Equitable Connected Partition and Structural Parameters Revisited: N-Fold Beats Lenstra

Václav Blažej
University of Warwick, UK

Dušan Knop
Czech Technical University in Prague, Czech Republic

Jan Pokorný
Czech Technical University in Prague, Czech Republic

Šimon Schierreich
Czech Technical University in Prague, Czech Republic

Abstract

In the Equitable Connected Partition (ECP for short) problem, we are given a graph $G = (V, E)$ together with an integer $p \in \mathbb{N}$, and our goal is to find a partition of V into p parts such that each part induces a connected sub-graph of G and the size of each two parts differs by at most 1. On the one hand, the problem is known to be NP-hard in general and W[1]-hard with respect to the path-width, the feedback-vertex set, and the number of parts p combined. On the other hand, fixed-parameter algorithms are known for parameters the vertex-integrity and the max leaf number.

In this work, we systematically study ECP with respect to various structural restrictions of the underlying graph and provide a clear dichotomy of its parameterised complexity. Specifically, we show that the problem is in FPT when parameterized by the modular-width and the distance to clique. Next, we prove W[1]-hardness with respect to the distance to cluster, the 4-path vertex cover number, the distance to disjoint paths, and the feedback-edge set, and NP-hardness for constant shrub-depth graphs. Our hardness results are complemented by matching algorithmic upper-bounds: we give an XP algorithm for parameterisation by the tree-width and the distance to cluster. We also give an improved FPT algorithm for parameterisation by the vertex integrity and the first explicit FPT algorithm for the 3-path vertex cover number. The main ingredient of these algorithms is a formulation of ECP as N-fold IP, which clearly indicates that such formulations may, in certain scenarios, significantly outperform existing algorithms based on the famous algorithm of Lenstra.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms; Theory of computation → Graph algorithms analysis; Mathematics of computing → Graph algorithms

Keywords and phrases Equitable Connected Partition, structural parameters, fixed-parameter tractability, N-fold integer programming, tree-width, shrub-depth, modular-width

Digital Object Identifier 10.4230/LIPIcs.MFCS.2024.29

Related Version *Full Version*: `https://arxiv.org/abs/2404.18968`

Funding This work was co-funded by the European Union under the project Robotics and advanced industrial production (reg. no. CZ.02.01.01/00/22_008/0004590).
Václav Blažej: Supported by the Engineering and Physical Sciences Research Council [grant number EP/V044621/1].
Dušan Knop: Acknowledges the support of the Czech Science Foundation Grant No. 22-19557S.
Jan Pokorný: Acknowledges the support of the Grant Agency of the CTU in Prague funded grant No. SGS23/205/OHK3/3T/18.
Šimon Schierreich: Acknowledges the support of the Czech Science Foundation Grant No. 22-19557S and of the Grant Agency of the CTU in Prague funded grant No. SGS23/205/OHK3/3T/18.

Acknowledgements We are grateful to all anonymous referees for their valuable comments.

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 29; pp. 29:1–29:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A partition of a set V into $p \in \mathbb{N}$ parts is a set $\pi = \{V_1, \ldots, V_p\}$ of subsets of V such that for every $i, j \in [p]$: $V_i \cap V_j = \emptyset$ and $\bigcup_{i=1}^{p} V_i = V$. In the Equitable Connected Partition problem, we are given an undirected graph $G = (V, E)$ together with an integer p, and our goal is to partition the vertex set V into p *parts* such that the sizes of each two parts differ by at most 1 and each class induces a connected sub-graph. Formally, our problem is defined as follows.

Equitable Connected Partition (ECP)

Input: A simple undirected and connected n-vertex graph $G = (V, E)$ and a positive integer $p \in \mathbb{N}$.

Question: Is there a partition $\pi = \{V_1, \ldots, V_p\}$ of V such that every part $G[V_i]$ is connected, and $||V_i| - |V_j|| \leq 1$ for every pair $i, j \in [p]$?

The Equitable Connected Partition problem naturally arises in many fields such as redistricting theory [2, 50, 55], which is a subfield of computational social choice theory, VLSI circuit design [7], parallel computing [5], or image processing [52], to name a few.

One of the most prominent problems in the graph partitioning direction is the Bisection problem, where our goal is to split the vertex set into two parts A and B, each part of size at most $\lceil \frac{n}{2} \rceil$, such that the number of edges between A and B is at most some given $k \in \mathbb{N}$. Bisection is NP-hard [36] even if we restrict the input to unit disc graphs [24] and is heavily studied from the parameterised complexity perspective; see, e.g., [6, 13, 19, 30, 54]. The natural generalisation of the Bisection problem is called Balanced Partitioning where we partition the vertices into $p \in \mathbb{N}$ parts, each of size at most $\lceil \frac{n}{p} \rceil$. Balanced Partitioning is NP-hard already on trees [27] and a disjoint union of cliques [4]. The parameterised study of this problem is due to Ganian and Obdržálek [35] and van Bevern et al. [6]. In all the aforementioned problems, we are given only the upper-bound on the size of each part; hence, the parts are not necessarily equitable. Moreover, there is no connectivity requirement for the parts. For a survey of graph partitioning problems, we refer the reader to the monograph of Buluç et al. [14].

On the equitability side, the most notable direction of research is the Equitable k-Colouring problem (EC for short). Here, we are given an undirected graph G and the goal is to decide whether there is a proper colouring of the vertices of G using k colours such that the sizes of each two colour classes differ by at most one. Note that the graph induced by each colour class is necessarily an independent set, and hence is disconnected. As the k-Colouring problem can be easily reduced to the Equitable k-Colouring, it follows that EC is NP-hard. Polynomial-time algorithms are known for many simple graph classes, such as graphs of bounded tree-width [9, 16], split graphs [15], and many others [31]. The parameterised study was, to the best of our knowledge, initiated by Fellows et al. [28] and continued in multiple subsequent works [26, 29, 40]. For a detailed survey of the results on EC, we refer the reader to the monograph by Lih [51].

The Equitable Connected Partition problem then naturally brings the concepts of equitability and connectivity of the vertex set together. It is known that ECP is NP-complete [2]. Moreover, the problem remains NP-complete even if G is a planar graph or for every fixed p at least 2 [23, 36]. Enciso et al. [26] were the first who studied ECP from the viewpoint of parameterised complexity. They showed that ECP is fixed-parameter tractable with respect to the vertex cover number and the maximum leaf number of G. On the negative side, they showed that it is W[1]-hard to decide the problem for the combined parameter the

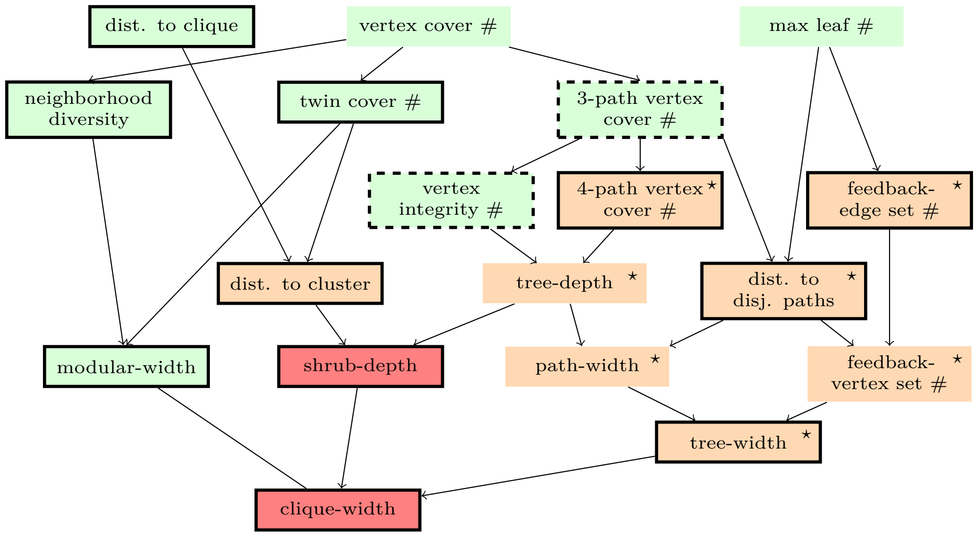

Figure 1 An overview of our results. The parameters for which the problem is in FPT are coloured green, the parameters for which ECP is W[1]-hard and in XP have an orange background, and para-NP-hard combinations are highlighted in red. Arrows indicate generalisations; e.g., modular width generalises both neighbourhood diversity and twin-cover number. The solid thick border represents completely new results, and the dashed border represents an improvement of previously known algorithms. All our W[1]-hardness results hold even when the problem is additionally parameterized by the number of parts p. Additionally, we show that the results marked with ⋆ becomes fixed-parameter tractable if the size of a larger part $\lceil n/p \rceil$ is an additional parameter.

path-width, the feedback-vertex set, and the number of parts p. Moreover, they gave an XP algorithm for ECP parameterised by tree-width. Later, Gima et al. [38] showed that the problem is fixed-parameter tractable when parameterised by the vertex-integrity of G. Very recently, Gima and Otachi [39] proved that ECP is W[1]-hard when parameterised by the tree-depth of G.

A more general variant of Equitable Connected Partition with parametric lower- and upper-bounds on the sizes of parts was studied by Ito et al. [43], and Blažej et al. [8] very recently introduced the requirement on the maximum diameter of each part.

It is worth pointing out that Equitable Connected Partition is also significant from a theoretical point of view. Specifically, this problem is a very common starting point for many W[1]-hardness reductions; see, e.,g., [6, 8, 20, 53]. Surprisingly, the graph in multiple of the before-mentioned reductions remains the same as in original instance, and therefore our study directly strengthens the results obtained in these works. Since the complexity picture with respect to structural parameters is rather incomplete, many natural questions arise. For example, what is the parameterised complexity of ECP when parameterised by the 4-path vertex cover number? Or, is ECP in FPT when parameterised by the feedback-edge set? Last but not least, is the problem easier to decide on graphs that are dense, such as cliques?

1.1 Our Contribution

In our work, we continue the line of study of the Equitable Connected Partition problem initiated by Enciso et al. [26] almost 15 years ago. For an overview of our results, we refer the reader to Figure 1; however, we believe that our contribution is much broader. We try to summarise it in the following four points.

First, directly following pioneering work on structural parameterisation of ECP, we provide a complete dichotomy between tractable and intractable cases for structural parameters that are bounded for sparse graphs. Namely, we provide W[1]-hardness proofs for ECP with respect to the 4-path vertex cover number and the feedback-edge set number, which encloses a gap between structural parameters that were known to be tractable – the vertex-cover number and the max-leaf number – and those that were known to be W[1]-hard– the path-width and the feedback-vertex set. It should also be mentioned that our constructions not only give much stronger intractability results but, at the same time, are much simpler compared to the original construction of Enciso et al. [26].

Second, we also turn our attention to dense graphs, which have, so far, been completely overlooked in the relevant literature. On our way to fixed-parameter tractable algorithms for various structural parameters, we prove polynomial-time solvability of some specific graph classes. Again, we provide a clear boundary between tractable and intractable cases. However, it turns out that for dense graphs, the problem is much easier.

Third, we clearly show where the limits of the parameterized complexity framework in the study of structural parameterisation of Equitable Connected Partition are. In particular, we show that the problem is NP-hard already on graphs of shrub-depth equal to 3, clique-width equal to 3, and twin-width equal to 2. Moreover, in some cases, our complexity results are tight. For example, we give a polynomial-time algorithm for graphs of clique-width 2.

Last but not least, in order to provide all the algorithms, we use multiple different techniques. Naturally, some algorithms are based on standard techniques such as dynamic-programming over decomposition or kernelisation; however, many of them still require deep insights into the structure of the solution and the instances. However, some of them use very careful branching together with formulation of the problem using N-fold integer linear programming, which is, informally speaking, an integer linear program with specific shape of the constraints. We are convinced that the technique of N-fold integer linear programming in the design and analysis of fixed-parameter tractable algorithms deserves more attention from the parameterised complexity community, as it in many scenarios significantly outperforms the classical FPT algorithms based on the famous Lenstra's algorithm; see, e.g., [3, 12, 45, 46, 47].

2 Preliminaries

We assume the reader to be familiar with the basic graph-theoretical notation as given by Diestel [21] and with the basics of parameterised complexity [18].

Structural Parameters. In this sub-section, we provide definitions of less widespread structural parameters we study in this work. Definitions of all the remaining parameters are provided in the full version.

▶ **Definition 1** (d-path vertex cover). *Let $G = (V, E)$ be an undirected graph and $d \in \mathbb{N}$ be an integer. A* d-path vertex cover *is a set $C \subseteq V$ such that the graph $G \setminus C$ contains no path with d vertices as a sub-graph. The* d-path vertex cover number $d\text{-pvcn}(G)$ *is the size of a minimum d-path vertex cover in G.*

Note that the 2-path vertex cover number is, in fact, the standard *vertex cover number* of a graph.

▶ **Definition 2** (Vertex integrity). *Let $G = (V, E)$ be an undirected graph.* Vertex integrity*, denoted* $\mathrm{vi}(G)$*, is the minimum $k \in \mathbb{N}$ such that there is a set $X \subseteq V$ of size at most k and every connected component C of $G \setminus X$ contains at most k vertices.*

▶ **Definition 3** (Distance to $\mathcal{G}$). *Let G be a graph, and $\mathcal{G}$ be a graph family. A set $M \subseteq V(G)$ is a* modulator *to $\mathcal{G}$, if $G \setminus M \in \mathcal{G}$. The* distance to $\mathcal{G}$*, denoted* $\mathrm{dist}_{\mathcal{G}}(G)$*, is the size of a minimum modulator to $\mathcal{G}$.*

In our paper, we will focus on the *distance to clique*, denoted by $\mathrm{dc}(G)$, the distance to disjoint union of cliques, which is usually referred to as the *distance to cluster graph* or the *cluster vertex deletion* [22] and denoted $\mathrm{dcg}(G)$, and the *distance to disjoint paths*, denoted by $\mathrm{ddp}(G)$.

▶ **Definition 4** (Neighbourhood-diversity [48]). *Let $G = (V, E)$ be a graph. We say that two vertices $u, v \in V$ have the same type iff $N(u) \setminus \{v\} = N(v) \setminus \{u\}$. The* neighbourhood diversity *of G is at most d, if there exists a partition of V into at most d sets such that all vertices in each set have the same type.*

Let $T_1, \ldots, T_d$ be a partition of V such that for each $u, v \in T_i$, $i \in [d]$, it holds that u and v are of the same type according to Definition 4. Observe that each type is either independent set or a clique. We define *type graph* to be an undirected graph with vertices being the types $T_1, \ldots, T_d$ and two vertices corresponding to some types T_i and T_j are connected by an edge iff there exists an edge $\{u, v\} \in E(G)$ such that $u \in T_i$ and $v \in T_j$.

▶ **Definition 5** (Modular-width [32]). *Consider graphs that can be obtained from an algebraic expression that uses only the following operations:*

1. *create an isolated vertex,*
2. *the disjoint union of two disjoint graphs G_1 and G_2 which is a graph $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2))$,*
3. *the complete join of two disjoint graphs G_1 and G_2 which produces a graph $(V(G_1) \cup V(G_2), E(G_1) \cup E(G_2) \cup \{\{u, v\} \mid u \in V(G_1) \text{ and } v \in V(G_2)\})$.*
4. *the substitution with respect to some pattern graph P – for a graph P with vertices $p_1, \ldots, p_\ell$ and disjoint graphs $G_1, \ldots, G_\ell$, the substitution of the vertices of P by the graphs $G_1, \ldots, G_\ell$ is the graph with vertex set $\bigcup_{i=1}^{\ell} V(G_i)$ and edge set $\bigcup_{i=1}^{\ell} E(G_i) \cup \{\{u, v\} \mid u \in V(G_i),\ v \in V(G_j),\ \text{and } \{p_i, p_j\} \in E(P)\}$.*

The width of such an algebraic expression is the maximum number of operands used by any occurrence of the substitution operation. The modular-width *of a graph G, denoted* $\mathrm{mw}(G)$*, is the least integer m such that G can be obtained from such algebraic expression of width m.*

N-fold Integer Programming. In recent years, integer linear programming (ILP) has become a very useful tool in the design and analysis of fixed-parameter tractable algorithms [37]. One of the best known results in this line of research is probably Lenstra's algorithm, roughly showing that ILP with bounded number of variables is solvable in FPT time [49].

In this work, we use the so-called *N-fold integer programming* formulation. Here, the problem is to minimise a linear objective over a set of linear constraints with a very restricted structure. In particular, the constraints are as follows. We use $x^{(i)}$ to denote a set of t_i variables (a so-called *brick*).

$$D_1 x^{(1)} + D_2 x^{(2)} + \cdots + D_N x^{(N)} = \mathbf{b}_0 \tag{1}$$

$$A_i x^{(i)} = \mathbf{b}_i \qquad \forall i \in [N] \tag{2}$$

$$\mathbf{0} \le x^{(i)} \le \mathbf{u}_i \qquad \forall i \in [N] \tag{3}$$

Where we have $D_i \in \mathbb{Z}^{r \times t_i}$ and $A_i \in \mathbb{Z}^{s_i \times t_i}$; let us denote $s = \max_{i \in [N]} s_i$, $t = \max_{i \in [N]} t_i$, and let the dimension be d, i.e., $d = \sum_{i \in [N]} t_i \leq Nt$. Constraints (1) are the so-called *linking constraints* and the rest are the *local constraints*. In the analysis of our algorithms, we use the following result of Eisenbrand et al. [25].

▶ **Proposition 6** ([25, Corollary 91]). *N-fold IP can be solved in $a^{r^2 s + rs^2} \cdot d \cdot \log(d) \cdot L$ time, where*

- *L is the maximum feasible value of the objective and*
- *$a = r \cdot s \cdot \max_{i \in [N]} (\max(\|D_i\|_\infty, \|A_i\|_\infty))$.*

3 Algorithmic Results

In this section, we provide our algorithmic results. The first algorithm is for ECP parameterised by the vertex-integrity and combines careful branching with N-fold integer programming. Specifically, Gima et al. [38] showed that ECP is in FPT with respect to this parameter by giving an algorithm running in $k^{k^{k^{\mathcal{O}(k)}}} \cdot n^{\mathcal{O}(1)}$ time, where $k = \mathrm{vi}(G)$. We show that using an N-fold IP formulation, we can give a simpler algorithm with a doubly exponential improvement in the running time.

▶ **Theorem 7.** *The* Equitable Connected Partition *problem is fixed-parameter tractable parameterised by the vertex-integrity* vi *and can be solved in $k^{\mathcal{O}(k^4)} \cdot n \log n$ time, where $k = \mathrm{vi}(G)$.*

Proof. First, if it holds that $p > k$, we use the algorithm of Gima et al. [38] which, in this special case, runs in time $k^{\mathcal{O}(k^2)} \cdot n$. Therefore, the bottleneck of their approach is clearly the case when $p \leq k$. In what follows, we introduce our own procedure for this case, which is based on the N-fold integer programming. Note that the algorithm of Gima et al. [38] for the case $p \leq k$ is based on the algorithm of Lenstra [49].

First, we guess (by guessing we mean exhaustively trying all possibilities) a partition of the modulator vertices X in the solution. Let this solution partition be $X_1, \ldots, X_p$. Furthermore, we guess which (missing) connections between the vertices in the modulator will be realised through the components of $G - X$. Let $E(X)$ be the set of these guessed connections.

Now, we check the validity of our guess using the (configuration) N-fold ILP. Each component of $G - X$ (call them *pieces*) has at most k vertices; therefore, it can be split in at most k chunks (not necessarily connected) that will be attached to some modulator vertices already assigned to the parts of the solution. Let $\mathcal{P}(G, X)$ be the set of all pieces of $G - X$. Now, we want to verify if there exists a selection of chunks for every piece so that when we collect these together the solution is indeed connected and contains the right number of vertices. Thus, there are altogether at most k^k configurations of chunks in a piece. Let $\mathcal{C}(Z)$ be the set of all configurations of a piece Z. Let $s^Z_{C,i}$ be the number of vertices in the chunk attached to the i-th part from a piece Z in the configuration C.Let Z be a piece and $C \in \mathcal{C}(Z)$, we set $e^Z_C(u, v) = 1$ if the chunk assigned by C to the part containing both $u, v \in X$ connects u and v.

Now, we have to ensure (local constraint) that each piece is in exactly one configuration

$$\sum_{C \in \mathcal{C}(Z)} x^Z_C = 1 \qquad \forall Z \in \mathcal{P}(G, X)\,. \tag{4}$$

Observe that these constraints have no variables in common for two distinct elements of $\mathcal{P}(G, X)$. The rest of the necessary computation uses global constraints. We ensure that the total contribution of chunks assigned to the parts is the correct number (x_i is a binary slack such that $\sum_i x_i = n \bmod p$):

$$x_i + \sum_{Z \in \mathcal{P}(G,X)} \sum_{C \in \mathcal{C}(Z)} s^Z_{C,i} \cdot x^Z_C = \lceil n/p \rceil - |X_i| \qquad \forall i \in [p] \tag{5}$$

Next, we have to verify the connectivity of parts in X

$$\sum_{Z \in \mathcal{P}(G,X)} \sum_{C \in \mathcal{C}(Z)} e^Z_C(u,v) \cdot x^Z_C \geq 1 \qquad \forall \{u,v\} \in E(X) \tag{6}$$

It is not hard to verify, that the parameters of N-fold IP are as follows:

- the number s of local constraints in a brick is exactly 1 as there is a single local constraint (4) for each piece,
- the number r of global constraints is in $\mathcal{O}\left(k^2\right)$: there are $p \leq k$ constraints (5) and $\binom{k}{2}$ constraints (6),
- the number t of variables in a brick is $|\mathcal{C}(Z)|$ which is $k^{\mathcal{O}(k)}$, and
- $a \in \mathcal{O}\left(k^3\right)$, since all coefficients in the constraints are bounded by k in absolute value.

Thus, using Proposition 6, the Equitable Connected Partition problem can be solved in $k^{\mathcal{O}\left(k^4\right)} \cdot n \log n$ time. ◀

Using techniques from the proof of Theorem 7, we may give a specialised algorithm for Equitable Connected Partition parameterised by the 3-path vertex cover. The core idea is essentially the same, but the components that remain after removing the modulator are much simpler: they are either isolated vertices or isolated edges. This fact allows us to additionally speed the algorithm up.

▶ **Theorem 8.** *The Equitable Connected Partition problem is fixed-parameter tractable parameterised by the 3-path vertex cover number and can be solved in $k^{\mathcal{O}\left(k^2\right)} \cdot n \log n$ time, where $k = 3\text{-pvcn}(G)$.*

The next result is an XP algorithm with respect to the tree-width of G. It should be noted that a similar result was reported already by Enciso et al. [26]; however, their proof was never published[1] and the only clue for the algorithm the authors give in [26] is that this algorithm "can be proved using standard techniques for problems on graphs of bounded treewidth". Therefore, to fill this gap in the literature, we give our own algorithm.

▶ **Theorem 9.** *The Equitable Connected Partition problem is in XP when parameterized by the tree-width tw of G.*

Proof sketch. The algorithm is, as is usual, a leaf-to-root dynamic programming algorithm along a nice tree decomposition. The crucial observation we need for the algorithm is that at every moment of the computation, there are at most $\mathcal{O}(\text{tw})$ opened parts. This holds because each bag forms a separator in G and therefore no edge can "circumvent" currently processed bag.

The algorithm then proceeds as follows. In each node x of the tree-decomposition, we remember all possible partitions of vertices into open parts and the sizes of each open cluster including already forgotten vertices. We require that each open part is connected. Once a new vertex v is introduced, we have three possibilities: create a new part consisting of only

[1] In particular, in their conference version, Enciso et al. [26] promised to include the proof in an extended version, which, however, has never been published. There is also a version containing the appendix of the conference paper available from `https://www.researchgate.net/publication/220992885_What_Makes_Equitable_Connected_Partition_Easy`; however, even in this version, the proof is not provided.

v, put v into an existing opened part, or merge (via v) multiple already existing parts into a new one. When a vertex v is forgotten we need to check whether v is the last vertex of its part and, if yes, whether the part v is member of is of size $\lfloor n/p \rfloor$ or $\lceil n/p \rceil$. In join nodes, we just merge two records from child nodes with the same partition of bag vertices, or we merge different partitions whose connectivity is secured by the past vertices.

Once the dynamic table is correctly filled, we ask whether the dynamic table for the root of the tree-decomposition stores `true` in its single cell. The size of each dynamic programming table is $\mathrm{tw}^{\mathcal{O}(\mathrm{tw})} \cdot n^{\mathcal{O}(\mathrm{tw})} = n^{\mathcal{O}(\mathrm{tw})}$, and each table can be computed in the same time. Therefore, the algorithm runs in $\mathcal{O}(n \cdot \mathrm{tw}) \cdot n^{\mathcal{O}(\mathrm{tw})}$ time, which is in XP. ◀

Observe that if the size of every part is bounded by a parameter ς, the size of each dynamic programming table is $\mathrm{tw}^{\mathcal{O}(\mathrm{tw})} \cdot \varsigma^{\mathcal{O}(\mathrm{tw})}$ and we need the same time to compute each cell. Therefore, the algorithm also shows the following tractability result.

▶ **Corollary 10.** *The* Equitable Connected Partition *problem is fixed-parameter tractable parameterised by the tree-width* tw *and the size of a large part* $\varsigma = \lceil n/p \rceil$ *combined.*

In other words, the Equitable Connected Partition problem becomes tractable if the tree-width is bounded and the number of parts is large.

So far, we investigated the complexity of the problem mostly with respect to structural parameters that are bounded for sparse graphs. Now, we turn our attention to parameters that our bounded for dense graphs. Note that such parameters are indeed interesting for the problem, as the problem becomes polynomial-time solvable on cliques. We were not able to find this result in the literature, and, therefore, we present it in its entirety.

▶ **Observation 11.** *The* Equitable Connected Partition *problem can be solved in linear time if the graph* G *is a clique.*

Proof. First, we determine the number of parts of size $\lceil n/p \rceil$ as $\ell = (n \bmod p)$, and the number of smaller parts of size $\lfloor n/p \rfloor$ as $s = p - \ell$. Now, we arbitrarily assign vertices to p parts such that the first ℓ parts contain $\lceil n/p \rceil$ vertices and the remaining s parts contain exactly $\lfloor n/p \rfloor$ vertices. This, in fact, creates an equitable partition. Moreover, every partition is connected, since each pair of vertices is connected by an edge in G. ◀

Following the usual approach of distance from triviality [1, 41], we study the problem of our interest with respect to the distance to clique. We obtain the following tractability result.

▶ **Theorem 12.** *The* Equitable Connected Partition *problem is fixed-parameter tractable when parameterised by the distance to clique* k.

Next, we prove polynomial-time solvability for a more general class of graphs than cliques. Namely, we provide a tractable algorithm for co-graphs.

▶ **Theorem 13.** *The* Equitable Connected Partition *problem can be solved in polynomial time if the graph* G *is a co-graph.*

Next structural parameter we study is the neighbourhood diversity, which is generalisation of the famous vertex cover number that, in contrast, allows for large cliques to be present in G. Later on, we will also provide a fixed-parameter tractable algorithm for a more general parameter called modular-width; however, the algorithm for neighbourhood diversity will serve as a building block for the later algorithm, and therefore we find it useful to present the algorithm in its entirety.

▶ **Theorem 14.** *The* Equitable Connected Partition *problem is fixed-parameter tractable parameterised by the neighbourhood diversity* $\mathrm{nd}(G)$*.*

Proof. We first observe that each connected sub-graph of G "induces" a connected graph of the type-graph of G. More precisely, each connected sub-graph of G is composed of vertices that belong to types of G that induce a connected sub-graph of the type-graph. Therefore, the solution is composed of various realisations of connected sub-graphs of the type-graph of G. Note that there are at most $2^{\mathrm{nd}(G)}$ connected sub-graphs of the type-graph of G. We will resolve this task using an ILP using integral variables x_H^t for a type t and a connected sub-graph H of the type-graph of G. Furthermore, we have additional variables x_H. That is, the total number of variables is (upper-bounded by) $\mathrm{nd}(G) \cdot 2^{\mathrm{nd}(G)} + 2^{\mathrm{nd}(G)}$. The meaning of a variable x_H^t is "how many vertices of type t we use in a realisation of H". The meaning of a variable x_H is "how many realisations of H there are in the solution we find". We write $t \in H$ for a type that belongs to H (a connected sub-graph of the type-graph). Let σ denote the lower-bound on the size of parts of a solution, that is, $\sigma = \lfloor n/k \rfloor$. In order for this to hold we add the following set of constraints (here, $\xi_G = 0$ if $n = k \cdot \sigma$ and $\xi_G = 1$, otherwise):

$$\sigma x_H \le \sum_{t \in H} x_H^t \le (\sigma + \xi_G) x_H \qquad \forall H \tag{7}$$

$$x_H \le x_H^t \qquad \forall H\, \forall t \in H \tag{8}$$

$$\sum_{H} x_H^t = n_t \qquad \forall t \in T(G) \tag{9}$$

$$0 \le x_H^t \le n_t,\ x_H^t \in \mathbb{Z} \qquad \forall H\, \forall t \in H \tag{10}$$

$$0 \le x_H,\ x_H^t \in \mathbb{Z} \qquad \forall H \tag{11}$$

That is, (9) ensures that we place each vertex to some sub-graph H. The set of conditions (7) ensures that the total number of vertices assigned to the pattern H is divisible into parts of sizes σ or $\sigma + 1$. The set of conditions (8) ensures that each type that participates in a realisation of H contains at least x_H vertices, that is, we can assume that each realisation contains at least one vertex of each of its types. It is not difficult to verify that any solution to the Equitable Connected Partition problem fulfils (7)–(11).

In the opposite direction, suppose that we have an integral solution x satisfying (7)–(11). Let $\mathcal{H}$ be the collection of graphs H with multiplicities corresponding to x, that is, a graph H belongs to $\mathcal{H}$ exactly x_H-times. First, we observe that $|\mathcal{H}| = k$. To see this note that

$$|G| = \sum_{t \in H} n_t = \sum_{t \in H} \sum_{H \in \mathcal{H}} x_H^t = \sum_{H \in \mathcal{H}} \sum_{t \in H} x_H^t \ge \sum_{H \in \mathcal{H}} \sigma \cdot 1 \ge \sigma \sum_{H} x_H = \sigma |\mathcal{H}| \,.$$

Similarly, we have $|G| \le (\sigma + \xi_G)|\mathcal{H}|$ and the claim follows. Now, we find a realisation for every $H \in \mathcal{H}$. We know that there are $x_H^t \ge \sigma x_H$ vertices allocated to H. We assign them to the copies of H in $\mathcal{H}$ as follows. First, from each type $t \in H$ we assign one vertex to each copy of H (note that this is possible due to (8)). We assign the rest of the vertices greedily, so that there are σ vertices assigned to each copy of H; then, we assign the leftover vertices (note that there are at most x_H of them in total) to the different copies of H. In this way, we have assigned all vertices and gave a realisation of $\mathcal{H}$.

As was stated before, the integer linear program has only parameter-many variables. Hence, we can use the algorithm of Lenstra [49] to solve it in FPT time. ◀

With the algorithm from the proof of Theorem 14 in hand, we are ready to derive the result also for the modular-width.

▶ **Theorem 15.** *The* Equitable Connected Partition *problem is fixed-parameter tractable parameterised by the modular-width* $\mathrm{mw}(G)$*.*

Proof sketch. Clearly, the leaf-nodes of the modular decomposition of G have bounded neighbourhood diversity. For the graph of a leaf-node, we employ the following ILP:

$$\sigma x_H \le \sum_{t \in H} x_H^t \le (\sigma + \xi_G) x_H \qquad \forall H \qquad (12)$$

$$x_H \le x_H^t \qquad \forall H \, \forall t \in H \qquad (13)$$

$$\sum_H x_H^t \le n_t \qquad \forall t \in T(G) \qquad (14)$$

$$0 \le x_H^t \le n_t,\ x_H^t \in \mathbb{Z} \qquad \forall H \, \forall t \in H \qquad (15)$$

$$0 \le x_H,\ x_H^t \in \mathbb{Z} \qquad \forall H \qquad (16)$$

Note the difference between (14) and (9); that is, this time we do not insist on assigning all vertices and can have some leftover vertices. We add an objective function

$$\max \sum_H \sum_{t \in H} x_H^t \, ,$$

that is, we want to cover as many vertices as possible already in the corresponding leaf-node. Next, we observe that based on the solution of the above ILP, we can replace the leaf-node with a graph of neighbourhood diversity 2. In order to do so, we claim that if we replace the graph represented by the leaf-node by a disjoint union of a clique of size $\sum_H \sum_{t \in H} x_H^t$ and an independent set of size $\sum_H (n_t - \sum_{t \in H} x_H^t)$, then we do not change the answer to the Equitable Connected Partition problem. That is, the answer to the original graph was *yes* if and only if the answer is *yes* after we alter the leaf-node. The algorithm for modular-width then follows by a repeated application of the above ILP. ◀

As the last result of this section, we give an XP algorithm for another structural parameter called distance to cluster graph. The algorithm, in its core, is based on the same ideas as our FPT algorithm for distance to clique. Nevertheless, the number of types of vertices is no longer bounded only by a function of a parameter, and to partition the vertices that are not in the neighbourhood of the modulator vertices, we need to employ dynamic programming.

▶ **Theorem 16.** *The* Equitable Connected Partition *problem is* XP *parameterised by the distance to cluster graph* $\mathrm{dcg}(G)$*.*

4 Hardness Results

In this section, we complement our algorithmic upper-bounds from the previous section with matching hardness lower-bounds. The results from this section clearly show that no XP algorithm introduced in this paper can be improved to a fixed-parameter tractable one, or pushed to a more general parameter.

First, we observe that ECP is W[1]-hard with respect to the feedback-edge set number fes of G. This negatively resolves the question from the introduction of our paper. In fact, the actual statement shows an even stronger intractability result.

▶ **Theorem 17.** *The* Equitable Connected Partition *problem is* W[1]*-hard with respect to the path-width* $\mathrm{pw}(G)$*, the feedback-edge set number* $\mathrm{fes}(G)$*, and the number of parts* p *combined.*

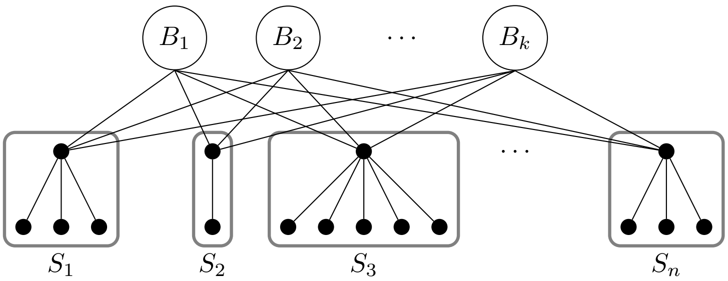

Figure 2 An illustration of the construction used to prove Theorem 19.

Note that the result from Theorem 17 can be, following the same arguments as used by [26], strengthened to show the same result for planar graphs.

▶ **Corollary 18.** *The* Equitable Connected Partition *problem is* W[1]*-hard when parameterised by the path-width, the feedback-edge set, and the number of parts combined, even if G is a planar graph.*

Next, we show that the d-pvcn parameter from Theorem 8 cannot be relaxed any more while the problem is kept tractable. Our reduction[2] is even more general and comes from the Unary Bin Packing problem, which is defined as follows.

Unary Bin Packing

Input: A number of bins k, a capacity of a single bin b, and a multi-set of integers $A = \{a_1, \ldots, a_n\}$ such that $\sum_{a \in A} a = bk$.

Question: Is there a surjective mapping $\alpha\colon A \to [k]$ such that for every $i \in [k]$ we have $\sum_{a \in \alpha^{-1}(i)} a = b$?

The Unary Bin Packing problem is well-known to be W[1]-hard when parameterised by the number of bins k and not solvable in $f(k) \cdot n^{o(k/\log k)}$ time for any computable function f, even if all numbers are given in unary [44].

▶ **Theorem 19.** *For every graph family $\mathcal{G}$ such that it contains at least one connected graph G with s vertices for every $s \in \mathbb{N}$, the* Equitable Connected Partition *problem is* W[1]*-hard parameterised by the distance to $\mathcal{G}$ referred to as $\operatorname{dist}_{\mathcal{G}}(G)$ and the number of parts p combined and, unless ETH fails, there is no algorithm running in $f(\ell) \cdot n^{o(\ell/\log \ell)}$ time for any computable function f, where $\ell = p + \operatorname{dist}_{\mathcal{G}}(G)$.*

Proof sketch. Let $\mathcal{I} = (A, k, b)$ be an instance of the Unary Bin Packing problem. We construct an equivalent instance $\mathcal{J} = (G, p)$ of the Equitable Connected Partition problem as follows (see Figure 2 for an overview of the construction). For the sake of exposition, we assume that $\mathcal{G}$ is a family containing all disjoint unions of stars; later we show how to tweak the construction to work with any $\mathcal{G}$ satisfying the conditions from the theorem statement.

For every number $a_i \in A$, we create a single *item-gadget* S_i which is a star with a_i vertices. Every S_i will be connected with the rest of the graph G only via the star centre c_i; we call this special vertex a *hub*. Next, we create k *bin-gadgets* $B_1, \ldots, B_k$. Each of these gadgets

[2] We would like to mention here that the construction used in the proof of Theorem 19 starts with the same problem and share similarities with the independent hardness construction of Gima and Otachi [39]; however, our construction is arguably easier and prove much more general hardness results.

consists of a single vertex. Slightly abusing the notation, we call this vertex also B_i, $i \in [k]$. As the last step of the construction, we add an edge connecting every bin-gadget with every item-gadget and set $p = k$. ◀

It is not hard to see that every graph G which is a disjoint union of stars has constant 4-path vertex cover number – 0, to be precise. Therefore, by Theorem 19 we obtain the desired hardness result.

▶ **Corollary 20.** *The* Equitable Connected Partition *problem is* W[1]*-hard parameterised by the* 4*-path vertex cover number* 4-pvcn *and the number of parts* p *combined.*

Now, a previous blind spot of our understanding of the Equitable Connected Partition problem's complexity with respect to the structural parameters that are bounded mostly for sparse graphs is the distance to disjoint paths. We again obtain hardness as a direct corollary of Theorem 19.

▶ **Corollary 21.** *The* Equitable Connected Partition *problem is* W[1]*-hard parameterised by the distance to disjoint paths* $\mathrm{ddp}(G)$ *and the number of parts* p *combined.*

Enciso et al. [26] stated (and we formalized in Theorem 9) that there is an XP algorithm for the Equitable Connected Partition problem parameterised by the tree-width of G. A natural question is then whether this algorithm can be improved to solve the problem in the same running-time also with respect to the more general parameter called clique-width. We give a strong evidence that such an algorithm is unlikely in the following theorem.

▶ **Theorem 22.** *The* Equitable Connected Partition *problem is* NP*-hard even if the graph* G *has clique-width* 3*, and is solvable in polynomial-time on graphs of clique-width at most* 2*.*

Proof. To show the hardness, we reuse the reduction used to prove Theorem 19. Recall that the construction can be done in polynomial time, thus, the reduction is also a polynomial reduction. What we need to show is that the clique-width of the constructed graph G is constant. We show this by providing an algebraic expression that uses 3 labels. First, we create a graph G_1 containing all bin-gadgets. This can be done by introducing a single vertex and by repeating disjoint union operation. We additionally assume that all vertices in G_1 have label 3. Next, we create a graph G_2 containing all item-gadgets. Every item-gadget is a star which can be constructed using two labels 1 and 2. Without loss of generality, we assume that all item-gadgets' centres have label 1 and all leaves have label 2. To complete the construction, we create disjoint union of G_1 and G_2 and, then, we perform a full join of vertices labelled 1 and 3. It is easy to see that the expression indeed leads to a desired graph. Polynomial-time solvability follows from Theorem 13 and the fact that co-graphs are exactly the graphs with clique-width at most 2 [17]. ◀

Using similar arguments, we can show para-NP-hardness also for a more restrictive parameter called shrub-depth [32, 34, 33].

▶ **Theorem 23.** *The* Equitable Connected Partition *problem is* NP*-hard even if the graph* G *has shrub-depth* 3*.*

As the last piece of the complexity picture of the Equitable Connected Partition problem, we show that ECP is W[1]-hard with respect to the distance to disjoint cliques. Recall that we give an XP algorithm for this parameter in Theorem 16.

▶ **Corollary 24.** *The* Equitable Connected Partition *problem is* W[1]*-hard with respect to the distance to cluster graph* $\mathrm{dcg}(G)$ *and the number of parts* p *combined.*

5 Conclusions

We revisit the complexity picture of the Equitable Connected Partition problem with respect to various structural restrictions of the graph. We complement the existing results with algorithmic upper-bounds and corresponding complexity lower-bounds that clearly show that no existing parameterised algorithm can be significantly improved.

Despite that the provided complexity study gives us a clear dichotomy between tractable and intractable cases, there still remain a few blind spots. One of the most interesting is the complexity classification of the Equitable Connected Partition problem with respect to the band-width parameter, which lies between the maximum leaf number and the path-width; however, is incomparable with the feedback-edge set number.

An interesting line of research can target the tightness of our results. For example, we show a clear dichotomy between the tractable and intractable cases of ECP when parameterised by the clique-width. Using similar arguments, we can show the following result for the recently introduced parameter twin-width [11].

▶ **Theorem 25.** *The Equitable Connected Partition problem is* NP*-hard even if the graph G has twin-width* 2*, and is solvable in polynomial-time on graphs of twin-width* 0*.*

We conjecture that the problem is polynomial-time solvable also on graphs of twin-width 1, which, unlike the twin-width 2 graphs, are additionally known to be recognisable efficiently [10]. Similarly, we can ask whether the provided FPT algorithms are optimal under some standard theoretical assumptions, such as the well-known Exponential-Time Hypothesis [42].

Last but not least, the parameterised complexity framework not only gives us formal tools for finer-grained complexity analysis of algorithms for NP-hard problems, but, at the same time, equips us also with the necessary formalism for analysis of effective preprocessing, which is widely known as *kernelisation*. A natural follow-up question is then whether the Equitable Connected Partition problem admits a polynomial kernel with respect to any of the studied structural parameters. We conjecture that there is a polynomial kernel with respect to the distance to clique and 3-path vertex cover number.

References

1 Akanksha Agrawal and M. S. Ramanujan. Distance from triviality 2.0: Hybrid parameterizations. In Cristina Bazgan and Henning Fernau, editors, *Proceedings of the 33rd International Workshop on Combinatorial Algorithms, IWOCA '22*, volume 13270 of *Lecture Notes in Computer Science*, pages 3–20. Springer, 2022. doi:10.1007/978-3-031-06678-8_1.

2 Micah Altman. The computational complexity of automated redistricting: Is automation the answer? *Rutgers Computer and Law Technology Journal*, 23(1):81–142, March 1997.

3 Kateřina Altmanová, Dušan Knop, and Martin Koutecký. Evaluating and tuning n-fold integer programming. *ACM Journal of Experimental Algorithmics*, 24(2), July 2019. doi:10.1145/3330137.

4 Konstantin Andreev and Harald Racke. Balanced graph partitioning. *Theory of Computing Systems*, 39(6):929–939, November 2006. doi:10.1007/s00224-006-1350-7.

5 Peter Arbenz, G. Harry van Lenthe, Uche Mennel, Ralph Müller, and Marzio Sala. Multi-level μ-finite element analysis for human bone structures. In Bo Kågström, Erik Elmroth, Jack Dongarra, and Jerzy Waśniewski, editors, *Proceedings of the 8th International Workshop on Applied Parallel Computing, PARA '06*, volume 4699 of *Lecture Notes in Computer Science*, pages 240–250. Springer, 2007. doi:10.1007/978-3-540-75755-9_30.

6 René van Bevern, Andreas Emil Feldmann, Manuel Sorge, and Ondřej Suchý. On the parameterized complexity of computing balanced partitions in graphs. *Theory of Computing Systems*, 57:1–35, July 2015. doi:10.1007/s00224-014-9557-5.

7 Sandeep N. Bhatt and Frank Thomson Leighton. A framework for solving VLSI graph layout problems. *Journal of Computer and System Sciences*, 28(2):300–343, April 1984. doi:10.1016/0022-0000(84)90071-0.

8 Václav Blažej, Robert Ganian, Dušan Knop, Jan Pokorný, Šimon Schierreich, and Kirill Simonov. The parameterized complexity of network microaggregation. In *Proceedings of the 37th AAAI Conference on Artificial Intelligence, AAAI '23*, pages 6262–6270. AAAI Press, 2023. doi:10.1609/aaai.v37i5.25771.

9 Hans L. Bodlaender and Fedor V. Fomin. Equitable colorings of bounded treewidth graphs. *Theoretical Computer Science*, 349(1):22–30, December 2005. doi:10.1016/j.tcs.2005.09.027.

10 Édouard Bonnet, Eun Jung Kim, Amadeus Reinald, Stéphan Thomassé, and Rémi Watrigant. Twin-width and polynomial kernels. *Algorithmica*, 84(11):3300–3337, 2022. doi:10.1007/S00453-022-00965-5.

11 Édouard Bonnet, Eun Jung Kim, Stéphan Thomassé, and Rémi Watrigant. Twin-width I: Tractable FO model checking. *Journal of the ACM*, 69(1), 2022. doi:10.1145/3486655.

12 Robert Bredereck, Andrzej Kaczmarczyk, Dušan Knop, and Rolf Niedermeier. High-multiplicity fair allocation: Lenstra empowered by N-fold integer programming. In Anna Karlin, Nicole Immorlica, and Ramesh Johari, editors, *Proceedings of the 20th ACM Conference on Economics and Computation, EC '19*, pages 505–523. ACM, 2019. doi:10.1145/3328526.3329649.

13 Thang Nguyen Bui and Andrew Peck. Partitioning planar graphs. *SIAM Journal on Computing*, 21(2):203–215, 1992. doi:10.1137/0221016.

14 Aydın Buluç, Henning Meyerhenke, Ilya Safro, Peter Sanders, and Christian Schulz. Recent advances in graph partitioning. In Lasse Kliemann and Peter Sanders, editors, *Algorithm Engineering: Selected Results and Surveys*, pages 117–158. Springer, 2016. doi:10.1007/978-3-319-49487-6_4.

15 Bor-Liang Chen, Ming-Tat Ko, and Ko-Wei Lih. Equitable and m-bounded coloring of split graphs. In Michel Deza, Reinhardt Euler, and Ioannis Manoussakis, editors, *Combinatorics and Computer Science, CCS '95*, volume 1120 of *Lecture Notes in Computer Science*, pages 1–5. Springer, 1996.

16 Bor-Liang Chen and Ko-Wei Lih. Equitable coloring of trees. *Journal of Combinatorial Theory, Series B*, 61(1):83–87, May 1994.

17 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1):77–114, 2000. doi:10.1016/S0166-218X(99)00184-5.

18 Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, Cham, 2015. doi:10.1007/978-3-319-21275-3.

19 Marek Cygan, Daniel Lokshtanov, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. Minimum bisection is fixed-parameter tractable. *SIAM Journal on Computing*, 48(2):417–450, 2019. doi:10.1137/140988553.

20 Argyrios Deligkas, Eduard Eiben, Robert Ganian, Thekla Hamm, and Sebastian Ordyniak. The parameterized complexity of connected fair division. In Zhi-Hua Zhou, editor, *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI '21*, pages 139–145. International Joint Conferences on Artificial Intelligence Organization, August 2021. Main Track. doi:10.24963/ijcai.2021/20.

21 Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics. Springer, Berlin, Heidelberg, 5th edition, 2017. doi:10.1007/978-3-662-53622-3.

22 Martin Doucha and Jan Kratochvíl. Cluster vertex deletion: A parameterization between vertex cover and clique-width. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science, MFCS '12*, volume 7464 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2012. doi:10.1007/978-3-642-32589-2_32.

23 M. E. Dyer and A. M. Frieze. A partitioning algorithm for minimum weighted Euclidean matching. *Information Processing Letters*, 18(2):59–62, 1984. doi:10.1016/0020-0190(84)90024-3.

24 Josep Díaz and George B. Mertzios. Minimum bisection is NP-hard on unit disk graphs. *Information and Computation*, 256:83–92, 2017. doi:10.1016/j.ic.2017.04.010.

25 Friedrich Eisenbrand, Christoph Hunkenschröder, Kim-Manuel Klein, Martin Koutecký, Asaf Levin, and Shmuel Onn. An algorithmic theory of integer programming. *CoRR*, abs/1904.01361, 2019. arXiv:1904.01361.

26 Rosa Enciso, Michael R. Fellows, Jiong Guo, Iyad Kanj, Frances Rosamond, and Ondřej Suchý. What makes equitable connected partition easy. In Jianer Chen and Fedor V. Fomin, editors, *Proceedings of the 4th International Workshop on Parameterized and Exact Computation, IWPEC '09*, volume 5917 of *Lecture Notes in Computer Science*, pages 122–133. Springer, 2009.

27 Andreas Emil Feldmann and Luca Foschini. Balanced partitions of trees and applications. *Algorithmica*, 71(2):354–376, February 2015. doi:10.1007/s00453-013-9802-3.

28 Michael R. Fellows, Fedor V. Fomin, Daniel Lokshtanov, Frances Rosamond, Saket Saurabh, Stefan Szeider, and Carsten Thomassen. On the complexity of some colorful problems parameterized by treewidth. *Information and Computation*, 209(2):143–153, 2011. doi:10.1016/j.ic.2010.11.026.

29 Jiří Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science*, 412(23):2513–2523, 2011. doi:10.1016/j.tcs.2010.10.043.

30 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM Journal on Computing*, 43(5):1541–1563, 2014. doi:10.1137/130910932.

31 H. Furmańczyk and M. Kubale. The complexity of equitable vertex coloring of graphs. *Journal of Applied Computer Science*, 13(2):95–106, 2005.

32 Jakub Gajarský, Michael Lampis, and Sebastian Ordyniak. Parameterized algorithms for modular-width. In Gregory Gutin and Stefan Szeider, editors, *Proceedings of the 8th International Symposium on Parameterized and Exact Computation, IPEC '13*, volume 8246 of *Lecture Notes in Computer Science*, pages 163–176. Springer, 2013.

33 Robert Ganian, Petr Hliněný, Jaroslav Nešetřil, Jan Obdržálek, and Patrice Ossona de Mendez. Shrub-depth: Capturing height of dense graphs. *Logical Methods in Computer Science*, 15(1), 2019. doi:10.23638/LMCS-15(1:7)2019.

34 Robert Ganian, Petr Hliněný, Jaroslav Nešetřil, Jan Obdržálek, Patrice Ossona de Mendez, and Reshma Ramadurai. When trees grow low: Shrubs and fast MSO1. In Branislav Rovan, Vladimiro Sassone, and Peter Widmayer, editors, *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science, MFCS '12*, volume 7464 of *Lecture Notes in Computer Science*, pages 419–430. Springer, 2012. doi:10.1007/978-3-642-32589-2_38.

35 Robert Ganian and Jan Obdržálek. Expanding the expressive power of monadic second-order logic on restricted graph classes. In Thierry Lecroq and Laurent Mouchard, editors, *Proceedings of the 24th International Workshop on Combinatorial Algorithms, IWOCA '13*, volume 8288 of *Lecture Notes in Computer Science*, pages 164–177. Springer, 2013. doi:10.1007/978-3-642-45278-9_15.

36 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

37 Tomáš Gavenčiak, Martin Koutecký, and Dušan Knop. Integer programming in parameterized complexity: Five miniatures. *Discrete Optimization*, 44:100596, 2022. Optimization and Discrete Geometry. doi:10.1016/j.disopt.2020.100596.

38 Tatsuya Gima, Tesshu Hanaka, Masashi Kiyomi, Yasuaki Kobayashi, and Yota Otachi. Exploring the gap between treedepth and vertex cover through vertex integrity. *Theoretical Computer Science*, 918:60–76, 2022. doi:10.1016/j.tcs.2022.03.021.

39 Tatsuya Gima and Yota Otachi. Extended MSO model checking via small vertex integrity. *Algorithmica*, 86(1):147–170, 2024. doi:10.1007/S00453-023-01161-9.

40 Guilherme C. M. Gomes, Matheus R. Guedes, and Vinicius F. dos Santos. Structural parameterizations for equitable coloring: Complexity, FPT algorithms, and kernelization. *Algorithmica*, 85:1912–1947, July 2023. doi:10.1007/s00453-022-01085-w.

41 Jiong Guo, Falk Hüffner, and Rolf Niedermeier. A structural view on parameterizing problems: Distance from triviality. In Rod Downey, Michael Fellows, and Frank Dehne, editors, *Proceedings of the 1st International Workshop on Parameterized and Exact Computation, IWPEC '04*, pages 162–173. Springer, 2004.

42 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

43 Takehiro Ito, Xiao Zhou, and Takao Nishizeki. Partitioning a graph of bounded tree-width to connected subgraphs of almost uniform size. *Journal of Discrete Algorithms*, 4(1):142–154, 2006. doi:10.1016/j.jda.2005.01.005.

44 Klaus Jansen, Stefan Kratsch, Dániel Marx, and Ildikó Schlotter. Bin packing with fixed number of bins revisited. *Journal of Computer and System Sciences*, 79(1):39–49, 2013. doi:10.1016/j.jcss.2012.04.004.

45 Dušan Knop, Martin Koutecký, Asaf Levin, Matthias Mnich, and Shmuel Onn. High-multiplicity N-fold IP via configuration LP. *Mathematical Programming*, 200(1):199–227, 2023. doi:10.1007/s10107-022-01882-9.

46 Dušan Knop, Martin Koutecký, and Matthias Mnich. Combinatorial N-fold integer programming and applications. *Mathematical Programming*, 184(1):1–34, 2020. doi:10.1007/s10107-019-01402-2.

47 Dušan Knop, Šimon Schierreich, and Ondřej Suchý. Balancing the spread of two opinions in sparse social networks (student abstract). In *Proceedings of the 36th AAAI Conference on Artificial Intelligence, AAAI '22*, pages 12987–12988. AAAI Press, 2022. doi:10.1609/aaai.v36i11.21630.

48 Michael Lampis. Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica*, 64(1):19–37, September 2012. doi:10.1007/s00453-011-9554-x.

49 Hendrik W. Lenstra, Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4), 1983. doi:10.1287/moor.8.4.538.

50 Harry A. Levin and Sorelle A. Friedler. Automated congressional redistricting. *ACM Journal of Experimental Algorithmics*, 24, April 2019. doi:10.1145/3316513.

51 Ko-Wei Lih. Equitable coloring of graphs. In Panos M. Pardalos, Ding-Zhu Du, and Ronald L. Graham, editors, *Handbook of Combinatorial Optimization*, pages 1199–1248. Springer, 2013. doi:10.1007/978-1-4419-7997-1_25.

52 Mario Lucertini, Yehoshua Perl, and Bruno Simeone. Most uniform path partitioning and its use in image processing. *Discrete Applied Mathematics*, 42(2):227–256, 1993. doi:10.1016/0166-218X(93)90048-S.

53 Kitty Meeks and Fiona Skerman. The parameterised complexity of computing the maximum modularity of a graph. *Algorithmica*, 82(8):2174–2199, August 2020. doi:10.1007/s00453-019-00649-7.

54 Manfred Wiegers. The k-section of treewidth restricted graphs. In Branislav Rovan, editor, *Proceedings of the 15th International Symposium on Mathematical Foundations of Computer Science, MFCS '90*, volume 452 of *Lecture Notes in Computer Science*, pages 530–537. Springer, 1990.

55 Justin C. Williams Jr. Political redistricting: A review. *Papers in Regional Science*, 74(1), 1995. doi:10.1111/j.1435-5597.1995.tb00626.x.