 0$ at the cost of introducing a second color:

Proposition 1 *FCD is NP-hard even if G is bipartite, $s_{\min} = 1$, $s_{\max} = \infty$, and (i) $|C| = 1$, $k = 2$ or (ii) $\ell = 0$, $|C| = 2$, $k = 2$.*

Proof To prove the second part, we present a polynomial-time reduction from Perfectly Balanced Connected Partition to FCD. Notably Perfectly Balanced Connected Partition is also NP-hard if we are given two vertices $v_1, v_2 \in V$ and the question is whether there is a partition (V_1, V_2) with $v_1 \in V_1$ and $v_2 \in V_2$ (in the hardness proof by Dyer and Frieze [18, Theorem 2.2], the vertex a always belongs to one partition and b to the other).

Construction. Given an instance $(G = (V, E), v_1, v_2)$ of Perfectly Balanced Connected Partition with $n := |V|$, we construct an instance of FCD: We set $C = \{c_1, c_2\}$, $k = 2$, and $\ell = 0$. We color all vertices of the given graph G in color c_1. Moreover, we modify G by introducing n new vertices of color c_2. We connect half of these vertices to v_1 and the other half to v_2.

Proof of Correctness. Let (V_1, V_2) be a solution to the FCD instance. As $\ell = 0$ and we require that both V_1 and V_2 need to be non-empty, it needs to hold that both districts contain vertices of both colors. Moreover, as both $G[V_1]$ and $G[V_2]$ are connected, all vertices of color c_2 adjacent to v need to be in the same district as v_1 (and similarly for v_2). Thus, v_1 and v_2 need to be in different districts, each consisting of $\frac{n}{2}$ vertices of color c_2 and $\frac{n}{2}$ vertices of color c_1. Thereby, after removing all vertices of color c_2, (V_1, V_2) is a solution to the given Perfectly Balanced Connected Partition instance.

To prove the reverse direction, let (V_1, V_2) be a solution to the given Perfectly Balanced Connected Partition instance with $v_1 \in V_1$ and $v_2 \in V_2$. Then, the FCD instance is a yes-instance, as it is possible to add the $\frac{n}{2}$ vertices of color c_2 attached to v_1 to V_1 and the $\frac{n}{2}$ vertices of color c_2 attached to v_2 to V_2 to arrive at a solution to the FCD instance.　□

Note that our results strengthen a result of Stoica et al. [48], who proved that FCD is NP-hard for $|C| = 2$ and $k = 2$ if we can additionally specify for each vertex the districts to which it can be assigned.

As our second basic result, using a simple dynamic programming approach, we show that an instance of FCD on a disconnected graph is polynomial-time solvable if FCD can be solved in polynomial time on its connected components. This will play a crucial role, e.g., in developing an XP-algorithm for the max leaf number (Theorem 1).

Proposition 2 *Let $\mathcal{G}$ be a class of graphs such that* FCD *is polynomial-time solvable on any graph $G \in \mathcal{G}$. Then,* FCD *is polynomial-time solvable on any graph from $\mathcal{G}'$, where $\mathcal{G}\simeq$ is the class of graphs obtained by taking disjoint unions of graphs from $\mathcal{G}$.*

Proof Let $G' \in \mathcal{G}'$ and let $G_1, \dots, G_p \in \mathcal{G}$ be the connected components of G'. Clearly, every district is contained in the vertex set of G_i for some $i \in [p]$. We solve the problem using a simple subset-sum like dynamic programming algorithm. To this end, we introduce a table $T[i, j]$ for $i \in [p]$ and $j \in [k]$. An entry $T[i, j]$ is true if one can partition the vertices of $G_1, \dots, G_i$ into j connected ℓ-fair districts respecting the size constraints. We also use a table H where $H[i, j]$ for $i \in [p]$ and $j \in [k]$ is true if the vertex set of G_i can be partitioned into j connected ℓ-fair districts respecting the size constraints. Note that H can be computed in polynomial time by our initial assumption on G.

To initialize T, we set $T[1, j]$ to $H[1, j]$ for all $j \in [k]$. Subsequently, for increasing $i > 1$, we update T as follows:

$$T[i,j] = \bigvee_{\substack{j',j'' \in [j] \\ j' + j'' = j}} T[i - 1, j'] \wedge H[i, j''].$$

In the end, we return $T[p, k]$. This algorithm runs in $\mathcal{O}(p \cdot k^2) = \mathcal{O}(n^3)$ time (aside from the computation of H). □

4 FCD on paths, cycles, and beyond

This section studies the computational complexity of FCD on simple graphs and graphs that can be partitioned into "few" paths and cycles. Specifically, in Sect. 4.1, we develop polynomial-time algorithms on paths, cycles, stars and caterpillars. Subsequently, in Sect. 4.2, we show that FCD is in XP when parameterized by the max leaf number, which generalizes polynomial-time solvability on paths and cycles.

4.1 Polynomial-time algorithms for FCD on simple graph classes

We start by proving that FCD is cubic-time solvable on paths using a simple dynamic programming approach.

Proposition 3 *FCD on paths can be solved in $\mathcal{O}(k \cdot n^2)$ time.*

Proof Let $G = \big(\{v_1, \ldots, v_n\}, \{\{v_i, v_{i+1}\} \mid i \in [n-1]\}\big)$ be the input path. We first create a table $A[i, j]$, where $A[i, j]$ for $i \leq j \in [n]$ is true if and only if $\{v_i, \ldots, v_j\}$ is ℓ-fair and $|\{v_i, \ldots, v_j\}| = j - i + 1 \in [s_{\min}, s_{\max}]$. We then create a table T with entries $T[i, t]$ for $i \in [n]$ and $t \in [k]$. The meaning of an entry $T[i, t]$ is that it is true if there is a partition of the vertices $\{v_1, \ldots, v_i\}$ into t paths (connected districts) that are all ℓ-fair and respect the size constraints. Note that the input is a yes-instance if and only if $T[n, k]$ is true.

We initialize table T by setting $T[i, 1]$ for $i \in [n]$ to $A[1, i]$. Subsequently, for $t > 1$, we update the table using:

$$T[i, t] = \bigvee_{j \in [i-1]} T[j, t-1] \wedge A[j+1, i].$$

The reasoning behind this is that if we partition $\{v_1, \ldots, v_i\}$ into t districts, then there needs to be some $j \in [i-1]$ such that $\{v_{j+1}, v_{j+2}, \ldots, v_i\}$ is one district in the solution. Thus, if $T[i, t]$ is true, then there needs to be some $j \in [i-1]$ such that the vertices from $\{v_1, \ldots, v_j\}$ can be partitioned into $t - 1$ ℓ-fair districts respecting the size constraints and the vertices $\{v_{j+1}, v_{j+2}, \ldots, v_i\}$ form an ℓ-fair district respecting the size constraints.

The table A can be filled in $\mathcal{O}(n^2)$ time: for a fixed $i \in [n]$, we can compute all $A[i, n-j]$ for $j \in [0, n-i]$ in linear time by starting for $j = 0$ with computing $\chi(\{v_i, \ldots, v_{n-j}\})$ and, subsequently, for increasing $j > 1$ compute $\chi(\{v_i, \ldots, v_{n-j}\}) = \chi(\{v_i, \ldots, v_{n-j+1}\}) - \chi(\{v_{n-j+1}\})$ (which can be done in constant time). It can be determined in $O(1)$ time whether $\chi(\{v_i, \ldots, v_{n-j}\})$ is ℓ-fair by additionally storing the number of occurrences of integers (which are at most n) in $\chi(\{v_i, \ldots, v_{n-j}\})$. Additional bookmarking allows us to find the two largest entries in $\chi(\{v_i, \ldots, v_{n-j}\})$ in $\mathcal{O}(1)$ time. Moreover, the number of occurrences in $\chi(\{v_i, \ldots, v_{n-j}\})$ can be updated in $\mathcal{O}(1)$ time as we increase j. We set $A[i, n-j]$ to true if and only if $\chi(\{v_i, \ldots, v_{n-j}\})$ is ℓ-fair and $s_{\min} \leq n - i - j + 1 \leq s_{\max}$.

Note that table T consists of $k \cdot n$ entries. We spend $\mathcal{O}(n)$ time for every entry, resulting in an overall running time of $\mathcal{O}(k \cdot n^2)$. □

To solve FCD on cycles, we iterate over all vertices of the cycle as the starting point of the first district and split the cycle at this point to convert it into a path. Subsequently, we employ the algorithm from above, which results in a running time of $\mathcal{O}(k \cdot n^3)$:

Corollary 1 *FCD on cycles is solvable in $\mathcal{O}(k \cdot n^3)$ time.*

Next, we proceed to stars, for which we derive a precise characterization of yes-instances. In fact, using a relatively involved analysis, we consider a more general problem, in which some set X containing the center vertex must belong to the same district, i.e., the vertex which is adjacent to all other vertices from the star. This proves useful in speeding up the algorithms to be presented in Propositions 5 and 7.

Proposition 4 (★) *Let $(G = (X \cup Y, E), C, \mathrm{col}, k, \ell, s_{\min}, s_{\max})$ be an FCD instance where G is a star with center vertex $v \in X$. Then, one can decide in linear time whether there is a partition of $X \cup Y$ into k connected ℓ-fair districts respecting $s_{\min}$ and $s_{\max}$ such that all vertices from X are part of the same district.*

Proposition 4 directly implies that FCD is linear-time solvable on stars, even if the center vertex has a weight for each color, i.e., the center vertex represents multiple vertices that all need to be put into one district.

Corollary 2 *FCD is linear-time solvable on stars.*

Lastly, we extend our polynomial-time algorithm for paths from Proposition 3 to caterpillar graphs. A caterpillar graph is a tree where every vertex is either on a central path (spine) or a neighbor of a vertex on the central path.

Proposition 5 *FCD on caterpillars can be solved in $\mathcal{O}(k \cdot n^3)$ time.*

Proof Let $G = (V, E)$ be the given caterpillar, let $(u_1, \ldots, u_p)$ denote the spine (central path) of G and let $U = \{u_1, \ldots, u_p\}$. Moreover, for $i \in [p]$, let U_i consist of u_i and all vertices from $V \setminus U$ adjacent to u_i. Note that each vertex from $V \setminus U$ is only adjacent to one vertex from U, as G is in particular a tree. We solve the problem by extending our algorithm for paths from Proposition 3.

As a first step, we introduce a table $A[i, j, t]$ for $i \leq j \in [n]$ and $t \in [k]$. An entry $A[i, j, t]$ is set to true if there is a partition $(V_1, \ldots, V_t)$ of $\bigcup_{i' \in [i,j]} U_{i'}$ into t districts such that:

- V_1 contains $u_{i'}$ for every $i' \in [i, j]$, and
- $V_{t'}$ is ℓ-fair, $|V_{t'}| \in [s_{\min}, s_{\max}]$, and $G[V_{t'}]$ is connected for every $t' \in [t]$.

We can fill table A in $\mathcal{O}(k \cdot n^3)$ time using Proposition 4 for each $A[i, j, t]$ (for this, it is necessary to slightly restructure $G[\bigcup_{i' \in [i,j]} U_{i'}]$ as a star G' on $\bigcup_{i' \in [i,j]} U_{i'}$ with u_i as the center vertex and $X = \{u_i, \ldots, u_j\}$). Using A, we now apply dynamic programming. To this end, we introduce a table $T[i, t]$ for $i \in [n]$ and $t \in [k]$. Entry $T[i, t]$ is set to true if it is possible to partition the vertices from $\bigcup_{j \in [i]} U_j$ into t connected districts that are ℓ-fair and respect the size constraints. We initialize the table by setting $T[i, 1]$ to true if $\bigcup_{j \in [i]} U_j$ is ℓ-fair. Subsequently, for $t > 1$, we update the table using:

$$T[i, t] = \bigvee_{\substack{j \in [i-1] \wedge t', t'' \in [t]: \\ t' + t'' = t}} T[j, t'] \wedge A[j + 1, i, t''].$$

The reasoning behind this equality is that in a partitioning of $\bigcup_{j \in [i]} U_j$ into t districts, there needs to be some $j \in [i - 1]$ such that there is one district containing vertices $\{u_{j+1}, \ldots, u_i\}$. However, not all vertices from $\bigcup_{t \in [j+1,i]} U_t$ need to be part of this district. In fact, because of the connectivity constraints, there is some t'' such that one district contains vertices $\{u_{j+1}, \ldots, u_i\}$ and all but $t'' - 1$ vertices from $\bigcup_{t \in [j+1,i]} U_t$ and $t'' - 1$ districts consist of a single vertex from $\bigcup_{t \in [j+1,i]} U_t$ (such a partitioning respecting fairness and size constraints exists if and only if $A[j + 1, i, t'']$ is true). The remaining $t - t''$ districts then consists of vertices from $\bigcup_{t \in [j]} U_t$ (such a partitioning respecting fairness and size constraints exists if and only if $T[j, t']$ is true). Table T has $k \cdot n$ entries each computable in $\mathcal{O}(k \cdot n)$ time. This leads to an overall running time of $\mathcal{O}(k \cdot n^3)$. □

Note that a graph G has pathwidth one ($\mathrm{pw}\,(G) = 1$) if and only if G is a disjoint union of caterpillars. By Proposition 2 and Proposition 5, it follows that FCD is

Fig. 2 An illustration of a graph with four branches: $(v_1, v_2, v_3), (v_3, v_5), (v_3, v_4, v_6, v_5), (v_5, v_7, v_8)$
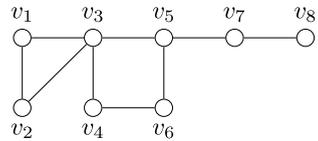
polynomial-time solvable on all graphs with pathwidth one (we later show in Corollary 4 that FCD is NP-hard on pathwidth-two graphs).

Corollary 3 *FCD on a graph G with* $\mathrm{pw}\,(G) = 1$ *is polynomial-time solvable.*

4.2 An XP-algorithm for mln

Now, we generalize the polynomial-time solvability on paths and cycles to a larger graph class. More precisely, we develop an XP-algorithm for the max leaf number (mln). Recall that the max leaf number of a connected graph G is the maximum number of leaves over all spanning trees of G. Notably, any path or cycle has mln $= 2$. In order to develop a polynomial-time algorithm for constant mln, we use the notion of *branches*. (See Fig. 2 for an illustration.)

Definition 1 A *branch* in a graph is either a maximal path in which all inner vertices have degree two or a cycle in which all but one vertex have degree two.

Using a classical theorem of Kleitman and West [31], Eppstein [22] showed that any graph has at most $\mathcal{O}(\mathrm{mln}^2)$ branches. For notational brevity, we assume that every branch B that is a cycle has exactly one endpoint, namely, the vertex in B with degree at least three in G (if there is no such vertex in B, then we fix an arbitrary vertex as its endpoint).

Theorem 1 *FCD can be solved in* $n^{\mathcal{O}(\mathrm{mln}^2)}$ *time.*

Proof Let $\mathcal{B}$ be the set of all branches of the given graph G and let X be the set of all endpoints of all branches. Suppose that there is a solution $\mathcal{V} = (V_1, \ldots, V_k)$. Observe that there are naturally at most $|X|$ subsets in $\mathcal{V}$ containing at least one vertex from X. Without loss of generality, assume that V_i contains at least one vertex from X for $i \in [k']$ for some $k' \in [\min(k, |X|)]$, and let $X_i = X \cap V_i$. Let us now consider the relationship between a district V_i for $i \in [k']$ and a branch $B = (v_1, \ldots, v_l) \in \mathcal{B}$ (where for all $i \in [l-1]$, v_i and v_{i+1} are adjacent in G and if B is a cycle, then $v_1 = v_l$). Since V_i induces a connected subgraph in G, the following holds:

- If $v_1, v_l \notin X_i$, then V_i and B are disjoint: Since V_i is connected and contains at least one vertex from $X \backslash \{v_1, v_l\}$, V_i contains no "inner" vertices from B.
- If $v_1 \in X_i$ and $v_l \notin X_i$, then there is an integer $j \in [l-1]$ such that $V_i \cap \{v_1, \ldots, v_l\} = \{v_1, \ldots, v_j\}$ (for $v_1 \notin X_i$ and $v_l \in X_i$, the situation is symmetric).
- If $v_1, v_l \in X_i$ for some $i \in [k]$, then there are two integers $j, j' \in [l]$ such that $V_i \cap \{v_1, \ldots, v_l\} = \{v_1, \ldots, v_j\} \cup \{v_{l-j'+1}, \ldots, v_l\}$. Note that V_i contains all vertices of B when $j + j' = l$.

For all other districts V_i, $i \in [k' + 1, k]$, it holds that there is a branch $(v_1, \ldots, v_l) \in \mathcal{B}$ with $V_i \subseteq \{v_2, \ldots, v_{l-1}\}$.

Using these observations, our algorithm proceeds as follows. We iterate over all possible combinations of the following (note that the number of combinations is $n^{\mathcal{O}(|\mathcal{B}|)} = n^{\mathcal{O}(\text{mln}^2)}$):

- An integer $k' \in [\min(k, |X|)]$.
- A partition $\mathcal{X} = (X_1, \ldots, X_{k'})$ of X into k' subsets.
- For every branch $B = (v_1, \ldots, v_l) \in \mathcal{B}$, two integers $j_B, j'_B \in [l]$ with $j_B + j'_B \leq l$.

Using these guesses, we can exactly determine the districts intersecting X: For every $i \in [k']$ and every branch $B = (v_1, \ldots, v_l) \in \mathcal{B}$, we define $V_{i,B}$ as follows:

- If $v_1, v_l \notin X_i$, then $V_{i,B} := \emptyset$.
- If $v_1 \in X_i$ and $v_l \notin X_i$, then $V_{i,B} := \{v_1, \ldots, v_{j_B}\}$. Symmetrically, if $v_1 \notin X_i$ and $v_l \in X_i$, then $V_{i,B} := \{v_{l-j'_B+1}, \ldots, v_l\}$.
- If $v_1, v_l \in X_i$, then $V_{i,B} = \{v_1, \ldots, v_{j_B}\} \cup \{v_{l-j'_B+1}, \ldots, v_l\}$.

For $i \in [k']$, let $V_i := \bigcup_{B \in \mathcal{B}} V_{i,B}$. We check whether the set V_i is ℓ-fair and respects the size constraints for every $i \in [k']$. If there is a V_i that is not ℓ-fair or violates the size constraints, then we proceed to the next combination. Otherwise, it remains to determine whether the vertices $V \setminus \bigcup_{i \in [k']} V_i$ can be partitioned into $k - k'$ ℓ-fair districts that respect the size constraints. Since $G[V \setminus \bigcup_{i \in [k']} V_i]$ is a disjoint union of paths as all endpoints of branches are contained in $\bigcup_{i \in [k']} V_i$, this can be done in polynomial time by Propositions 2 and 3.   □

We leave it open whether FCD parameterized by mln is fixed-parameter tractable or W[1]-hard.

5 FCD on trees and tree-like graphs

After having seen in Sect. 4 that FCD is polynomial-time solvable on paths, cycles, stars, and caterpillars, we now turn to trees. In Sect. 5.1, we prove that these polynomial-time results do not extend to trees. In particular, we prove that FCD on trees is NP-hard and W[1]-hard parameterized by $|C| + k$. We complement these hardness results with an XP-algorithm for FCD on trees parameterized by $|C|$ and another XP-algorithm for parameter k. In fact, both XP-algorithms further extend to tree-like graphs: In Sect. 5.2, we prove that FCD parameterized by the treewidth of the given graph plus $|C|$ is in XP. Subsequently, in Sects. 5.3 and 5.4, we consider the feedback edge number (fen) and the feedback vertex number (fvn), which are both alternative measures for the tree-likeness of a graph (the treewidth of a graph can be upper bounded in a function of fen and in a function of fvn). Thus, the XP algorithm for tw $+ |C|$ extends to the parameter combinations fvn $+ |C|$ and fen $+ |C|$, which is why we focus on the parameter combinations fvn $+ k$ and fen $+ k$ here. We prove that FCD parameterized by fen $+ k$ is in XP and we show that there is (presumably) no such result for the treewidth and the feedback vertex number (fvn) by showing that FCD is NP-hard even if fvn $= 1$, tw $= 2$, and $k = 2$.

5.1 W[1]-hardness on trees

In this subsection, we show that in contrast to paths, cycles, stars, and caterpillars, FCD on trees is NP-hard even without size constraints. Simultaneously, we show that FCD parameterized by the number k of districts and the number $|C|$ of colors is W[1]-hard on trees. To this end, we present a parameterized reduction from the following version of GRID TILING which is NP-hard and W[1]-hard with respect to t [41] (in this subsection, all indices are taken modulo t):

GRID TILING

Input: A collection $\mathcal{S}$ of t^2 (tile) sets $S^{i,j} \subseteq [m] \times [m]$, $i, j \in [t]$, where each $S^{i,j}$ consists of n pairs and the first entries of all pairs (tiles) from $S^{i,j}$ sum up to the same number X for all $i, j \in [k]$ and the second entries of all pairs from $S^{i,j}$ sum up to the same number Y for all $i, j \in [k]$.

Question: Can one choose one tile $(x^{i,j}, y^{i,j}) \in S^{i,j}$ for each $i, j \in [t]$ such that $x^{i,j} = x^{i+1,j}$ and $y^{i,j} = y^{i,j+1}$?

Without loss of generality, we assume that $n > 2$.

The general idea of the reduction is as follows. Each solution to the constructed FCD instance has a *center district*. The center district contains some large number Z of vertices of two "dummy" colors c and c'. The instance is constructed such that in every solution, vertices of any fixed color can appear at most Z times in the center district. By definition, each tile (x, y) belongs to one of t^2 tile sets $S^{i,j} \in \mathcal{S}$. We construct a star $T_{x,y}^{i,j}$ for each tile such that for each tile set $S^{i,j} \in \mathcal{S}$, all but exactly one star $T_{x,y}^{i,j}$ need to be contained in the center district. Thus, the center district (respectively its complement) basically encodes a selection of one tile from each tile set. Moreover, we construct the FCD instance in such a way that for two stars from two "adjacent" tile sets the respective first or second entries of the tiles need to match; otherwise the number of vertices of some color in the central district will exceed Z.

5.1.1 Construction

Let $(\mathcal{S}, t, m, n, X, Y)$ be an instance of GRID TILING. We construct an instance of FCD as follows.

First of all, we set $\ell = 0$, $s_{\min} = 1$, $s_{\max} = \infty$, and $k = t^2 + 1$. For each $i, j \in [t]$, we introduce three distinct colors $b_{i,j}$, $d_{i,j}$, and $c_{i,j}$. Moreover, we introduce three distinct colors c, c', and $c^\star$. Now, we fix some constants which we use later. Let $W := 5n(t^2 + t) + 1$, $Z := 2(n - 1) \cdot 5mW$, $f(i,j) := i \cdot t + j$, and $g(i,j) := t^2 + t + i \cdot t + j$. Note that this implies that $W \geq 2.5 \cdot n \cdot g(i,j)$ and $W \geq 2.5 \cdot n \cdot f(i,j)$ for all $i, j \in [t]$.

We are now ready to construct the vertex-colored graph $G = (V, E)$. We start by introducing a *center vertex* v_{center} of color $c^\star$. In a solution, we call the district containing v_{center} the *center district*. We add Z vertices of color c and Z vertices of color c', all of which are only adjacent to v_{center}.

For each $i, j \in [t]$ and $(x, y) \in S^{i,j}$, we construct a star $T_{x,y}^{i,j}$ and connect its center to v_{center}. We color the center of $T_{x,y}^{i,j}$ in $c^\star$. Moreover, $T_{x,y}^{i,j}$ has the following leaves:

- $\frac{Z}{2(n-1)} + W \cdot x - f(i,j)$ vertices of color $d_{i,j}$,
- $\frac{Z}{2(n-1)} - W \cdot x - f(i',j)$ vertices of color $d_{i',j}$ for $i' := i + 1 \bmod k$,
- $\frac{Z}{2(n-1)} + W \cdot y - g(i,j)$ vertices of color $b_{i,j}$,
- $\frac{Z}{2(n-1)} - W \cdot y - g(i,j')$ vertices of color $b_{i,j'}$ for $j' := j + 1 \bmod k$, and
- $\max(\frac{Z}{2(n-1)} + W \cdot x - f(i,j), \frac{Z}{2(n-1)} + W \cdot y - g(i,j))$ vertices of color $c_{i,j}$.

Observe that the constructed star $T_{x,y}^{i,j}$ is 0-fair, as the number of occurrences of $c_{i,j}$ matches the number of occurrences of the otherwise most frequent color. This concludes the construction.

5.1.2 Proof of correctness

We start with a simple observation on the total number of vertices of some of the colors:

Observation 1 For each $i,j \in [t]$, it holds that $\chi_{d_{i,j}}(V) = \frac{n}{n-1}Z - 2nf(i,j)$ and $\chi_{b_{i,j}}(V) = \frac{n}{n-1}Z - 2ng(i,j)$. For all $i,j \in [k]$, it holds that $\chi_{c_{i,j}}(V) \leq Z$.

Proof For each $i,j \in [t]$, vertices of color $d_{i,j}$ occur in $2n$ different stars, that is, in all stars corresponding to tiles from $S^{i,j}$ and $S^{i-1,j}$. As the first entries of all tiles from $S^{i,j}$ and of all tiles from $S^{i-1,j}$ sum up to X, it follows that

$$\chi_{d_{i,j}}(V) = \sum_{(x,y) \in S^{i,j}} \frac{Z}{2(n-1)} + Wx - f(i,j) + \sum_{(x,y) \in S^{i-1,j}} \frac{Z}{2(n-1)} - Wx - f(i,j)$$

$$= \left(\frac{n}{2(n-1)}Z + WX - nf(i,j) \right) + \left(\frac{n}{2(n-1)}Z - WX - nf(i,j) \right)$$

$$= \frac{n}{n-1}Z - 2nf(i,j).$$

The same reasoning applies for all colors $b_{i,j}$ proving that $\chi_{b_{i,j}}(V) = \frac{n}{n-1}Z - 2ng(i,j)$. Lastly, for some $i,j \in [t]$, vertices of color $c_{i,j}$ appear only in stars corresponding to tiles from $S^{i,j}$, each of them containing at most $\frac{Z}{2(n-1)} + W \cdot m$ such vertices. Thus, the number of vertices of color $c_{i,j}$ is upper-bounded by $n \cdot (\frac{Z}{2(n-1)} + W \cdot m)$. As $W \cdot m = \frac{1}{8(n-1)}Z$ and $n > 2$, this is smaller than Z from which $\chi_{c_{i,j}}(V) \leq Z$ follows. $\square$

Using Observation 1, we now prove the forward direction of the correctness of the construction.

Lemma 1 *If the given* GRID TILING *instance is a yes-instance, then the constructed FCD instance is a yes-instance.*

Proof Let $S = \{(x^{i,j}, y^{i,j}) \in S^{i,j} \mid i,j \in [t]\}$ be a solution for the given GRID TILING instance. From this we construct a solution of the FCD instance as follows. For each $(x^{i,j}, y^{i,j}) \in S$, we create a separate district and put into this district all vertices from the star $T_{x,y}^{i,j}$ corresponding to $(x^{i,j}, y^{i,j})$. We put all other vertices in the center district. Note that this construction respects the given number $k = t^2 + 1$ of districts and that all districts are by construction non-empty and connected.

It remains to argue that all created districts are 0-fair. Since every star is 0-fair by construction, all non-center districts are 0-fair. For the center district, note that it contains Z vertices of color c and Z vertices of color c'. Thus, it is sufficient to argue that there is no color such that the number of its occurrences in the center district exceeds Z. For color $c_{i,j}$ this directly follows from Observation 1. Next we consider color $d_{i,j}$ for some $i,j \in [t]$. All stars with some vertices of color $d_{i,j}$ are part of the center district, except for the one corresponding to $(x^{i,j}, y^{i,j})$ and the one corresponding to $(x^{i-1,j}, y^{i-1,j})$. Since S is a solution, we have $x^{i-1,j} = x^{i,j}$. Consequently, exactly $\frac{Z}{n-1} - 2f(i,j)$ vertices of color $d_{i,j}$ are excluded from the center district. Now it follows from Observation 1 that the number of vertices of color $d_{i,j}$ in the center district is at most Z. An analogous argument also holds for $b_{i,j}$ for $i,j \in [t]$, which concludes the proof. $\qquad\square$

It remains to prove the correctness of the backward direction. To do this, we first observe that for every star, all of its vertices have to belong to the same district. Subsequently, we prove that the center district can contain at most Z vertices of each color. We then show that in order to respect this bound, for each tile set, exactly one star corresponding to a tile from this set must be excluded from the center district. Again exploiting the fact that every color has at most Z occurrences in the center district, we show that those excluded tiles form indeed a solution to the given GRID TILING instance. We start by observing that the vertices from one star need to be part of the same district.

Observation 2 For each tile $(x, y) \in S$, all vertices from the corresponding star need to be part of the same district in a solution to the constructed FCD instance.

Proof As we set $\ell = 0$ in the constructed FCD instance, there cannot exist a district containing just a single vertex. Thus, all vertices from a star need to belong to the same district as the center of the star. $\qquad\square$

Using this observation, we make some more involved arguments dealing with the possible number of vertices of a color in the center district in a solution. In the following, let V_{center} denote the center district in a solution to the constructed FCD instance. We make this observation in order to ensure that no two different colors appear the same number of times and in particular more than Z times in the center district.

Lemma 2 For each $i,j \in [t]$, if $\chi_{d_{i,j}}(V_{\text{center}}) \geq Z$, then the number α of stars $T_{x,y}^{i,j}$ with some vertex of color $d_{i,j}$ that are not part of the center district is at most two. In particular, it holds that $\chi_{d_{i,j}}(V_{\text{center}}) = (\frac{n}{n-1}Z - 2nf(i,j)) - \alpha(\frac{Z}{2(n-1)} - f(i,j)) + Wq$ for some $q \in [-2m, 2m]$.

Proof Let us consider some $i,j \in [t]$. By Observation 2, for every star $T_{x,y}^{i,j}$, all the vertices from $T_{x,y}^{i,j}$ are in V_{center} or no vertex from $T_{x,y}^{i,j}$ is in V_{center}.

By construction, every star with some vertex of color $d_{i,j}$ contains $\frac{Z}{2(n-1)} + W \cdot x - f(i,j)$ vertices of this color for some $x \in [-m, m]$. So as there are $\frac{n}{n-1}Z - 2nf(i,j)$ vertices of color $d_{i,j}$ (Observation 1), if $\alpha \geq 3$, then the number of vertices of color $d_{i,j}$ in the center district is at most $Z - \frac{Z}{2(n-1)} + 3mW$. By the definition of Z this is smaller than Z.

Thus, we obtain $\alpha \leq 2$. Since there are in total $\frac{n}{n-1}Z - 2nf(i,j)$ vertices of color $d_{i,j}$ by Observation 2, the lemma holds. $\qquad\square$

Analogously, we can prove similar bounds for the number of vertices of color $b_{i,j}$ for some $i, j \in [t]$ in the center district.

Lemma 3 *For each $i, j \in [t]$, if $\chi_{b_{i,j}}(V_{center}) \geq Z$, then the number α of stars $T_{x,y}^{i,j}$ with some vertex of color $b_{i,j}$ that are not part of the center district is at most two. In particular, it holds that $\chi_{b_{i,j}}(V_{center}) = (\frac{n}{n-1}Z - 2ng(i,j)) - \alpha(\frac{Z}{2(n-1)} - g(i,j)) + Wv$ for some $v \in [-2\,m, 2\,m]$.*

Using these two lemmas, we can now prove that there are at most Z vertices of the same color in the center district in a solution to the constructed FCD instance.

Lemma 4 *For every color q, $\chi_q(V_{center}) \leq Z$.*

Proof We claim that for two distinct colors q, q' with $\chi_q(V_{center}), \chi_{q'}(V_{center}) > Z$, it holds that $\chi_q(V_{center}) \neq \chi_{q'}(V_{center})$. The statement of the lemma directly follows from the claim, as V_{center} needs to be 0-fair.

Assume for a contradiction that $\chi_q(V_{center}) = \chi_{q'}(V_{center}) > Z$ for some colors $q \neq q'$. Since there are at most Z vertices of color $c, c', c^\star, c_{i,j}$ for $i, j \in [t]$, q and q' are among the colors $b_{i,j}$ and $d_{i,j}$ for $i, j \in [t]$. By Lemmas 2 and 3 and as it holds that $\chi_q(V_{center}) = \chi_{q'}(V_{center})$, from this it follows that $2nh(i,j) + a\left(\frac{Z}{2(n-1)} - h(i,j)\right) + Wv = 2nh'(i',j') + a'\left(\frac{Z}{2(n-1)} - h'(i',j')\right) + Wv'$, for some $a, a' \in \{0, 1, 2\}$, $v, v' \in [-2\,m, 2\,m]$, $h(i,j) \in \{f(i,j), g(i,j)\}$, and $h'(i',j') \in \{f(i',j'), g(i',j')\}$. Rewriting yields that $(5\,m(a - a') + v - v')W = (2n - a')h'(i',j') - (2n - a)h(i,j)$. Since W is sufficiently large, the absolute value of the right-hand side does not exceed W. Thus, we have $5\,m(a - a') + v - v' = 0$ and $(2n - a)h(i,j) = (2n - a')h'(i',j')$. The first equation implies $a = a'$ since $v - v' \in [-4\,m, 4\,m]$. Cancelling out $2n - a > 0$ in the second equation, we obtain $h(i,j) = h'(i',j')$. Now we have $q = q'$, as $f(\tilde{i}, \tilde{j}) \neq g(\tilde{i}, \tilde{j}')$ for all $\tilde{i}, \tilde{j}, \tilde{i}', \tilde{j}' \in [t]$, $f(\tilde{i}, \tilde{j}) = f(\tilde{i}', \tilde{j}')$ only if $\tilde{i} = \tilde{i}'$ and $\tilde{j} = \tilde{j}'$, and $g(\tilde{i}, \tilde{j}) = g(\tilde{i}', \tilde{j}')$ only if $\tilde{i} = \tilde{i}'$ and $\tilde{j} = \tilde{j}'$. We have reached a contradiction. □

Using Lemma 4, we can prove that each solution to the constructed FCD instance induces a selection of one tile from each tile set in the given GRID TILING instance:

Lemma 5 *For each $i, j \in [t]$, exactly one star $T_{x,y}^{i,j}$ for some $(x, y) \in S^{i,j}$ is not part of V_{center}.*

Proof Lemma 2 and Lemma 4 imply that for each $i, j \in [t]$ at least two stars containing vertices of color $d_{i,j}$ are not part of V_{center}. As each star $T_{x,y}^{i,j}$ contains vertices of colors $d_{i,j}$ and $d_{i+1,j}$ and as there exist t^2 non-center districts in the end, this means that for each $i, j \in [t]$ exactly two stars containing vertices of this color are not part of V_{center}.

For the sake of contradiction, let us assume that there exists $i, j \in [t]$ such that two stars corresponding to tiles $(x, y), (x', y') \in S^{i,j}$ are not part of V_{center}. This implies by our previous observation that all other stars containing vertices of color $d_{i+1,j}$ need to belong to V_{center}. However, in this case, $\frac{Z}{n-1} - W(x + x') - 2f(i,j) \leq \frac{W}{n-1} - 2W$ vertices of color $d_{i+1,j}$ are not part of V_{center}. By Observation 1, this implies that the number of vertices of color $d_{i+1,j}$ in V_{center} is at least $Z + 2W - 2nf(i,j)$. As $W > n(t^2 + t)$, this number is greater than Z, contradicting Lemma 4. □

Putting all pieces together, we are now ready to prove the correctness of the backward direction of the construction.

Lemma 6 *If the constructed FCD instance is a yes-instance, then the given GRID TILING instance is a yes-instance.*

Proof Let $\mathcal{V}$ be a solution to the constructed FCD instance and let $V_{center} \in \mathcal{V}$ be the district containing v_{center}. Lemma 5 implies that for each $i, j \in [t]$ exactly one star corresponding to a tile from $S^{i,j}$ is not part of V_{center}. Let

$$S := \{(x^{i,j}, y^{i,j}) \in S^{i,j} \mid i, j \in [t] \wedge T^{i,j}_{x,y} \text{ is not part of } V_{center}\}$$

be a set of all tiles corresponding to excluded stars (one for each $i, j \in [t]$). We claim that S is a valid solution to the given GRID TILING instance.

For the sake of contradiction, assume that this is not the case. That is, there either (a) exist $i, j \in [t]$ such that $x^{i,j} \neq x^{i-1,j}$ or (b) exist $i', j' \in [t]$ such that $y^{i',j'} \neq y^{i',j'-1}$. Let us start by assuming that (a) is the case. Note that it is possible to assume without loss of generality that $x^{i,j} < x^{i-1,j}$, as from the fact that there exists some $x^{i,j} \neq x^{i-1,j}$ it follows that there also exists some $\tilde{i} \in [t]$ such that $x^{\tilde{i},j} < x^{\tilde{i}-1,j}$. The only stars containing vertices of color $d_{i,j}$ that are not part of V_{center} are the star corresponding to $(x^{i,j}, y^{i,j})$, which contains $\frac{Z}{2(n-1)} + W \cdot x^{i,j} - f(i,j)$ vertices of this color, and the star corresponding to $(x^{i-1,j}, y^{i-1,j})$, which contains $\frac{Z}{2(n-1)} - W \cdot x^{i-1,j} - f(i,j)$ vertices of this color. This implies that at most $\frac{Z}{n-1} + W \cdot (x^{i,j} - x^{i-1,j}) - 2f(i,j) \leq \frac{Z}{n-1} - W - 2f(i,j)$ vertices of this color are not part of V_{center}, where the inequality holds by our assumption that $x^{i,j} < x^{i-1,j}$. Combining this with Observation 1, it follows that the number of vertices of color $d_{i,j}$ in V_{center} is at least $Z + W - 2(n-1)f(i,j)$. By the definition of W, this is strictly greater than Z. Applying Lemma 4, we reach a contradiction.

The same argument can also be applied if (b) holds, which proves that S is a solution to the given GRID TILING instance. $\square$

From Lemma 1 and Lemma 6 the correctness of the reduction follows. As our construction takes only polynomial time and $|C| + k = 3t^2 + 4$ is bounded in a function of t, the NP-hardness and the W[1]-hardness with respect to $|C| + k$ of FCD on trees follows.

Theorem 2 *FCD on trees is NP-hard and W[1]-hard with respect to $|C| + k$, even if $s_{min} = 1$ and $s_{max} = \infty$.*

Recall that by Corollary 3, FCD can be solved in polynomial time on all graphs with pathwidth one (disjoint unions of caterpillars). Observe that the tree constructed in the reduction above has pathwidth two: Graph G' obtained from G by deleting v_{center} is a disjoint union of stars. Thus, G' admits a path decomposition of width one. Placing v_{center} into every bag yields a path decomposition of G of width two. This results in the following.

Corollary 4 *FCD on graphs G with $pw(G) = 2$ is NP-hard and W[1]-hard with respect to $|C| + k$, even if $s_{min} = 1$ and $s_{max} = \infty$.*

To be more precise, we even get NP-hardness and W[1]-hardness with respect to $|C| + k$ on a pathwidth one graph to which we add a single vertex and adjacent edges. Notably, this

result is tight in the sense that we have proved polynomial-time solvability on pathwidth-one graphs in Corollary 3.

As the tree constructed in the previous reduction has diameter four (it consists of a center vertex and centers of stars attached to it), we conclude that FCD is computationally intractable even on trees with a small constant diameter:

Corollary 5 *FCD is NP-hard and W*[1]-*hard with respect to* $|C| + k$ *on trees with diameter four, even if* $s_{\min} = 1$ *and* $s_{\max} = \infty$.

We will complement this result in Corollary 8 where we show that FCD is polynomial-time solvable on trees with diameter at most three.

5.2 An XP-algorithm for $\mathbf{tw} + |C|$

Motivated by the hardness result from the previous subsection, we search for an XP-algorithm for the parameters $|C|$ and k for FCD on trees. We start by considering the parameter $|C|$ here and the parameter k in the next two subsections. Specifically, we show that there exists an XP-algorithm for $|C|$ on tree-like graphs—more precisely, we present an XP-algorithm with respect to $|C| + \mathrm{tw}$, where tw is the treewidth of the underlying graph.

Given a graph $G = (V, E)$, a tree decomposition $(T = (V_T, E_T), \{B_x\}_{x \in V_T})$ of G (as defined in the Preliminaries) is *nice* if each node $x \in V_T$ has one of the following types[5]:

Leaf node.	A leaf of T with $B_x = \{v\}$ for $v \in V$.
Introduce vertex v node.	An internal node of T with one child $y \in V_T$ such that $B_x = B_y \cup \{v\}$.
Introduce edge $\{u, v\}$ node.	An internal node of T with one child $y \in V_T$ such that $u, v \in B_x = B_y$.
Forget v node.	An internal node of T with one child $y \in V_T$ such that $B_x = B_y \backslash \{v\}$ for $v \in B_y$.
Join node.	An internal node of T with two children $y \in V_T$ and $z \in V_T$ such that $B_x = B_y = B_z$.

We will implicitly assume that every introduce edge $\{u, v\}$ node is labeled by $\{u, v\}$ and that for each edge there is exactly one such node. Given a tree decomposition, a nice tree decomposition of equal width can be computed in linear time [32]. By applying dynamic programming on top of the nice tree decomposition of the given graph, we establish the following:

Theorem 3 (★) *FCD can be solved in* $\mathcal{O}(n^{\mathcal{O}(\mathrm{tw} \cdot |C|)})$ *time.*

Since a tree is of treewidth one, we have the following:

Corollary 6 *FCD on trees can be solved in* $n^{\mathcal{O}(|C|)}$ *time.*

---

[5] Note that in the following we refer to the elements of V as vertices and the elements of V_T as nodes. We refer to the set B_x for a node $x \in V_T$ as a bag.

5.3 An XP-algorithm for fen + k

Having constructed a polynomial-time algorithm for constant $|C|$ on trees and on graphs with a constant treewidth, we now consider the number k of districts as our parameter for FCD on trees. We show that there is a simple XP-algorithm with respect to k for FCD on trees, which naturally extends to an XP-algorithm with respect to fen $+ k$, where fen is the number of edges that need to be deleted to make the given graph a tree:

Proposition 6 *FCD can be solved in $n^{\mathcal{O}(\text{fen}+k)}$ time.*

Proof Suppose that there is a solution $(V_1, \dots, V_k)$. As for each $i \in [k]$, V_i is connected (for which at least $|V_i| - 1$ edges are needed), the number of edges whose endpoints both lie in the same district is at least $\sum_{i=1}^{k}(|V_i| - 1) = n - k$. Since the input graph $G = (V, E)$ has at most $n + \text{fen} - 1$ edges (by the definition of fen), it follows that there are at most $(n + \text{fen} - 1) - (n - k) = \text{fen} + k - 1$ edges whose endpoints belong to different districts. Accordingly, in our algorithm for each edge set $E' \subseteq E$ of size at most fen $+ k - 1$, we verify whether $(V, E \setminus E')$ has k connected components each of which being ℓ-fair and respecting the size constraints. We return yes if this is the case for some subset of edges and no otherwise. Overall, it takes $(n + \text{fen} - 1)^{\text{fen}+k-1} \cdot n^{\mathcal{O}(1)} = n^{\mathcal{O}(\text{fen}+k)}$ time to do so.   □

As the treewidth of a graph is upper-bounded in a function of its feedback edge number, we can conclude from Theorem 3 that FCD parameterized by fen $+ |C|$ is in XP. However, Proposition 6 leaves it open whether there is an XP-algorithm for tw $+ k$. We answer this question negatively in the next subsection.

5.4 NP-hardness for f vn = 1 and k = 2

Having already considered the parameters treewidth and feedback edge number, we now consider a third way to measure the distance from a tree, the number of vertices to delete to make it a tree (feedback vertex number fvn). As tw $+ 1 \le$ fvn, from Theorem 3 it follows that there is an XP-algorithm for fvn $+ |C|$. We now prove that FCD is NP-hard even for fvn $= 1$ and $k = 2$, in contrast to the result from the previous subsection, where we gave an $n^{\mathcal{O}(\text{fen}+k)}$-time algorithm. Notably, this NP-hardness result also excludes the existence of an XP-algorithm for tw $+ k$ unless P = NP:

Theorem 4 *FCD is NP-hard for* fvn $= 1$ *and* $k = 2$, *even if* $s_{\min} = 1$ *and* $s_{\max} = \infty$.

The rest of the section is devoted to proving the theorem. We reduce from the NP-hard NOT-ALL-EQUAL 3-SAT problem [46] where the input is a set X of n boolean variables and a set Y of m clauses over X such that each clause $y \in Y$ contains three different literals, and the question is whether there exists a truth assignment to the variables in X such that for each clause $y \in Y$ at least one literal is set to true and at least one literal is set to false. Notably, given an assignment fulfilling these constraints, the assignment that assigns all variables in X the opposite truth value also fulfills the constraints.

The general idea of the construction is that we introduce one vertex for each literal and that the two districts in the solution correspond to two opposite truth assignments of the variables from X that are both solutions of the NOT-ALL-EQUAL 3-SAT instance.
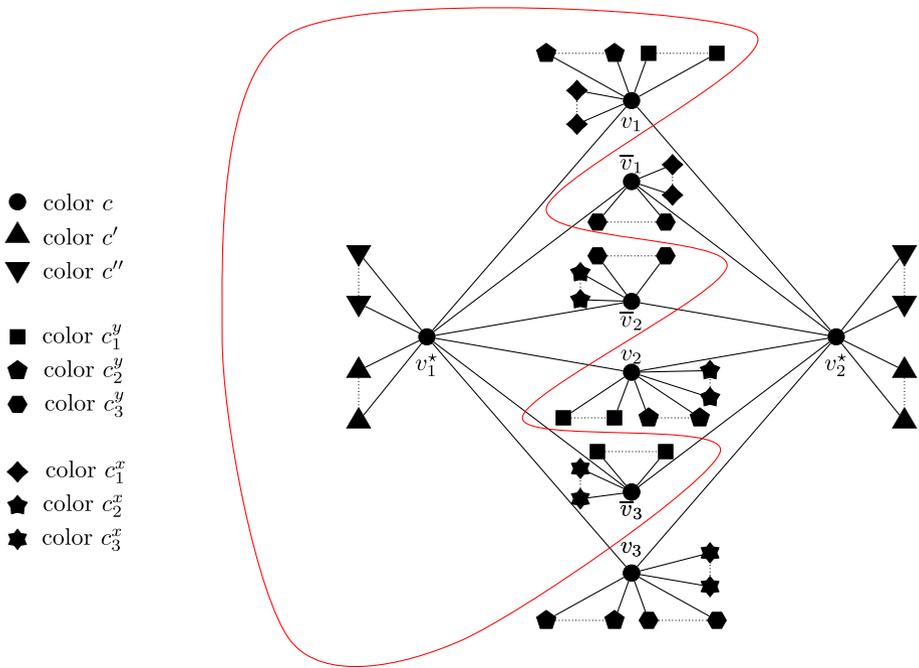
Fig. 3 Example of the hardness reduction from Theorem 4 for the Not-All-Equal 3-Sat instance $X = \{x_1, x_2, x_3\}$, $Y = \{y_1 = \{x_1, x_2, \overline{x}_3\}, y_2 = \{x_1, x_2, x_3\}, y_3 = \{\overline{x}_1, \overline{x}_2, x_3\}\}$. The district marked in red corresponds to the solution setting x_1 to true and x_2 and x_3 to false

5.4.1 Construction

Given an instance $(X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_m\})$ of Not-All-Equal 3-Sat, we construct an instance of FCD as follows. We add a color c_i^x for each variable $x_i \in X$ and a color c_j^y for each clause $y_j \in Y$. In addition, we add three colors c, c', and c''. Moreover, we set $k = 2$, $\ell = 0$, $s_{\min} = 1$, and $s_{\max} = \infty$. Let $Z := 2 \cdot n \cdot m + 1$.

We start the construction of the vertex colored graph $G = (V, E)$ by introducing two *central vertices* $v_1^\star$ and $v_2^\star$ of color c. We will construct the instance in a way that these two vertices need to lie in different districts. For each central vertex $v_i^\star$, $i \in \{1, 2\}$, we introduce $3Z$ vertices of color c' and $3Z$ vertices of color c'' and connect them to $v_i^\star$.

Subsequently, for each variable $x_i \in X$, we introduce two *literal vertices* v_i and $\overline{v}_i$ of color c and connect both these vertices to the two central vertices. For each literal vertex $\tilde{v}_i \in \{v_i, \overline{v}_i\}$, we introduce $3Z - i$ vertices of color c_i^x and connect them to $\tilde{v}_i$ (these vertices make sure that the two literal vertices end up in different districts). For each clause $y_j \in Y$ in which x_i occurs positively, we introduce $Z + j$ vertices of color c_j^y and connect them to v_i. For each clause $y_j \in Y$ in which x_i occurs negatively, we introduce $Z + j$ vertices of color c_j^y and connect them to $\overline{v}_i$ (these vertices ensure that there is no clause in which all three literal vertices corresponding to literals from the clause lie in the same district). See Fig. 3 for a visualization of the construction.

We start by showing the forward direction of the correctness of the construction.

Lemma 7 *If the given* Not-All-Equal *3-*Sat *instance is a yes-instance, then the constructed FCD instance is a yes-instance.*

Proof Let $X' \subseteq X$ be the set of variables set to true in a solution to the given Not-All-Equal 3-Sat instance. From this, we construct a solution (V_1, V_2) to the constructed FCD instance as follows. We include $v_1^\star$ and all leaves attached to it in V_1. Moreover, we include in V_1 the following vertices: v_i and all leaves attached to it for all $x_i \in X'$ and $\bar{v}_i$ and all leaves attached to it for all $x_i \in X \setminus X'$. We include all other vertices in V_2.

It is easy to verify that V_1 and V_2 are both connected. We will show that V_1 is 0-fair. By symmetry, an analogous argument will show that V_2 is 0-fair. First, observe that V_1 contains exactly $3Z$ vertices of color c' and $3Z$ vertices of color c''. We show that $\chi_{\tilde{c}}(V_1) \leq 3Z$ for every color $\tilde{c} \in C$. For color c, we have that $\chi_c(V_1) \leq \chi_c(V) = n + 2 < 3Z$. For each variable $x_i \in X$, as the two corresponding literal vertices are part of different districts, we have that $\chi_{c_i^x}(V_1) = 3Z - i < 3Z$. For each clause $y_j \in Y$, as X' is a solution, either one or two literal vertices corresponding to literals in y_j and the attached leaves are part of V_1. Thus, the number of vertices of color c_j^y in V_1 is either $Z + j$ or $2Z + 2j$. As $Z > 2m$, it follows that $\chi_{c_j^y}(V_1) < 3Z$. Thus, both districts V_1 and V_2 are 0-fair.   □

It remains to prove the correctness of the backward direction of the reduction. For this, note that the FCD instance is constructed such that the two central vertices need to end up in different districts and for each $x_i \in X$, the two literal vertices v_i and $\bar{v}_i$ need to end up in different districts. Thus, the two districts correspond to two inverse truth assignments. Subsequently, we will prove that there is no clause in which all corresponding vertices are in the same district. This will show that the two truth assignments induced by the two districts are a solution to the given Not-All-Equal 3-Sat instance.

We start by observing that all leaves need to lie in the same district as the vertex they are attached to.

Observation 3 *In a solution to the constructed FCD instance, all leaves attached to a vertex $v \in V$ need to lie in the same district as v.*

Proof If a leaf is not in the same district as its neighbor, then the leaf needs to form its own district. However, this is not possible, as we require each district to be 0-fair.   □

Next, we show that indeed the two central vertices $v_1^\star$ and $v_2^\star$ always lie in different districts.

Observation 4 *The two central vertices $v_1^\star$ and $v_2^\star$ cannot belong to the same district in any solution to the constructed FCD instance.*

Proof Assume that $v_1^\star$ and $v_2^\star$ belong to the same district. By Observation 3, the other district V' needs to consist of a single literal vertex v_i or $\bar{v}_i$ for some $x_i \in X$ and all leaves attached to it. The most frequent color appearing in V' is c_i^x which occurs $3Z - i > 2Z$ times (by the definition of Z). The second most frequent color is either c_j^y for some $j \in [m]$ occurring $Z + j < 2Z$ times or c occurring once (by the definition of Z). Thus, the district cannot be 0-fair.   □

To prove that, for each variable, the two corresponding literal vertices need to be part of different districts and that not all literal vertices corresponding to literals from a clause can lie in the same district, we prove that neither district can contain more than $3Z$ vertices of the same color.

Lemma 8 *For any solution to the constructed FCD instance, no district contains more than $3Z$ vertices of the same color.*

Proof Assume that there exists a solution with a district V' containing more than $3Z$ vertices of the same color. Since V' is 0-fair, there are two colors q and q' with $\chi_q(V') = \chi_{q'}(V') > 3Z$. We have $\chi_c(V') \leq \chi(V) = n + 2 < 3Z$ and $\chi_{c'}(V') = \chi_{c''}(V') = 3Z$ by Observation 4. Thus, we have $q, q' \in \{c_i^x \mid x_i \in X\} \cup \{c_j^y \mid y_j \in Y\}$.

For $x_i \in X$, there are two literal vertices each of which have $3Z - i$ leaves of color c_i^x attached to it. For $y_j \in Y$, there are three literal vertices each of which have $Z + j$ leaves of color c_j^y attached to it. It follows from Observation 3 that $\chi_{c_i^x}(V') \in \{0, 3Z - i, 6Z - 2i\}$ for each $x_i \in X$ and that $\chi_{c_j^y}(V') \in \{0, Z + j, 2Z + 2j, 3Z + 3j\}$ for each $y_j \in Y$. Since $i \leq n$, $j \leq m$, $Z = 2 \cdot n \cdot m + 1$, and $\chi_q(V') = \chi_{q'}(V')$ for $q, q' \in \{c_i^x \mid x_i \in X\} \cup \{c_j^y \mid y_j \in Y\}$, it needs to hold that $\chi_q(V') = \chi_{q'}(V') = 0$, which contradicts $\chi_q(V') = \chi_{q'}(V') > 3Z$. $\square$

Recalling that $3Z - i$ vertices of color c_i^x are attached to each literal vertex corresponding to $x_i \in X$ and that $6Z - 2i > 3Z$, the next observation directly follows from the previous lemma and Observation 3.

Observation 5 Let (V_1, V_2) be a solution of the constructed FCD instance. Then, for each $x_i \in X$, exactly one of the two corresponding literal vertices v_i and $\bar{v}_i$ is part of V_1 and the other is part of V_2.

We are now ready to prove the backward direction of the correctness of our construction.

Lemma 9 *If the constructed FCD instance is a yes-instance, then the given NOT-ALL-EQUAL 3-SAT instance is a yes-instance.*

Proof Let (V_1, V_2) be a solution to the constructed FCD instance. From Observation 5, it follows that for each $x_i \in X$, exactly one of v_i and $\bar{v}_i$ is part of V_1. Let φ be the truth assignment induced by V_1, i.e., φ sets x_i to true if $v_i \in V_1$ and x_i to false if $\bar{v}_i \in V_1$. We claim that φ is a solution to the given NOT-ALL-EQUAL 3-SAT instance. Firstly, for the sake of contradiction, assume that there exists a clause $y_j \in C$ containing only literals that are satisfied by φ. However, by Observation 3, this implies that all vertices of color c_j^y are part of V_1, contradicting Lemma 8 as there exist $3Z + 3j$ such vertices. Secondly, for the sake of contradiction, assume that y_j contains no literal satisfied by φ. However, by Observation 3, this implies that all vertices of color c_j^y are part of V_2, contradicting again Lemma 8. Consequently, φ is a solution. $\square$

Observing that $\{v_1^\star\}$ is a feedback vertex set of the constructed graph and that the construction can be computed in polynomial time, Theorem 4 follows directly from Lemma 7 and Lemma 9

From our reduction, we can further conclude that FCD is also para-NP-hard with respect to the treewidth plus the number k of districts.

Corollary 7 *FCD is NP-hard for* tw $= 2$ *and $k = 2$, even if $s_{\min} = 1$ and $s_{\max} = \infty$.*

6 FCD on graphs of bounded vertex cover number

Motivated by our hardness results for graphs with constant treewidth, we now turn to the size vcn of a minimum vertex cover, a parameter never smaller than the treewidth. In this section, we present two parameterized algorithms, namely, an XP-algorithm for vcn and an FPT-algorithm for vcn $+ |C|$. Unfortunately, we were unable to settle whether FCD parameterized by vcn is W[1]-hard or fixed-parameter tractable. In contrast, we develop an FPT-algorithm for the number of vertices with degree at least two (a parameter which is never smaller than vcn).

Both algorithms for vcn rely on the following lemma.

Lemma 10 *Let S be a vertex cover of minimum size and $\mathcal{V} = (V_1, \ldots, V_k)$ a solution to an FCD instance on a graph G. There are at most vcn districts in $\mathcal{V}$ that contain at least one vertex from S. Moreover, for every $V_i \in \mathcal{V}$ with $S \cap V_i \neq \emptyset$, there is a set $J_i \subseteq V_i \backslash S$ of at most $|S \cap V_i| - 1$ vertices such that $G[(S \cap V_i) \cup J_i]$ is connected.*

Proof As each vertex is only contained in one district, the first part of the lemma follows from the definition of vcn. To prove the second part, fix some $V_i \in \mathcal{V}$. Consider a minimum spanning tree $T = (V_i, F)$ of $G[V_i]$. Let $J_i \subseteq V_i \setminus S$ be the set of vertices of degree at least two in T. Let $T' := ((S \cap V_i) \cup J_i, F')$ be the result of deleting from T each vertex $v \in V_i \setminus (S \cup J_i)$ along with an edge incident to it. Observe that T' is connected and thus $G[(S \cap V_i) \cup J_i]$ is connected, which contains T' as a subgraph. We show that $|J_i| \leq |S \cap V_i| - 1$. As by the definition of J_i vertices from J_i are only adjacent to vertices from $S \cap V_i$ and as every vertex from J_i has degree at least two in T and T', we have $|F'| \geq 2|J_i|$. We also have $|F'| = |S \cap V_i| + |J_i| - 1$ since T' is a tree. Thus, we have $|J_i| \leq |S \cap V_i| - 1$. □

We first show that FCD parameterized by vcn is in XP using the following approach: We first guess how the vertex cover is partitioned into districts in the sought solution, which gives partial (not necessarily connected) districts. For every partial district, we then guess some vertices outside the vertex cover to include such that the partial district becomes connected, the two most frequent colors, and how often they occur in the resulting district. The remaining problem can be reduced to the polynomial-time solvable (g, f)-Factor problem [25].

Theorem 5 *FCD is solvable in $n^{\mathcal{O}(\mathrm{vcn})}$ time.*

Proof Suppose there is a solution $\mathcal{V} = (V_1, \ldots, V_k)$ to the given FCD instance. Let S be a vertex cover of minimum size and let $I = V \setminus S$ be the vertices outside the vertex cover. As in the proof of Theorem 6, our algorithm first tries all possibilities to fix some structure with respect to S. Then, we will construct an instance of the polynomial-time solvable (g, f)-Factor problem, which generalizes Maximum Matching, for each choice of the following:

- An integer $k' \leq \min(k, \text{vcn})$. If $\ell = 0$ or $s_{\min} \geq 2$, then we consider only one choice for k', namely $k' = k$.
- A partition of S into k' non-empty subsets $S_1, \ldots, S_{k'}$. There are at most vcn^{vcn} such partitions.
- For every $i \in [k']$, a set $J_i \subseteq I$ of at most $|S_i| - 1$ vertices such that $G[S_i \cup J_i]$ is connected and $J_i \cap J_{i'} = \emptyset$ for $i \neq i' \in [k']$. We can assume that $|J_i| \leq |S_i| - 1$ vertices are sufficient to make $G[S_i]$ connected by Lemma 10. The number of choices for J_i is at most $\prod_{i \in [k']} n^{|S_i| - 1} \leq n^{\text{vcn}}$.
- For every $i \in [k']$, a pair (c_i, c_i') of colors, which have the largest numbers of occurrences in the sought V_i among all colors. There are at most $|C|^{2\,\text{vcn}}$ choices for all pairs of colors.
- For every $i \in [k']$, the numbers $z_{c_i}, z_{c_i'}$ of occurrences of color c_i and c_i', respectively, with $z_{c_i} - \ell \leq z_{c_i'} \leq z_{c_i}$. Since $z_{c_i} \leq n$ and $z_{c_i'} \leq n$, there are at most $n^{2\,\text{vcn}}$ choices.

Since $|C| \leq n$, there are at most $n^{\mathcal{O}(\text{vcn})}$ choices. Let $I' = I \setminus \bigcup_{i \in [k']} J_i$. Now the question is whether we can partition the vertices V into k districts $(V_1, \ldots, V_k)$ such that, for $i \in [k', k]$, V_i consists of a single vertex from I', and for $i \in [k']$, $S_i \cup J_i \subseteq V_i$, $G[V_i]$ is connected, $|V_i| \in [s_{\min}, s_{\max}]$, $\chi_{c_i}(V_i) = z_{c_i}$, $\chi_{c_i'}(V_i) = z_{c_i'}$ and $\chi_c(V_i) \leq \chi_{c_i'}(V_i)$ for all $c \in C \setminus \{c_i\}$. We reject the current combination, if for some $i \in [k']$ $\chi_{c_i}(S_i \cup J_i) > z_{c_i}$, $\chi_{c_i'}(S_i \cup J_i) > z_{c_i'}$, $\chi_c(S_i \cup J_i) > z_{c_i'}$ for some $c \in C \setminus \{c_i\}$, or $|S_i \cup J_i| > s_{\max}$. Otherwise, to decide whether it is possible to distribute the vertices I' to construct a partition respecting the above-described properties, we reduce to (g, f)-FACTOR, where given a graph $H = (U, F)$ and two functions $g, f : V \mapsto \mathbb{N}$, the question is whether there is a subgraph $H' = (U, F')$ with $F' \subseteq F$ such that every vertex has at least $g(v)$ and at most $f(v)$ neighbors in H'.

We construct a bipartite graph H and f, g as follows. The left bipartition of H consists of all vertices $v \in I'$. For every $i \in [k']$, we add the following vertices to the right bipartition:

- For each color $c \in \{c_i, c_i'\}$, we add $z_c - \chi_c(S_i \cup J_i)$ vertices (call them A_i) and connect them via edges to all vertices $v \in I'$ having color c in G which have at least one neighbor in S_i in G.
- For each color $c \in C \setminus \{c_i, c_i'\}$, we add $z_{c_i'} - \chi_c(S_i \cup J_i)$ vertices (call them B_i^c) and connect them via edges to all vertices $v \in I'$ having color c in G which have at least one neighbor in S_i in G. Let $B_i := \bigcup_{c \in C \setminus \{c_i, c_i'\}} B_i^c$.

If $s_{\min} = 1$, then we add a set A' of $k - k'$ vertices to the right bipartition and connect them via edges to all vertices of I'. For every vertex v introduce thus far, let $f(v) = g(v) = 1$. Finally, for every $i \in [k']$, add a vertex v_i to the left side and let $f(v_i) = |B_i| - \max(0, s_{\min} - |A_i \cup S_i \cup J_i|)$ and $g(v_i) = |B_i| - s_{\max} + |A_i \cup S_i \cup J_i|$ (if it does not hold that $0 \leq g(v_i) \leq f(v_i)$, then we continue with the next combination). We connect v_i to all vertices from B_i^c.

If the constructed (g, f)-FACTOR instance is a yes-instance, we return yes; otherwise, we continue with the next combination. To see why the algorithm works correctly, suppose that there is a subgraph $H' = (U, F')$ of H such that the degree of every vertex v in H' is in the range $[g(v), f(v)]$. From H', we can construct a solution of the given FCD instance by putting each vertex $v \in I'$ into the district corresponding to the neighbor of v in H' (if $s_{\min} = 1$, then every vertex adjacent to some vertex from A' forms a district of its own). Moreover, for all $i \in [k']$, we add $S_i \cup J_i$ to V_i. As all vertices from I' have one neighbor in H', each vertex from I' is assigned to a district. As all vertices from A have one neighbor in H', for $i \in [k']$, it holds that $\chi_{c_i} = z_{c_i}$ and $\chi_{c_i'} = z_{c_i'}$ and thus that the resulting districts are ℓ

-fair. Moreover, the number of neighbors of v_i in H' is between $|B_i| - s_{\max} + |A_i \cup S_i \cup J_i|$ and $|B_i| - \max(0, s_{\min} - |A_i \cup S_i \cup J_i|)$, implying that there are at least $s_{\min} - |A_i \cup S_i \cup J_i|$ and at most $s_{\max} - |A_i \cup S_i \cup J_i|$ vertices in B_i that adjacent to some vertex from I'. Thus, the size of each district is in $[s_{\min}, s_{\max}]$.　　　　　　　　　$\square$

Recall that FCD is NP-hard and W[1]-hard with respect to $|C| + k$ on trees with diameter four (Corollary 5). In contrast to this, since a tree of diameter at most three has a vertex cover of size at most two, by Theorem 5, the following holds.

Corollary 8 *FCD is polynomial-time solvable on trees with diameter at most three.*

Recall that we have shown in Corollary 5 that FCD is NP-hard on trees with diameter four.

Using a similar approach as for Theorem 5, we show that FCD is fixed-parameter tractable with respect to vcn $+ |C|$. The overall idea is the following: We guess the partition of the vertex cover into the districts. We categorize vertices outside the vertex cover according to their neighborhood and color. We formulate the resulting problem as an ILP whose number of variables only depends on vcn $+ |C|$. Subsequently we employ Lenstra's FPT-algorithm for ILPs [30, 36] to obtain the following:

Theorem 6 *FCD parameterized by* vcn $+ |C|$ *is fixed-parameter tractable.*

Proof Suppose there is a solution $\mathcal{V} = (V_1, \ldots, V_k)$. Let S be a vertex cover of minimum size. Then the vertices $I = V \backslash S$ that are not part of the vertex cover form an independent set. Let k' be the number of districts in $\mathcal{V}$ that contain at least one vertex of S. By Lemma 10, we have $k' \leq$ vcn. Assume without loss of generality that exactly for every $i \in [k']$, $S_i \neq \emptyset$ where $S_i = S \cap V_i$. Then, for every $i \in \{k' + 1, \ldots, k\}$, we have $V_i = \{v\}$ where v is some vertex from I.

We say that a vertex $v \in I$ has type (c, X) for $c \in C$ and $X \subseteq S$ if the color of v in G is c and its neighborhood is X. Let $\mathcal{T}$ denote the set of all types (note that $|\mathcal{T}| = 2^{\text{vcn}} \cdot |C|$). For type $T \in \mathcal{T}$, let n_T be the number of vertices of type T, and for type T with $n_T > 0$ let $v_T \in I$ be an arbitrary vertex of type T. Our algorithm will construct an ILP instance for each choice of the following:

- An integer $k' \leq$ vcn. If $\ell = 0$ or $s_{\min} > 1$, then we consider only one choice for k', namely $k' = k$.
- A partition of S into non-empty subsets $S_1, \ldots, S_{k'}$. There are at most vcn $^{\text{vcn}}$ such partitions.
- For every $i \in [k']$, a set $\mathcal{T}_i$ of at most $|S_i| - 1$ types such that $G[S_i \cup J_i]$ is connected for $J_i = \{v_T \mid T \in \mathcal{T}_i\}$. If $|\{i \mid T \in \mathcal{T}_i\}| > n_T$ for some $T \in \mathcal{T}$, then we reject the current combination. Note that we can assume that $|\mathcal{T}_i| \leq |S_i| - 1$ vertices are sufficient to make $G[S_i]$ connected by Lemma 10. As there are at most $2^{\text{vcn}} \cdot |C|$ types, the number of choices for all $\mathcal{T}_i$ is at most $\prod_{i \in [k']} (2^{\text{vcn}} \cdot |C|)^{|S_i| - 1} \leq 2^{\text{vcn}^2} \cdot |C|^{\text{vcn}}$.
- For every $i \in [k']$, a pair (c_i, c_i') of colors, which have the largest numbers of occurrences in V_i among all colors. In the sought solution it holds that $\chi_{c_i}(V_i) \geq \chi_{c_i'}(V_i)$. There are at most $|C|^{2\,\text{vcn}}$ choices for all pairs of colors.

To decide on the distribution of the vertices from I, we now construct an ILP. For the ILP formulation, we introduce an integer variable $x_{i,T}$ for each $i \in [k']$ and $T \in \mathcal{T}$. The variable $x_{i,T}$ will indicate the number of vertices of type T that we put in V_i. Clearly, there are at most n_T vertices for each type $T \in \mathcal{T}$ that belong to one of $V_1, \ldots V_{k'}$:

$$\sum_{i \in [k']} x_{i,T} \leq n_T.$$

Then for every $i \in [k']$ and $T \in \mathcal{T}_i$, at least one vertex of type T is included in V_i to satisfy our previous guesses:

$$x_{i,T} \geq 1.$$

Further for every $i \in [k']$, only vertices from I that are adjacent to a vertex from S_i can be part of V_i. Thus, for all types $T = (c, X) \in \mathcal{T}$ with $X \cap S_i = \emptyset$ it holds that:

$$x_{i,T} = 0.$$

Moreover, there are exactly $k - k'$ vertices which are contained in none of $V_1, \ldots, V_{k'}$ as they form their own districts:

$$\sum_{T \in \mathcal{T}} \left(n_T - \sum_{i \in [k']} x_{i,T} \right) = k - k'.$$

For $i \in [k']$ and $c \in C$, let $n_{i,c}$ be the number of vertices of color c in V_i:

$$n_{i,c} = \chi_c(S_i) + \sum_{T \in \mathcal{T} \text{ with } T=(c,X) \text{ for some } X \subseteq S} x_{i,T}.$$

For every $i \in [k']$, the following two constraints will ensure that the district V_i is ℓ-fair.

$$n_{i,c_i'} \leq n_{i,c_i} \leq n_{i,c_i'} + \ell, \text{ and } n_{i,c_i'} \geq n_{i,c} \text{ for all } c \in C \setminus \{c_i\}.$$

Finally, for every $i \in [k']$, the following imposes the size constraints:

$$s_{\min} \leq \sum_{c \in C} n_{i,c} \leq s_{\max}.$$

For the running time, observe that we construct at most $2^{\mathcal{O}(\text{vcn}^2)} \cdot |C|^{\text{vcn}}$ ILP instances. Since each ILP instance uses at most $\text{vcn} \cdot 2^{\text{vcn}} \cdot |C|$ variables, it can be solved in $f(\text{vcn} + |C|) \cdot n^{\mathcal{O}(1)}$ time for some computable function f due to a result of Lenstra [30, 36]. □

It remains open whether FCD is fixed-parameter tractable or W[1]-hard with respect to vcn. However, we can prove fixed-parameter tractability with respect to the number of vertices with degree at least two. Neglecting connected components consisting of two vertices, the set of vertices with degree at least two is always also a vertex cover; therefore, this parameter upper-bounds the vertex cover number in connected graphs on at least three vertices. The idea of the algorithm is to guess the partitioning of the degree-two vertices into the districts. Then, for each of these districts, there exists a set of degree-one vertices where each of these vertices either belongs to the district or needs to form its own. Lastly, we can distribute the degree-one vertices using Proposition 4.

Proposition 7 *Let p be the number of vertices with degree at least two. FCD is solvable in $\mathcal{O}(p^p \cdot k \cdot n)$ time.*

For the sake of simplicity, we assume that the graph does not contain connected components of size two (we can deal with them easily separately). Let $X \subseteq V$ be the set of vertices with degree at least two and $Y = V \setminus X$ be the set of degree-one vertices. We iterate over all combinations of the following:

- An integer $k' \leq k$. If $\ell = 0$ or $s_{\min} > 1$, then we only consider $k' = k$.
- A partition of X into k' non-empty subsets $X_1, \dots, X_{k'}$. There are at most p^p such partitions.

For $i \in [k']$, let $Y_i \subseteq Y$ be the set of vertices from Y that are adjacent to a vertex from X_i. If $\ell = 0$ or $s_{\min} > 1$, then $k' = k$ and hence we accept if $(X_1 \cup Y_1, \dots, X_k \cup Y_k)$ is a solution and otherwise continue with the next combination.

Otherwise, we put all vertices from X_i in district V_i. Each vertex $v \in Y_i$ is either part of V_i or forms its own district. Thereby, for each $i \in [k']$, we know by applying Proposition 4 an interval $[\alpha_i, \beta_i]$ for the number of ℓ-fair districts respecting the size constraints in which the vertices $X_i \cup Y_i$ can be partitioned or we know that no solution exists (in this case, we continue with the next combination). We now check whether the given k lies between $\sum_{i \in [k']} \alpha_i$ and $\sum_{i \in [k']} \beta_i$ and return the answer.

As there exist p^p partitions of X, the running time of the algorithm is $\mathcal{O}(p^p \cdot k \cdot n)$.

7 Conclusion

We initiated a thorough study of the NP-hard FAIR CONNECTED DISTRICTING (FCD) problem. We considered FCD on specific graph classes and analyzed the parameterized complexity of FCD with a focus on the number of districts, the number of colors, and various graph parameters. We have shown that while FCD can be solved on simple graph classes in polynomial time (mostly using approaches based on dynamic programming), on trees it is already NP-hard and W[1]-hard with respect to the combined parameter number of colors plus number of districts. Nevertheless, for graph parameters such as the vertex cover number and the max leaf number, we developed XP-algorithms.

As to challenges for future research, we left open whether FCD is fixed-parameter tractable or W[1]-hard with respect to the vertex cover number or max leaf number. The former question is particularly intriguing because it is related to an open question of Stoica et al. [48] on the parameterized complexity of FAIR REGROUPING with respect to the number of districts (which has been very recently partly solved by Boehmer and Koana [10]): Recall that the input of FAIR REGROUPING is additionally endowed with a function $f : V \to 2^{[k]}$ with the requirement that every vertex $v \in V$ should belong to district V_i for some $i \in f(v)$ (and that the connectivity requirement is dropped). Consider the FCD instance on the bipartite graph with bipartition $(V, [k])$, where there is an edge between $i \in [k]$ and $v \in V$ if and only if $i \in f(v)$. The vertex cover number of this graph is at most the number k of districts. Under certain constraints, FCD on this graph becomes essentially equivalent to FAIR REGROUPING. Thus an algorithm for the vertex cover number for FCD would also shine further light on the complexity of FAIR REGROUPING with respect to the number of districts. The latter question concerning the max leaf number is interesting because FCD

could turn out to be one of few problems that are in XP yet W[1]-hard with respect to the max leaf number. It would be also promising to consider additional graph parameters (such as cliquewidth, treedepth or sparsity related parameters like the maximum degree or the degeneracy) or to examine further graph classes (such as grids).

From a broader perspective, there are several natural extensions of FCD. For instance, as already suggested by Stoica et al. [48], there may be a function that specifies for each vertex a set of districts to which the vertex can belong.[6] While all our hardness results still hold for this more complicated setting, it is open which of our algorithmic results can be adapted. Moreover, we did not study the generalization of FCD where each vertex has an integer weight for each color (such a study has been done in the context of gerrymandering by Cohen-Zemach et al. [15] and Ito et al. [29]). Weighted FCD is motivated by the following application scenarios: First, in some settings we might be restricted to put a certain group of agents always in the same district (those can be combined into one vertex). Second, if each vertex represents a voter and the colors represent alternatives, then voters might want to give points to different alternatives (such as done in the context of positional scoring rules).

Lastly, one can modify the definition of FCD. For instance, as done, among others, by Lewenberg et al. [38] and Eiben et al. [20] in the context of GERRYMANDERING and, among others, by Lu et al. [39] and Fotakis and Tzamos [24] in the context of facility location problems, instead of placing agents on a social network, the agents may be placed in a metric space. The task is then to place k ballot boxes in the space where each agent is assigned to the closest ballot box. The goal is again to make the resulting districts as fair as possible.

Appendix

Proof of proposition 4

proposition 4 (★). *Let $(G = (X \cup Y, E), C, \mathrm{col}, k, \ell, s_{\min}, s_{\max})$ be an* FCD *instance where G is a star with center vertex $v \in X$. Then, one can decide in linear time whether there is a partition of $X \cup Y$ into k connected ℓ-fair districts respecting $s_{\min}$ and $s_{\max}$ such that all vertices from X are part of the same district.*

Proof In the following, we call a partition of the vertices in $X \cup Y$ into k connected districts a solution if each district is ℓ-fair, respects the size constraints, and all vertices from X are part of the same district to which we refer as the center district. Note that each solution consists of the center district and several districts consisting of a single vertex from Y. Hence, we may assume that $s_{\min} = 1$. (For $s_{\min} \geq 2$, we can immediately conclude that there is no solution unless $k = 1$, $X \cup Y$ is ℓ-fair, and $|X \cup Y| \in [s_{\min}, s_{\max}]$.) We may also assume that $|X| \leq s_{\max}$. We now describe a linear-time algorithm that decides whether a given FCD instance admits a solution.

---

[6] However, it seems that in most applications where connectivity plays a crucial rule districts are "isomorphic" to each other, i.e., districts do not carry any particular meaning; for instance, when dividing a city into voting districts or people placed on a social network into teams for a competition. Thus, a priori, it is irrelevant for a vertex in which district it is put.

We start with several special cases: For $\ell = 0$, a solution exists if and only if $\mathrm{MOV}(X \cup Y) = 0$, $k = 1$, and $|X \cup Y| \leq s_{\max}$. If $|C| = 1$, then all solutions containing k districts are isomorphic, i.e., they contain $k - 1$ districts containing a single vertex and a single district containing the remaining vertices. Thus, the given instance admits a solution if and only if $|Y| \geq k - 1$, $\ell \geq |Y \cup X| - (k - 1)$, and $|Y \cup X| - (k - 1) \leq s_{\max}$.

Let $c_1 \in C$ be the most frequent color in $X \cup Y$ and $c_2 \in C$ be the second most frequent color in $X \cup Y$. If $\chi_{c_1}(X) > \chi_{c_2}(X \cup Y) + \ell$, then the given instance has no solution because the center district cannot be ℓ-fair.

Having dealt with several special cases separately, we now present a precise characterization of yes-instances assuming that $|C| > 1$, $\ell > 0$, $s_{\min} = 1$, $|X| \leq s_{\max}$, and $\chi_{c_1}(X) \leq \chi_{c_2}(X \cup Y) + \ell$. Let $c_1^\star$ be the most frequent color in X. Further, let $c_2^\star \neq c_1^\star$ be the most frequent color in X among colors c' such that $\chi_{c'}(X \cup Y) \geq \chi_{c_1^\star}(X) - \ell$. Note that $c_2^\star$ is always well-defined. If $c_1^\star \neq c_1$, then by the definition of c_1 it holds that $\chi_{c_1}(X \cup Y) \geq \chi_{c_1^\star}(X \cup Y) > \chi_{c_1^\star}(X) - \ell$. On the contrary, if $c_1^\star = c_1$, then, by our assumption that $\chi_{c_1}(X) \leq \chi_{c_2}(X \cup Y) + \ell$, we have $\chi_{c_2}(X \cup Y) \geq \chi_{c_1^\star}(X) - \ell$.

Let $b_l := \max(0, \chi_{c_1}(X \cup Y) - \chi_{c_2}(X \cup Y) - \ell)$ and $b_u := \max(0, \chi_{c_1^\star}(X) - \ell - \chi_{c_2^\star}(X))$. Then, there is a solution if and only if $k \in [b_l + 1, |Y| + 1 - b_u]$ and $k > |X| + |Y| - s_{\max}$.

We first prove the forward direction, i.e., that the existence of a solution implies that $k \in [b_l + 1, |Y| + 1 - b_u]$ and $k > |X| + |Y| - s_{\max}$. For any solution, the center district contains at most $\chi_{c_2}(X \cup Y) + \ell$ vertices of color c_1 (otherwise it cannot be ℓ-fair). Thus, at least $\chi_{c_1}(X \cup Y) - \chi_{c_2}(X \cup Y) - \ell$ vertices of color c_1 have to be part of a non-center district, which implies that $k > b_l$. Next, we show that $k \leq |Y| + 1 - b_u$. The center district has at least $\chi_{c_1^\star}(X)$ vertices of color $c_1^\star$. Since the center district is ℓ-fair, it has at least $\chi_{c_1^\star}(X) - \ell$ vertices of color c for some color $c \in C \setminus \{c_1^\star\}$. As only $\chi_c(X)$ of the $\chi_{c_1^\star}(X) - \ell$ vertices come from X, it follows that there are at least $\chi_{c_1^\star}(X) - \ell - \chi_c(X)$ vertices of color c from Y that are included in the center district. As $c_2^\star$ is the most frequent color in X among all colors occurring at least $\chi_{c_1^\star}(X) - \ell$ times, it follows that $\chi_{c_1^\star}(X) - \ell - \chi_{c_2^\star}(X)$ minimizes the expression from the previous sentence. Using this, we can conclude that $k \leq |Y| + 1 - b_u$. Finally, we show that $k > |X| + |Y| - s_{\max}$. Since the center district contains at most $s_{\max}$ vertices, at least $|X| + |Y| - s_{\max}$ vertices are not part of the center district, which implies that $k > |X| + |Y| - s_{\max}$.

We now prove the backward direction, i.e., that $k \in [b_l + 1, |Y| + 1 - b_u]$ and $k > |X| + |Y| - s_{\max}$ imply the existence of a solution. First, we show that if $k \in [b_l, |Y| + 1 - b_u]$, ignoring the size constraints, a solution always exists. We do this by first constructing two "extreme" solutions where as many or as few vertices as possible are contained in the center district. Recall that by our assumption $\chi_{c_1}(X) \leq \chi_{c_2}(X \cup Y) + \ell$. So we have $\chi_{c_1}(Y) = \chi_{c_1}(X \cup Y) - \chi_{c_1}(X) \geq \chi_{c_1}(X \cup Y) - \chi_{c_2}(X \cup Y) - \ell$, and hence $\chi_{c_1}(Y) \geq b_l$. A solution consisting of $b_l + 1$ districts exists: Put all vertices from $X \cup Y$ in one district and move b_l vertices of color c_1 from Y each in their own district (this is always possible by our above observation). Let S be the center district in this solution. S is ℓ-fair, as $\chi_{c_1}(S) \geq \chi_{c_2}(S)$, $\chi_{c_1}(S) = \chi_{c_1}(X \cup Y) - b_l \leq \chi_{c_2}(X \cup Y) - \ell = \chi_{c_2}(S) - \ell$, and $\chi_{c_2}(S) \geq \chi_c(S)$ for all $c \in C \setminus \{c_1\}$. A solution consisting of $|Y| + 1 - b_u$ districts is also guaranteed to exist: Put the vertices of X and b_u vertices of color $c_2^\star$ from Y into the center district; put all remaining vertices from Y into their own district. Let S' be the center district in this solution. S' is ℓ-fair, as $\chi_{c_1^\star}(S) - \ell = \chi_{c_1^\star}(X) - \ell \leq \chi_{c_2^\star}(X) + b_u = \chi_{c_2^\star}(S) \leq \chi_{c_1^\star}(S)$. Note that from these two solutions, we can construct a solution consisting of k districts for arbitrary $k \in [b_l + 1, |Y| + 1 - b_u]$. For this, we start with the first solution and one by one move vertices from $Y \cap (S \setminus S')$ from the center district into their own district while maintaining that the center district is ℓ-fair. Such a vertex always exists: If there is a vertex v of

the most frequent color of the center district in $Y \cap (S \setminus S')$, then we move v from S into its own district (removing v only increases the margin of victory by one if the center district is currently 0-fair and we have $\ell > 0$). Otherwise, we remove the remaining vertices from $Y \cap (S \setminus S')$ in an arbitrary order and put each into its own district (doing so results in an ℓ-fair center district, since otherwise S' would not be ℓ-fair). If $k > |X| + |Y| - s_{\max}$, then the above constructed solution for this k satisfies the size constraints as it consists of a center district containing $|X| + |Y| - (k - 1) \leq s_{\max}$ vertices where the inequality holds by our observation on k. $\square$

Proof of theorem 3

Theorem 3 ($\bigstar$). FCD *can be solved in* $\mathcal{O}(n^{\mathcal{O}(\mathrm{tw} \cdot |C|)})$ *time.*

Proof Suppose that we are given an instance $\mathcal{I} = (G, C, \mathrm{col}, k, \ell, s_{\min}, s_{\max})$ of FCD admitting a solution $\mathcal{V} = \{V_1, \dots, V_k\}$. Let $(T = (V_T, E_T), \{B_x\}_{x \in V_T})$ be a tree decomposition of G.

For $x \in V_T$, let G_x be the graph whose vertices and edges are those introduced in a node from the subtree of T rooted at x. Further, let U_x be the set of vertices in G_x. Our algorithm employs a dynamic programming method that traverses T from bottom to top.

Observe that each district $V_i \in \mathcal{V}$ from the solution is of one of the following three types with respect to each $x \in V_T$:

   (i) $V_i \cap B_x \neq \emptyset$, i.e., the district V_i overlaps with the vertices in the bag of x.
   (ii) $V_i \cap B_x = \emptyset$ and $V_i \subseteq U_x$, i.e., the district V_i is disjoint from B_x but fully contained in vertices from bags from the subtree rooted at x.
   (iii) $V_i \cap B_x = \emptyset$ and $V_i \subseteq V \setminus U_x$, i.e., the district V_i is disjoint from B_x and does not contain any vertex occurring in a bag from the subtree rooted at x.

These three cases are exhaustive, as by definition of the tree decomposition, for each vertex, all nodes in which bags the vertex appears form a connected subgraph in T. Note that for each district $V_i \in \mathcal{V}$ of type (i), the induced subgraph $G_x[V_i]$ can have multiple connected components, in which case $G[V_i]$ includes some vertices or edges not in G_x.

For each $x \in V_T$, we keep track of the following information:

(1)   the intersection of B_x and every $V_i \in \mathcal{V}$ of type (i),
(2)   the intersection of B_x with the connected components of the subgraph $G_x[V_i]$ for all districts $V_i \in \mathcal{V}$ of type (i),
(3)   the number of occurrences of color c for every color $c \in C$ in $V_i \cap U_x$ for every district $V_i \in \mathcal{V}$ of type (i), and
(4)   the number of districts of type (ii).

We capture this information in the following variables: For each $x \in V_T$, let (1) $\mathcal{B}$ be a partition of B_x, (2) $\mathcal{D}$ be a partition of B_x where each subset from $\mathcal{D}$ is contained in a subset of $\mathcal{B}$, (3) $\zeta : \mathcal{B} \to \mathbb{N}^{|C|}$ be some function, and (4) $\kappa \in \mathbb{N}$ be some integer with $\kappa \leq k$.

On an intuitive level, moving in the tree from the bottom to the top, we keep track of the number of "completed" districts (type (ii) districts). Moreover, for each "unfinished" district (type (i) district), we store its current color distribution (and thereby implicitly its current size of so far added vertices), and how the vertices from the current bag are distributed to the unfinished districts. Also, for each unfinished district, we keep track of which of its vertices from the current bag are already connected over vertices that have been added to the district and edges that have been introduced in the respective subtree. It is necessary to keep track of the connectivity of vertices from one district (stored in $\mathcal{D}$), as we need this information to ensure that we make the district connected by adding further vertices and edges to it further up in the tree. Moreover, we cannot directly infer it from $\mathcal{B}$, as we cannot remember all vertices put in the districts so far.

It remains to formally describe the dynamic program we use. For all combinations of the variables $\mathcal{B}$, $\mathcal{D}$, ζ, and κ, we compute $A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa) \in \{0, 1\}$ such that $A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = 1$ if and only if there is a partition of the vertices U_x into exactly $|\mathcal{D}| + \kappa$ subsets $(W_1, \dots, W_{|\mathcal{D}|+\kappa})$ with the following properties:

- For every $i \in [|\mathcal{D}| + \kappa]$, $G_x[W_i]$ is connected.
- Exactly κ subsets from $(W_1, \dots, W_{|\mathcal{D}|+\kappa})$ are disjoint from B_x. Moreover, all of them are ℓ-fair and respect the size constraints.
- For every subset W_i intersecting B_x, it holds that $W_i \cap B_x \in \mathcal{D}$.
- For every $B \in \mathcal{B}$, it holds that $\zeta(B) = \sum_{i \in [|\mathcal{D}|+\kappa], W_i \cap B \neq \emptyset} \chi(W_i)$.

From the subsets in the partitioning $(W_1, \dots, W_{|\mathcal{D}|+\kappa})$, κ are ℓ-fair connected districts of type (ii) with respect to x. The other $|\mathcal{D}|$ subsets are partly contained in B_x and restricted to B_x form the partition $\mathcal{D}$ describing the intersection of B_x with the connected components of $G_x[V_i]$ for districts V_i of type (i). Note that some of these $|\mathcal{D}|$ subsets may end up in the same district in the solution. Moreover, assuming that $\mathcal{B}$ corresponds to the intersections of B_x and districts of type (i), it needs to hold that for each set in the intersection $B \in \mathcal{B}$, $\zeta(B)$ is the distribution of colors in the union of all subsets W_i intersecting with B (these are the subsets that will finally end up being one district in the solution).

Observe that a given instance of FCD is a yes-instance if and only if there exists a partition $\mathcal{B}_r$ of B_r for the root r of T and a function ζ_r such that for all $B \in \mathcal{B}_r$, $\zeta_r(B)$ is ℓ-fair and its L_1-norm is in the range $[s_{\min}, s_{\max}]$ and $A_r(\mathcal{B}_r, \mathcal{B}_r, \zeta_r, k - |\mathcal{B}_r|) = 1$.

We compute $A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa)$ by traversing the nodes V_T of the tree T from bottom to top. Depending on the type of the current node $x \in V_T$, we apply one of the following five cases. In the following, let $y \in V_T$ (and $z \in V_T$) be the child node(s) of x if x has one (two) child node(s) in T.

Leaf node x.

For a leaf node x with $B_x = \{v\}$, we have $A_x(\{v\}, \{v\}, \zeta, 0) = 1$ if and only if $\zeta(\{v\}) = \chi(\{v\})$.

Introduce vertex v node x.

Note that v is isolated in G_x. Thus, $A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = 0$ if $\{v\}$ is not contained in $\mathcal{D}$. Otherwise we have

$$A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = A_y(\mathcal{B}', \mathcal{D}', \zeta', \kappa),$$

where $\mathcal{B}' = (\mathcal{B}\setminus\{B\}) \cup \{B\setminus\{v\}\}$ for $B \in \mathcal{B}$ with $v \in B$ ($\mathcal{B}' = \mathcal{B}\setminus\{B\}$ if $B = \{v\}$), $\mathcal{D}' = \mathcal{D} \setminus \{\{v\}\}$, and $\zeta'(B') = \zeta(B) - \chi(\{v\})$ if $B' = B\setminus\{v\}$ (we assume that $\zeta'(\emptyset)$ is a zero vector here) and $\zeta'(B') = \zeta(B')$ otherwise.

Introduce edge $\{u, v\}$ **node** x.

If u and v belong to distinct subsets of $\mathcal{B}$, then we have $A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = A_y(\mathcal{B}, \mathcal{D}, \zeta, \kappa)$. We also have the same recurrence if u and v belong to the same subset of $\mathcal{D}$. This is because in both cases the connected components of districts of type (i) in G_x do not change. Otherwise, we have

$$A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = A_y(\mathcal{B}, \mathcal{D}, \zeta, \kappa) \vee \bigvee_{\mathcal{D}'} A_y(\mathcal{B}, \mathcal{D}', \zeta, \kappa),$$

where $\bigvee$ is over all partitions $\mathcal{D}'$ of B_y from which $\mathcal{D}$ can be obtained by removing $D_u, D_v \in \mathcal{D}'$ with $u \in D_u$ and $v \in D_v$ from $\mathcal{D}'$ and adding $D_u \cup D_v$ to $\mathcal{D}'$.

Forget v **node** x.

We have two cases. In the first case, the district $V_i \in \mathcal{V}$ to which v belongs does not intersect B_x. This implies that V_i is now fully contained in G_x and becomes a district of type (ii). Then $\{v\}$ should have been part of $\mathcal{B}$ and $\mathcal{D}$ for child y. Let us define

$$A_x^1(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = \bigvee_{\zeta'} (A_y(\mathcal{B} \cup \{\{v\}\}, \mathcal{D} \cup \{\{v\}\}, \zeta', \kappa - 1)),$$

where $\bigvee$ is over all functions ζ' such that $\zeta'(B) = \zeta(B)$ for all $B \in \mathcal{B}$, $\mathrm{MOV}(\zeta'(\{v\})) \leq \ell$, and $s_{\min} \leq \sum_{c \in C} \zeta_c'(\{v\}) \leq s_{\max}$. Note that we also need to decrease κ, as V_i is an additional district of type (ii) for the node x. In the other case, the district in which v is contained intersects B_x. In this case, it holds that

$$A_x^2(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = \bigvee_{\mathcal{B}', \mathcal{D}'} A_y(\mathcal{B}', \mathcal{D}', \zeta, \kappa),$$

where $\bigvee$ is over all partitions $\mathcal{B}'$ and $\mathcal{D}'$ of B_y such that $\mathcal{B}' = (\mathcal{B} \setminus \{B\}) \cup (B \cup \{v\})$ and $\mathcal{D}' = (\mathcal{D} \setminus \{D\}) \cup (D \cup \{v\})$ for some $B \in \mathcal{B}$ and $D \in \mathcal{D}$. All in all, we have $A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa) = A_x^1(\mathcal{B}, \mathcal{D}, \zeta, \kappa) \vee A_x^2(\mathcal{B}, \mathcal{D}, \zeta, \kappa)$.

Join node x.

Note that two vertices $u, w \in D$ for some $D \in \mathcal{D}$ can reach each other in G_x via connections in G_y and G_z. So we have

$$A_x(\mathcal{B}, \mathcal{D}, \zeta, \kappa)$$
$$= \bigvee_{\mathcal{D}_y, \mathcal{D}_z, \zeta_y, \zeta_z, \kappa_y, \kappa_z} A_y(\mathcal{B}, \mathcal{D}_y, \zeta_y, \kappa_y) \wedge A_z(\mathcal{B}, \mathcal{D}_z, \zeta_z, \kappa_z).$$

Here, $\bigvee$ is over all $\mathcal{D}_y, \mathcal{D}_z, \zeta_y, \zeta_z, \kappa_y, \kappa_z$ such that all the following hold:

- $\mathcal{D}$ are the connected components of the graph whose vertex set is B_x and edge set is $\{\{u, w\} \mid (\exists D_y \in \mathcal{D}_y : u, w \in D_y) \vee (\exists D_z \in \mathcal{D}_z : u, w \in D_z)\}$.
- $\zeta_y(B) + \zeta_z(B) = \zeta(B) - \chi(B)$ for every $B \in \mathcal{B}$.
- $\kappa_y + \kappa_z = \kappa$.

Now we examine the running time. Note that there are $\mathrm{tw}^{\mathcal{O}(\mathrm{tw})}$ partitions of B_x for every node x of T. Since $\zeta(B) \in \{0, \dots, n\}^{|C|}$ for every $B \in B_x$ and $x \in V_T$, there are at most $n^{\mathcal{O}(\mathrm{tw} \cdot |C|)}$ states for ζ. It is easy to see that the computation of all values takes $n^{\mathcal{O}(\mathrm{tw} \cdot |C|)}$ time. $\qquad\square$

References

1. Agarwal, A., Elkind, E., Gan, J., Igarashi, A., Suksompong, W., & Voudouris, A. A. (2021). Schelling games on graphs. *Artificial Intelligence 301*, 103576.
2. Autry, E. A., Carter, D., Herschlag, G. J., Hunter, Z., & Mattingly, J. C. (2021). Metropolized multi-scale forest recombination for redistricting. *Multiscale Modelling Simulation, 19*(4), 1885–1914.
3. Bachrach, Y., Lev, O., Lewenberg, Y., & Zick, Y. (2016). Misrepresentation in district voting. In *Proceedings of the 25th international joint conference on artificial intelligence (IJCAI '16)*. AAAI Press, (pp. 81–87).
4. Banerjee, A.V., & Duflo, E. (2011). *Poor economics: A radical rethinking of the way to fight global poverty*. Public Affairs.
5. Banerjee, A.V., & Pande, R. (2007). *Parochial politics: Ethnic preferences and politician corruption*.
6. Bentert, M., Koana, T., & Niedermeier, R. (2021). *The complexity of gerrymandering over graphs: Paths and trees. In: Proceedings of the 47th international workshop on graph-theoretic concepts in computer science (WG '21)*. Springer, (pp. 195–206).
7. van Bevern van Bevern, R., Bredereck, R., Chen, J., Froese, V., Niedermeier, R., & Woeginger, G. J. (2015). Network-based vertex dissolution. *SIAM Journal Discrete Mathematics* 29(2), 888–914.
8. Bhakta, P., Miracle, S., & Randall, D. (2014). Clustering and mixing times for segregation models on z^2. In *Proceedings of the 25th annual ACM-SIAM symposium on discrete algorithms (SODA '14)*. SIAM, (pp. 327–340).
9. Boehmer, N., Bredereck, R., Knop, D., & Luo, J. (2020). Fine-grained view on bribery for group identification. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence (IJCAI '20)*. (pp. 67–73). ijcai.org
10. Boehmer, N., & Koana, T. (2022). The complexity of finding fair many-to-one matchings. In *Proceedings of the 49th international colloquium on automata, languages, and programming (ICALP '22)*. (pp. 27:1–27:18).
11. Brandt, C., Immorlica, N., Kamath, G., & Kleinberg, R. (2012). An analysis of one-dimensional Schelling segregation. In *Proceedings of the 44th symposium on theory of computing conference (STOC '12)*. ACM, (pp. 789–804).
12. Brill, M., Schmidt-Kraepelin, U., & Suksompong, W. (2022). Margin of victory for tournament solutions. *Artificial Intelligence, 302*, 103600.
13. Campagna, J., & Grofman, B. (1990). Party control and partisan bias in 1980s congressional redistricting. *The Journal of Politics, 52*(4), 1242–1257.
14. Chlebíková, J. (1996). Approximating the maximally balanced connected partition problem in graphs. *Information Processing Lettering, 60*(5), 223–230.
15. Cohen-Zemach, A., Lewenberg, Y., & Rosenschein, J. S. (2018). Gerrymandering over graphs. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems (AAMAS '18)*. IFAAMAS, (pp. 274–282).
16. DeFord, D., Duchin, M., & Solomon, J. (2021). Recombination: a family of Markov chains for redistricting. *Harvard Data Science Review*.

17. Dey, P., & Narahari, Y. (2015). Estimating the margin of victory of an election using sampling. In *Proceedings of the twenty-fourth international joint conference on artificial intelligence (IJCAI '15)*. AAAI Press, (pp. 1120–1126).
18. Dyer, M. E., & Frieze, A. M. (1985). On the complexity of partitioning graphs into connected subgraphs. *Discrete Applied Mathematics, 10*(2), 139–153.
19. EdBuild: Non-white school districts get $23 billion less than white districts, despite serving the same number of students (2019), edbuild.org/content/23-billion
20. Eiben, E., Fomin, F. V., Panolan, F., & Simonov, K. (2020). Manipulating districts to win elections: Fine-grained complexity. In *Proceedings of the 34th AAAI conference on artificial intelligence (AAAI '20)*. AAAI Press, (pp. 1902–1909).
21. Engstrom, E. J. (2006). Stacking the states, stacking the house: the partisan consequences of congressional redistricting in the 19th century. *American Political Science Review, 100*(3), 419–427.
22. Eppstein, D. (2015). Metric dimension parameterized by max leaf number. *Journal of Graph Algorithms Application, 19*(1), 313–323.
23. Erikson, R. S. (1972). Malapportionment, gerrymandering, and party fortunes in congressional elections. *American Political Science Review, 66*(4), 1234–1245.
24. Fotakis, D., & Tzamos, C. (2014). On the power of deterministic mechanisms for facility location games. *ACM Transaction on Economic and Computation, 2*(4), 15:1-15:37.
25. Gabow, H. N. (1983). An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the 15th annual ACM symposium on theory of computing (STOC '83)*. ACM, (pp. 448–456).
26. Gupta, S., Jain, P., Panolan, F., Roy, S., & Saurabh, S. (2021). Gerrymandering on graphs: computational complexity and parameterized algorithms. In: Proceedings of the 14th international symposium on algorithmic game theory (SAGT '21). Springer, (pp. 140–155).
27. Hirsch, S. (2003). The united states house of unrepresentatives: What went wrong in the latest round of congressional redistricting. *Election Law Journal, 2*(2), 179–216.
28. Issacharoff, S. (2002). Gerrymandering and political cartels. *Harvard Law Review, 116*, 593–648.
29. Ito, T., Kamiyama, N., Kobayashi, Y., & Okamoto, Y. (2021). Algorithms for gerrymandering over graphs. *Theoritical Computer Science, 868*, 30–45.
30. Kannan, R. (1987). Minkowski's convex body theorem and integer programming. *Mathematics of Operation Research, 12*(3), 415–440.
31. Kleitman, D. J., & West, D. B. (1991). Spanning trees with many leaves. *SIAM Journal on Discrete Mathematics, 4*(1), 99–106.
32. Kloks, T. (1994). *Treewidth, computations and approximations* lecture notes in computer science (Vol. 842). New York: Springer.
33. Kreisel, L., Boehmer, N., Froese, V., & Niedermeier, R. (2021). Equilibria in schelling games: computational complexity and robustness. In *Proceedings of the 21st international conference on autonomous agents and multiagent systems (AAMAS '22)*. IFAAMAS, (pp. 761–769).
34. Landau, Z., Reid, O., & Yershov, I. (2009). A fair division solution to the problem of redistricting. *Social Choice and Welfare, 32*(3), 479–492.
35. Landau, Z., & Su, F. E. (2013). *Fair division and redistricting AMS Special Sessions on The Mathematics of Decisions Elections and Games*. Rodhe Island: American mathematical society, (pp. 17–36).
36. Lenstra, H. W. (1983). Integer programming with a fixed number of variables. *Mathematics of Operation Research, 8*, 538–548.
37. Levin, H. A., & Friedler, S. A. (2019). Automated congressional redistricting. *ACM Journal of Experimental Algorithmics, 24*(1), 1.10:1-1.10:24.
38. Lewenberg, Y., Lev, O., & Rosenschein, J. S. (2017). Divide and conquer: Using geographic manipulation to win district-based elections. In *Proceedings of the 16th conference on autonomous agents and multiagent systems (AAMAS '17)*. ACM, (pp. 624–632).
39. Lu, P., Sun, X., Wang, Y., & Zhu, Z. A. (2010). Asymptotically optimal strategy-proof mechanisms for two-facility games. In *Proceedings of the 11th ACM conference on electronic commerce (EC '10)*. ACM, (pp. 315–324).
40. Lublin, D. (1999). *The Paradox of Representation: Racial Gerrymandering and Minority Interests in Congress*. New Jersey: Princeton University Press.
41. Marx, D. (2007). On the optimality of planar and geometric approximation schemes. In *Proceedings of the 48th annual IEEE symposium on foundations of computer science (FOCS 07)*. (pp. 338–348).
42. McCartan, C., Imai, K. (2020). Sequential monte carlo for sampling balanced and compact redistricting plans. CoRR http://arxiv.org/2008.06131
43. Mitra, A. (2020). Electoral david versus goliath: how does the spatial concentration of electors affect district-based elections? http://arxiv.org/2006.11865

44. Puppe, C., & Tasnádi, A. (2008). A computational approach to unbiased districting. *Mathematical and Computer Modelling, 48*(9–10), 1455–1460.
45. Puppe, C., & Tasndi, A. (2009). Optimal redistricting under geographical constraints: Why "pack and crack'' does not work. *Economics Letters, 105*(1), 93–96.
46. Schaefer, T. J. (1978). The complexity of satisfiability problems. In *Proceedings of the 10th annual ACM symposium on theory of computing (STOC '78)*. ACM, (pp. 216–226).
47. Schelling, T. C. (1969). Models of segregation. *American Economic Review, 59*(2), 488–493.
48. Stoica, A., Chakraborty, A., Dey, P., & Gummadi, K. P. (2020). Minimizing margin of victory for fair political and educational districting. In *Proceedings of the 19th international conference on autonomous agents and multiagent systems (AAMAS '20)*. IFAAMAS, (pp. 1305–1313).
49. Xia, L. (2012). Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM conference on electronic commerce (EC '12)*. ACM, (pp. 982–999).
50. Zhao, Z., Hettle, C., Gupta, S., Mattingly, J., Randall, D., & Herschlag, G. (2022). Mathematically quantifying gerrymandering and the non-responsiveness of the 2021 Georgia congressional districting plan. In *Proceedings of the second ACM conference on equity and access in algorithms, mechanisms, and optimization (EAAMO ''22)*. ACM, (pp. 11–15).