What Makes Equitable Connected Partition Easy

Rosa Enciso[1], Micheal R. Fellows[2], Jiong Guo[3], Iyad Kanj[4], Frances Rosamond[2], and Ondřej Suchý[5]*

[1] School of Electrical Engineering and Computer Science
University of Central Florida, Orlando, FL
renciso@cs.ucf.edu
[2] University of Newcastle, Newcastle, Australia
{michael.fellows|frances.rosamond}@newcastle.edu.au
[3] Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
jiong.guo@uni-jena.de
[4] School of Computing, DePaul University
243 S. Wabash Ave, Chicago, IL 60604
ikanj@cs.depaul.edu
[5] Department of Applied Mathematics and Institute for Theoretical Computer Science Charles University,
Malostranské nám. 25, 118 00 Praha, Czech Republic
suchy@kam.mff.cuni.cz

Abstract. We study the Equitable Connected Partition problem, which is the problem of partitioning a graph into a given number of partitions, such that each partition induces a connected subgraph, and the partitions differ in size by at most one. We examine the problem from the parameterized complexity perspective with respect to the number of partitions, the treewidth, the pathwidth, the size of a minimum feedback vertex set, the size of a minimum vertex cover, and the maximum number of leaves in a spanning tree of the graph. In particular, we show that the problem is W[1]-hard with respect to the first four parameters (even combined), whereas it becomes fixed-parameter tractable when parameterized by the last two parameters. The hardness result remains true even for planar graphs. We also show that the problem is in XP when parameterized by the treewidth (and hence any other mentioned structural parameter). Furthermore, we show that the closely related problem, Equitable Coloring, is FPT when parameterized by the maximum number of leaves in a spanning tree of the graph.

* Work partially supported by the ERASMUS program and by the DFG, project NI 369/4 (PIAF) while visiting Friedrich-Schiller-Universität Jena (October 2008–March 2009), by grant 201/05/H014 of the Czech Science Foundation and by grant 1M0021620808 of the Czech Ministry of Education.

1 Introduction

Let $G = (V, E)$ be an undirected graph. We say that $V_1, V_2, \ldots, V_r$ is *a partitioning of* V if and only if $\bigcup_{i=1}^{r} V_i = V$, and $\forall i, j, 1 \leq i < j \leq r : V_i \cap V_j = \emptyset$. A partitioning is *equitable* if $\forall i, j, 1 \leq i < j \leq r : ||V_i| - |V_j|| \leq 1$. In this paper we consider the following two NP-hard problems:

Equitable Connected Partition (ECP)
Instance: A simple graph $G = (V, E)$; a nonnegative integer r.
Question: Is there an equitable partitioning $V_1, V_2, \ldots, V_r$ of V such that each partition induces a connected subgraph (i.e., $\forall i, 1 \leq i \leq r : G[V_i]$ is connected)?

Equitable Coloring (EC)
Instance: A simple graph $G = (V, E)$; a nonnegative integer r.
Question: Is there an equitable partitioning $V_1, V_2, \ldots, V_r$ of V such that each partition induces an independent set?

We study the parameterized complexity of Equitable Connected Partition with respect to the following parameters:

- the treewidth of the input graph $tw(G)$,
- the pathwidth of the input graph $pw(G)$,
- the minimum size of a feedback vertex set in the input graph $fvs(G)$,
- the minimum size of a vertex cover in the input graph $vc(G)$,
- the maximum number of leaves in a spanning tree of the input graph $ml(G)$,
- the number of partitions r, and
- the above parameters combined.

We show that ECP is W[1]-hard when parameterized by $pw(G)$, $fvs(G)$, and the number of partitions r combined. We show that this result holds true even for planar graphs. On the positive side, we show that ECP becomes *fixed-parameter tractable* (FPT) when parameterized by $vc(G)$, or by $ml(G)$. We also show that ECP, parameterized by treewidth, is in XP.

For the Equitable Coloring problem, we show that the problem is FPT when parameterized by $ml(G)$.

Our results extend the recent results in [4, 6]. These results studied the parameterized complexity of some NP-hard labeling and coloring problems with respect to graph-structural parameters, such as the treewidth, the vertex cover number, and the feedback vertex set number. For example, it was shown in [4] that EC is W[1]-hard when parameterized by $tw(G)$ and r combined. More recently, it was shown in [6] that EC is FPT when parameterized by $vc(G)$.

This line of research is generally motivated by the following. Many W-hard problems (w.r.t. some parameter such as the solution size), are practically important. The parameterized complexity working hypothesis (FPT $\neq$ W[1]) implies that these problems are fixed-parameter intractable. Therefore, it is both interesting and important to study whether these problems become parameterized tractable when parameterized by other (tractable) structural parameters.

For the background and terminologies on graphs, we refer the reader to West [8], and for that on parameterized complexity, we refer the reader to Downey and Fellows' book [2].

In the rest of the paper, we will denote by $l := (n \mod r)$ the number of classes whose size is larger by one than the size of the other classes, i.e., we have $r - l$ classes of size $s := \lfloor n/r \rfloor$ and l classes of size $s + 1$.

2 Hardness results

This section is mainly devoted to proving the following theorem:

Theorem 1. Equitable Connected Partition *is W[1]-hard with respect to the pathwidth $pw(G)$, the minimum size of a feedback vertex set $fvs(G)$ and the number of partitons r combined.*

Proof. We will provide a parameterized reduction from the Multi-Colored Clique (MCC) problem, which was proven to be W[1]-complete by Fellows et al. [5]. In MCC we are given an undirected graph that is properly colored by k colors and the question is whether there is a size-k clique in this graph consisting of exactly one vertex from each color class. The parameter is k.

A basic building block of our construction is an *anchor*. It is a vertex (called *root*) with a lot of degree-one vertices pending on it (see Fig. 1). On one hand, as the prescribed class size will be much greater than one, the pending vertices must fall into the same class as the root. On the other hand, each anchor has too many pending vertices (the actual number will be provided later) that two anchors can never get into the same class. Finally we create exactly as many anchors as the number of classes; thus there is exactly one anchor in each class. Hence the situation can be viewed as if each class is started with one single anchor and then some more vertices are added. The number of vertices that need to be added to the class started by a particular anchor will differ for different anchors and is forced by the number of pending vertices that the anchor is missing for the prescribed class size. Anchors will be denoted by uppercase letters and by connecting something to an anchor we mean connecting it to the root of the anchor.

To interconnect the anchors we will use a building block called *a choice*. If $A = \{a_1, \ldots, a_t\}, 0 \leq a_1 < a_2 < a_3 < \ldots < a_t$ is a set of integers and $b \geq a_t$, then an (A, b)-*choice* is a path with $t+1$ vertices $v_1, \ldots, v_{t+1}$, where each vertex of the path can have some degree-one vertices pending on it (see Fig. 1). In particular, vertex v_1 has a_1 vertices pending on it, vertex v_{t+1} has $b - a_t$ pending vertices, and for all $i, 2 \leq i \leq t$, vertex v_i has $a_i - a_{i-1} - 1$ pending vertices. Observe again that the pending vertices must always fall into the same class as their unique neighbor.

Now if an anchor X is connected to an anchor Y by an (A, b)-choice (which is done by simply identifying vertex v_1 and v_{t+1} with the root of anchor X and Y, respectively), then there is an i, with $1 \leq i \leq t$, such that the vertices $v_1, \ldots, v_i$ (and their pending vertices) fall into the partition of X, while the

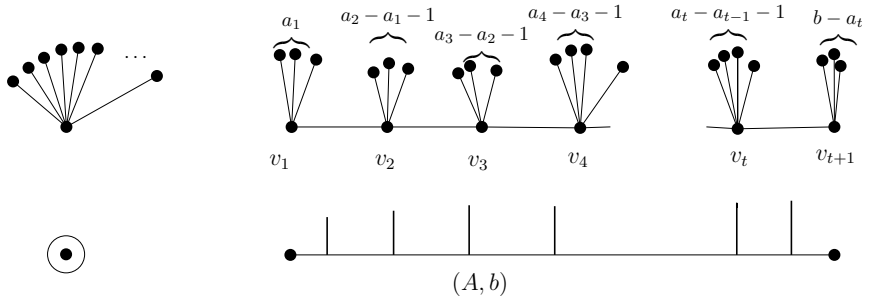

Fig. 1. Basic building blocks of our construction: Anchor (left), (A, b)-choice (right) and the way they are depicted in further figures (bellow).

vertices $v_{i+1}, \ldots, v_{t+1}$ fall into the partition of Y. Actually, the vertices v_1 and v_{t+1} were identified with the respective anchors, and hence we do not count them. Thus the number of vertices from this choice that fall into the partition of X is $a_1 + \sum_{j=2}^{i}((a_j - a_{j-1} - 1) + 1) = a_i \in A$, while the number of vertices falling into the partition of Y is $b - a_t + \sum_{j=i+1}^{t}((a_j - a_{j-1} - 1) + 1) = b - a_i$. Note that the vertices pending on v_1 and v_{t+1} are in fact pending on the roots of the appropriate anchors, but we still consider them as a part of this choice.

The construction is based on sending "signals" between anchors. Let A and B be two anchors connected by a choice. The vertices of the choice have to fall into the two partitions corresponding to A and B. The more vertices that fall into the partition of A the less vertices that will fall into the one of B. However, since the sizes of the two partitions differ by at most one, the partition of B must get some vertices somewhere else. This generates the signal. The choices allow us to control the signal sent.

Now suppose that $G = (V, E), k, c : V \to \{1, \ldots, k\}$ is an instance of MCC. Also suppose that, for each i, there are n_i vertices of color i denoted by $v_p^i, 1 \leq p \leq n_i$. Furthermore, we give each edge an integer ID, i.e., there is a bijective labeling $l : E \to \{1, \ldots, |E|\}$. Our construction has a selection gadget for each vertex color, which ensures both the selection of the vertex of this color and the selection of edges from the selected vertex to the vertices of the other colors. The selection gadgets are interconnected in a way that an equitable partitioning is only possible if the IDs of the edges selected match.

The selection gadget for color i is formed by $2(k-1)$ anchors $N_j^i, P_j^i, 1 \leq j \leq k, j \neq i$, connected into a cycle, each having a connection outside the gadget. The vertex selection is represented by a "big" signal that the outgoing connections are unable to handle, and hence is forced to run along the cycle without a change. The selection of an edge going from the selected vertex to the vertices of color j is then done between N_j^i and P_j^i via a "small" signal that equals the label of the selected edge. This signal is then sent to the anchors N_i^j, P_i^j of the selection gadget for color j, from anchor P_j^i to N_i^j and from N_j^i to P_i^j (in opposite directions).

Now we present the selection gadget more formally. First let $Z_0 := 2|E|+10$. The "big" signal is formed by the order of the vertex selected times the number Z_0, i.e., the possible signal states are $A_0^i := \{p \cdot Z_0 \mid 1 \leq p \leq n_i\}$. As we mentioned, the small signal is formed by the edge IDs. Between the anchors N_j^i and P_j^i both the big and the small signal is sent, and a particular small signal can only be used with an appropriate big signal. Thus, between N_j^i and P_j^i, the possible signal states are $A_j^i := \{p \cdot Z_0 + l(uv_p^i) \mid c(u) = j \text{ and } uv_p^i \in E\}$. To catch the order of the anchors along the cycle, we introduce the notion of a *successor*. For each $j, 1 \leq j \leq k$ set $succ(j) := j+1$ for $j \neq k$ and $j \neq (i-1)$, set $succ(k) := 1$ and $succ(i-1) := succ(i)$. Now for each $j, 1 \leq j \leq k, j \neq i$, the anchor P_j^i is connected to the anchor $N_{succ(j)}^i$ by an $(A_0^i, n_i \cdot Z_0)$-choice, the anchor N_j^i is connected to the anchor P_j^i by an $(A_j^i, n_i \cdot Z_0 + |E|)$-choice, the anchor P_j^i to the anchor N_i^j, and the anchor P_i^j to the anchor N_j^i by two $(\{1, \ldots, |E|\}, |E|)$-choices (see Fig. 2).

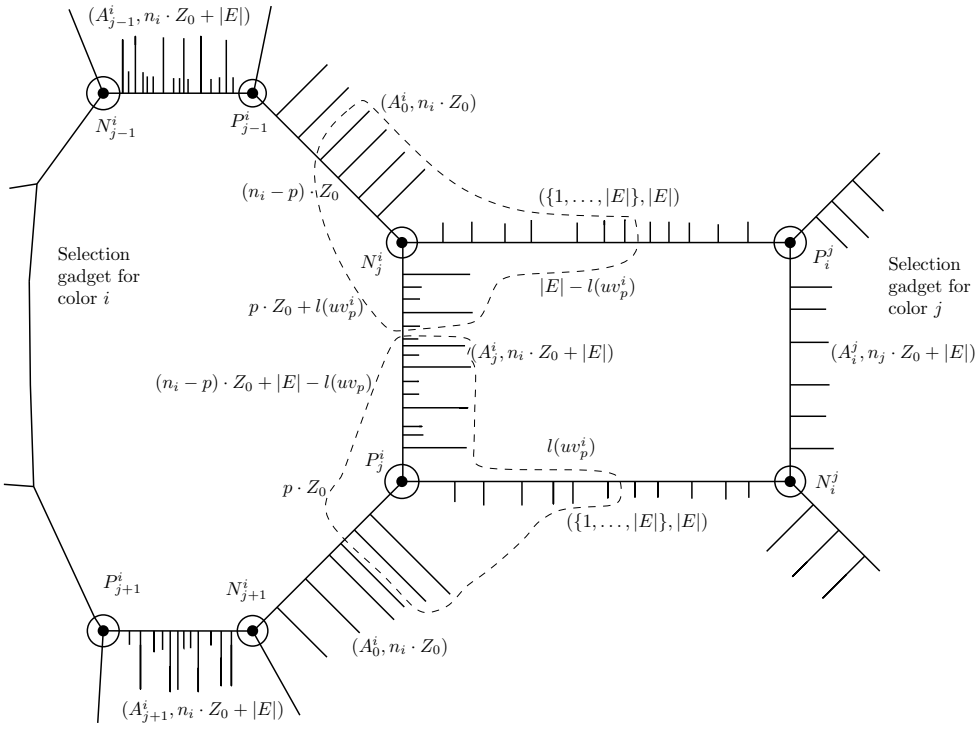

Fig. 2. A part of the selection gadget with possible partitioning shown by dashed line.

Now the class sizes are set such that each of the anchors in this selection gadget needs to get $n_i \cdot Z_0 + |E|$ vertices from the choices that it is incident with. Hence, if the anchor P_j^i takes $p \cdot Z_0$ vertices from the choice connecting it to the

anchor $N^i_{succ(j)}$, then the anchor $N^i_{succ(j)}$ gets $(n_i - p) \cdot Z_0$ vertices from this choice, and it must get $p \cdot Z_0 + |E|$ vertices from the two remaining choices that it is incident with. Since it can take at most $|E|$ vertices from the connection to the selection gadget for color $succ(j)$, it must take at least $p \cdot Z_0$ (but at most $p \cdot Z_0 + |E| < (p+1) \cdot Z_0$) vertices from the connection to $P^i_{succ(j)}$. Hence, it has to take $p \cdot Z_0 + l(uv^i_p)$ vertices for some $c(u) = succ(j)$ and $uv^i_p \in E$ from this connection and $|E| - l(uv^i_p)$ vertices from the connection to the other selection gadget. Hence $P^i_{succ(j)}$ gets $(n_i - p) \cdot Z_0 + |E| - l(uv_p)$ vertices from the connection from $N^i_{succ(j)}$, and by a similar reasoning it is forced to take $p \cdot Z_0$ vertices from the connection to $N^i_{succ(succ(j))}$ and $l(uv_p)$ vertices from the connection to the other selection gadget. Thus the anchor $N^i_{succ(succ(j))}$ is again forced to select some edge incident with vertex v^i_p, etc.

The anchor N^i_j is connected to anchor P^j_i by a $(\{1, \ldots, |E|\}, |E|)$-choice, and the number of vertices it takes into its partition out of this choice is $|E| - l(uv^i_p)$, where v^i_p is the vertex selected in the selection gadget for color i, $c(u) = j$ and $uv^i_p \in E$. The number of vertices that the anchor P^j_i takes out of this choice is $l(wv^j_q)$ where v^j_q is the vertex selected in the selection gadget for color j, $c(w) = i$ and $wv^j_q \in E$. Since the $|E|$ vertices of the choice must be partitioned into the partitions of its endpoints, it follows that $l(wv^j_q) = l(uv^i_p)$ and $wv^j_q = uv^i_p = v^j_q v^i_p$ is an edge of G. Hence the partitioning of the constructed graph is possible if and only if the selected vertices form a multi-colored clique in the graph G.

Now to determine the right size of the anchors, it is enough to ensure that each class is more than half full, once the starting anchor is added. The maximum demand (the number of vertices that should be added to its class except for itself and vertices pending on it) of any anchor is less than $n \cdot Z_0 + |E|$, hence it is enough to set the desired class size to $s := (2n+1) \cdot Z_0 = (2n+1) \cdot (2|E| + 10)$.

The number of partitions r is equal to the number of anchors. Since there are k selection gadgets, each containing $2(k-1)$ anchors, we have $2k(k-1)$ anchors in total. Now observe that if we delete all roots of the anchors, the resulting graph consists of paths with pending vertices. Hence, the roots form a feedback vertex set in the graph, and the pathwidth of the graph is also bounded by the number of roots plus one. The construction can be clearly carried out in polynomial time; in particular, the graph has $r \cdot s = O(k^2 \cdot n^3)$ vertices. □

Corollary 1. Equitable Connected Partition *is W[1]-hard for planar graphs with respect to the pathwidth $pw(G)$, the minimum size of a feedback vertex set $fvs(G)$ and the number of partitons r combined.*

Proof. The graph H' constructed in Theorem 1 is in general not planar. But there is a drawing of this graph such that only the edges of the choices connecting two different selection gadgets cross. Moreover, we can assume that each pair of them crosses at most once, and only in the edges of their paths, not in the edges connecting the pending vertices. We replace each such crossing one by one by a planar *crossing gadget*, such that the resulting planar graph H can only be partitioned into equitable connected partitions if and only if the graph H' can.

Suppose that in our drawing of H' the $(\{1,\ldots,|E|\},|E|)$-choices between anchors A and B and between C and D cross. The crossing gadget is formed by four anchors R, S, X, Y such that the anchor R is connected to A, X to C, S to B, Y to D and X to Y by a $(\{1,\ldots,|E|\},|E|)$-choice, respectively. X is connected to both R and S by $(\{z\cdot(Z_0+1)\mid 1\leq z\leq |E|\},|E|\cdot(Z_0+1))$-choices and Y is connected to both R and S by $(\{z\cdot Z_0\mid 1\leq z\leq |E|\},|E|\cdot Z_0)$-choices (see Fig. 3). The anchors R, S and Y need $|E|\cdot(Z_0+1)$ vertices to be added to their respective classes, while X needs $|E|\cdot(Z_0+2)$ vertices.

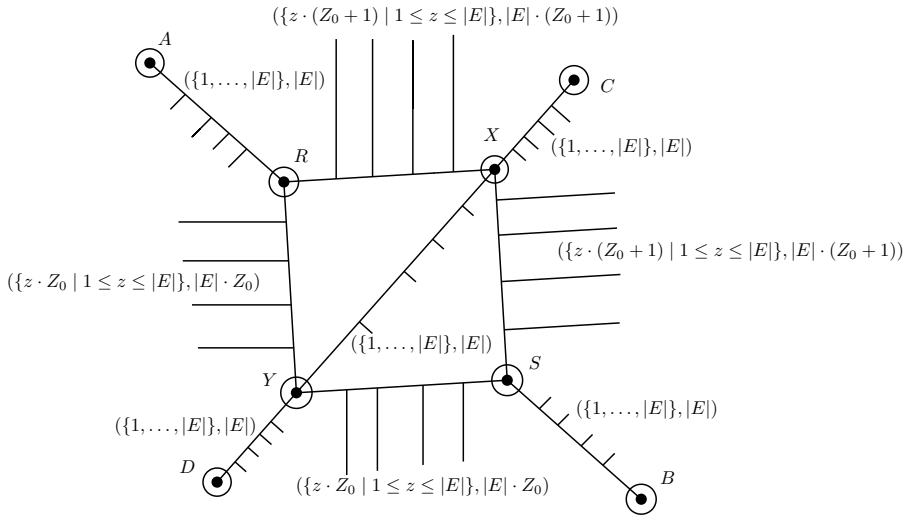

Fig. 3. The crossing gadget.

If X gets $p\cdot(Z_0+1)$ vertices from the choice connecting it to R, then it can get between 2 and $2\cdot|E|$ vertices from the connections to Y and C, and hence it must take between $|E|\cdot(Z_0+2)-p\cdot(Z_0+1)-2<(|E|-p+1)\cdot(Z_0+1)$ and $|E|\cdot(Z_0+2)-p\cdot(Z_0+1)-2\cdot|E|>(|E|-p-1)\cdot(Z_0+1)$ and thus $(|E|-p)\cdot(Z_0+1)$ vertices out of the choice connecting it to S. The anchor Y works in a similar way. Thus, if C takes q vertices from the connection to X, then X gets $|E|-q$ vertices, and takes q vertices from the connection to Y; Y does similarly, and thus D gets $|E|-q$ vertices, as if it were connected to C directly.

On the other hand, if R takes $(|E|-p)\cdot(Z_0+1)$ vertices from the connection to X, it can get between 1 and $|E|$ vertices from the connection to A, and hence it must take $p\cdot(Z_0+1)-|E|>(p-1)\cdot Z_0$ and $p\cdot(Z_0+1)-1<(p+1)\cdot Z_0$ and thus $p\cdot Z_0$ vertices from the connection to Y. It follows that it takes p vertices out of the connection to A. Due to the way X and Y work, S gets $p\cdot(Z_0+1)$ vertices from the connection to X and $(|E|-p)\cdot Z_0$ vertices from the connection

to Y, hence it takes $|E| - p$ vertices from the connection to B and B gets p vertices, as if it were directly connected to A.

It is now clear that the resulting graph H is planar, and can be equitably partitioned into connected partitions if and only if the graph H' constructed in Theorem 1 can. In the graph H' the selection gadgets are interconnected by at most $k(k-1)$ choices, which can cross at most $\binom{k(k-1)}{2} = \frac{1}{2}k(k-1)(k(k-1)-1)$ times. Each such crossing is in H replaced by a crossing gadget with four anchors. Hence, in total, there are at most $2k(k-1) + 2k(k-1)(k(k-1)-1) = 2k^2(k-1)^2 = O(k^4)$ anchors in graph H. Note, that the roots of all anchors form again a feedback vertex set in the graph H, and the pathwidth of the graph H is also bounded by the number of roots plus one. The maximum demand of an anchor is now $|E| \cdot (Z_0 + 2)$, hence we have to set the desired class size to be $s := 2|E| \cdot (Z_0 + 2) + 10 = 2|E| \cdot (2|E| + 10 + 2) + 10 = 4|E|^2 + 24|E| + 10$. The construction can again be clearly carried out in polynomial time; the graph H has $r \cdot s = O(k^4 \cdot |E|^2)$ vertices. □

Since the treewidth of a graph is never greater than its pathwidth, we immediately get also the following corollary:

Corollary 2. Equitable Connected Partition *is W[1]-hard for planar graphs with respect to the treewidth* $tw(G)$.

3 Algorithmic results

Before we give the FPT results, we note that the Equitable Connected Partition problem is in XP, parameterized by the treewidth. This can be proved using standard techniques for problems on graphs of bounded treewidth.

Theorem 2. Equitable Connected Partition *is in XP with respect to the treewidth* $tw(G)$.

Now we present two FPT results for ECP:

Theorem 3. Equitable Connected Partition *is in FPT with respect to the minimum size of a vertex cover* $vc(G)$.

Proof. Assume that we are given a vertex cover $C \subseteq V$ of size $c := vc(G)$. If not, we can compute it in time $O(1.2738^c + cn)$ by [1]. Each class that contains at least 2 vertices must contain some vertex of C; otherwise, it would not be connected. Hence, if the minimum size of each class s is at least 2, then either $r \leq c$ or (G, r) is a no-instance. If $s = 1$, then we have l classes of size 2 and $r - l$ of size 1. Since a size-2 class contains two vertices connected by an edge, such a partitioning is in fact a matching of size l in G. The existence of such a matching can be decided in polynomial time. The case of $s = 0$ is trivial and yields a yes-instance. Hence, in what follows we can assume that $s \geq 2$ and $r \leq c$.

We search for an equitable partitioning such that the first l classes are the larger ones and the last $r - l$ are the smaller ones. We start by trying all the

possibilities of partitioning the vertices of C into r (not necessarily connected) non-empty classes $V_1^C, \ldots, V_r^C$. For each such partitioning, and each disconnected class V_i^C, we try the possibilities of adding at most $|V_i^C| - 1$ vertices of $V \setminus \bigcup_{i=1}^{r} V_i^C$ into V_i^C to make it connected. But we do not try all the vertices. Instead, each vertex tried must have a different neighborhood. We try all such possibilities with different neighborhoods. It remains to distribute the remaining vertices among the classes so that the partitioning becomes equitable. We construct a network such that there is a flow of certain size in it if and only if the vertices can be distributed among the classes.

Let us denote by $D := V \setminus \bigcup_{i=1}^{r} V_i^C$ the set of vertices that are not used yet. The network consists of three intermediate layers, in addition to the source z and the target t. There are r vertices in the first layer, denoted $a_1, \ldots, a_r$. Vertex a_i is connected to the source z by an arc of capacity equal to the number of vertices that should still be added to class i, i.e., $s - |V_i^C|$ if $i > l$ and $s + 1 - |V_i^C|$ if $i \leq l$. The second layer is formed by the vertices of C, and for each i, there are arcs of capacity ∞ from a_i to each vertex in $V_i^C \cap C$. The third layer is formed by vertices $b_J, J \subseteq C$, and there is an arc between $v \in C$ and b_J if and only if $v \in J$. Such arcs have also infinite capacity. Finally each b_J is connected to t by an arc with capacity equal to the number of vertices in D with neighborhood J.

The flow on arcs between the vertices of C and the vertices of type b_J directly shows how many vertices of the particular type should be put into the same class as a vertex of C. Hence it is easy to see that there is a flow of size $|D|$ in the constructed network if and only if the vertices can be distributed among the classes. Concerning the running time of the algorithm, there are at most r^c different colorings of C; for each of them we try adding at most $c-1$ vertices to the classes, each of at most 2^c types. Hence there are at most $O(2^{c^2})$ possibilities to do so. The network can be constructed in time linear in the number of edges of the original graph and the number of vertex types. The flow can be found in the time cubic in the number of vertices of the network, i.e., $O((2^c+2c+2)^3)$. Hence the overal running time of the algorithm is bounded by $O(2^{c^2+c\cdot\log c+4c} \cdot n^2)$. □

Theorem 4. Equitable Connected Partition*(ECP) is in FPT with respect to the maximum number of leaves in a spanning tree* $ml(G)$.

Proof. We construct an instance of Integer Linear Programing (ILP) whose number of variables is a function of $ml(g)$ for ECP. It is known that G is a subdivision of some graph H on (at most) $4k$ vertices, for $ml(G) = k$ [3]. Such a graph H can be easily found in linear time. We say that a class is *simple* if it contains no vertex of H. For an edge uv of H we use P_{uv} to denote the unique path in G having as endpoints u and v, and whose internal vertices are in $G \setminus H$; let $|P_{uv}|$ denote the number of its internal vertices.

We start by trying all the possibilities of partitioning the vertices of H into the first at most $4k$ classes. For each such partitioning, we construct an ILP instance as follows. Assume that the classes used by the vertices of H are $V_1, \ldots, V_p$. Recall that $s := \lfloor n/r \rfloor$. If a path P_{uv} has at least s internal vertices, then it cannot be fully contained in one partition. Hence, it must be split into several parts.

The first part is put into the same partition as the vertex u, the last part into the same class as v and the rest is divided into several simple classes. Since the order of the simple classes on the path does not matter, we only have to know the number of classes having size s, and the number of classes having size $s+1$. Hence, for such an edge uv of H, we introduce four variables: $t_{u,uv}$ and $t_{v,uv}$ representing the number of internal vertices of the path to be placed in the same class as u and v, respectively, and a_{uv} and b_{uv} representing the number of simple classes of size s and $s+1$ on P_{uv}, respectively.

If a path P_{uv} contains less than s internal vertices, then there is no simple class on this path, and each vertex of the path is in the same partition as one of the endpoints. In particular, if u and v are in the same class, then the whole path P_{uv} is in that class. If u and v are in different classes, then we introduce two variables $t_{u,uv}$ and $t_{v,uv}$ for that path, with the same meaning as in the previous case. To simplify the equations, we denote by E_1 the set $\{uv \in E(H) \mid |P_{uv}| < s$ and u and v lie in different partitions$\}$, E_2 the set $\{uv \in E(H) \mid |P_{uv}| \geq s\}$, $E_3^i := \{uv \in E(H) \mid |P_{uv}| < s \text{ and } u, v \in V_i\}$ and $E_3 := \bigcup_{i=1}^{p} E_3^i$. The partition V_i is connected if and only if the graph (V_i, E_3^i) is connected. We check that before we call the procedure to solve the ILP. Finally, we introduce a $\{0,1\}$-variable c_i, for each $1 \leq i \leq p$, so that the size of V_i in the final partitioning will be $s + c_i$. The variables introduced are subject to following constraints:

$$\forall uv \in E_1 \cup E_2 : 0 \leq t_{u,uv}, t_{v,uv}, \tag{1}$$

$$\forall uv \in E_2 : 0 \leq a_{uv}, b_{uv}, \tag{2}$$

$$\forall i, 1 \leq i \leq p : 0 \leq c_i \leq 1, \tag{3}$$

$$\forall uv \in E_1 : t_{u,uv} + t_{v,uv} = |P_{uv}|, \tag{4}$$

$$\forall uv \in E_2 : t_{u,uv} + t_{v,uv} + s \cdot a_{uv} + (s+1) \cdot b_{uv} = |P_{uv}|, \tag{5}$$

$$\forall i, 1 \leq i \leq p : \sum_{v \in V(H) \cap V_i} (1 + \sum_{uv \in E_1 \cup E_2} t_{v,uv}) + \sum_{uv \in E_3^i} |P_{uv}| = s + c_i, \tag{6}$$

$$\sum_{uv \in E_2} a_{uv} + \sum_{i=1}^{p} (1 - c_i) = r - l, \tag{7}$$

$$\sum_{uv \in E_2} b_{uv} + \sum_{i=1}^{p} c_i = l. \tag{8}$$

Equation 4 ensures that the paths corresponding to the edges in E_1 are correctly divided. Equation 5 ensures the same for the edges in E_2. Equation 6 ensures that the parts containing some vertices of H have the right size, and the last two equations (7 and 8) ensure that there are the right numbers of large and small classes. It is easy to see that there is a solution to this ILP instance if and only if there is an equitable connected partitioning of G with r partitions, which extends the starting partitioning of $V(H)$. Since there are at most $4 \cdot \binom{4k}{2} + 4k \leq 32k^2$ variables, each of them used at most three times, the overall size of the instance is bounded by $O(k^2)$ and it can be solved in $O((32k^2)^{2.5 \cdot 32k^2 + o(k^2)} \cdot k^2)$

time [7], which yields a running time of $O(m \cdot 2^{160k^2 \log k + o(k^2 \log k)})$ for the whole algorithm. □

4 Equitable coloring

In this section we show that Equitable Coloring is in FPT with respect to the maximum number $ml(G)$ of leaves in a spanning tree of G. Since a graph with bounded $ml(G)$ contains a lot of induced paths, we first examine the situation on the paths. There is a nice characterisation lemma for that case:

Lemma 1 (Lemma 2, Appendix). *Let $k \geq 2$ be an integer. Let P be a path with endpoints possibly colored by one of the colors $1, \ldots, k$. Let n be the number of uncolored vertices on the path, and $\forall i$ let $t(i) \in \{0, 1, 2\}$ be the number of endpoints colored by color i. Then P can be properly colored by the colors $1, \ldots, k$ such that there are $n_i + t(i)$ vertices of color i if and only if $\forall i : 0 \leq n_i \leq \lceil \frac{1}{2}(n - t(i)) \rceil$ and $\sum_i n_i = n$.*

Theorem 5. Equitable Coloring *is in FPT with respect to the maximum number of leaves in a spanning tree $ml(G)$.*

Proof. First we show that if there are many classes and the graph is big, then we have a yes-instance; otherwise, we construct an instance of Integer Linear Programing (ILP) for Equitable Coloring. It is known that G is a subdivision of some graph H on (at most) $4k$ vertices, for $ml(G) = k$ [3]. Such a graph H can be easily found in linear time. For an edge uv of H we use P_{uv} to denote the unique path in G having as endpoints u and v, and whose internal vertices are in $G \setminus H$. Let $|P_{uv}|$ denote the number of its internal vertices.

Claim. If $r \geq 16k$ and $n := |V(G)| \geq 32k^2$ then (G, r) is a yes-instance.

Proof (of Claim). We color the vertices of H arbitrarily by at most $4k$ colors. Then we fill these color classes with some (at most $4k(\lceil n/r \rceil - 1) \leq 4kn/r \leq n/4$) vertices. To avoid conflicts, we do not use the (at most $4k(4k-1)$) vertices that are neighbors in G of the H-vertices to fill the at most $4k$ classes. Since the rest of the graph is just a collection of paths, we can color at least each second vertex by one of the at most $4k$ colors used for the H-vertices. Hence there are at least $(n - 16k^2)/2 \geq n/4$ available vertices which are at least as many as we need. Now we are left with the task of equitably coloring the collection of paths with at least $12k$ colors, which is possible due to Lemma 1.

If the graph has at most $32k^2$ vertices, then we can solve the instance by brute force. If this is not the case, we can assume, due to the claim, that the number of colors r is less than $16k$. Now we try all the possibilities $c : V(H) \to \{1, \ldots r\}$ to color the vertices of H. For each such possibility, we construct an instance of ILP, which will have a variable q^i_{uv} for each combination of color i and an edge uv of H. This variable expresses the number of the vertices of color i on the path P_{uv}. They are subject to the constraints given by Lemma 1 and the constraints

that enforce the classes to have the right number of vertices. In the following formal description of the constaints, for a logical formulae ϕ the expression $[\phi]$ is 1 if ϕ is true and 0 otherwise. Note that these expressions as well as the ceilings only appear on the constant sides of the equations.

$$\forall uv \in E(H), 1 \leq i \leq r : 0 \leq q_{uv}^i \leq \left\lceil \frac{1}{2}(|P_{uv}| - [c(u) = i] - [c(v) = i]) \right\rceil,$$

$$\forall uv \in E(H) : \sum_{i=1}^{r} q_{uv}^i = |P_{uv}|,$$

$$\forall i, 1 \leq i \leq l : \sum_{uv \in E(H)} q_{uv}^i = s + 1 - \sum_{v \in V(H)} [c(v) = i],$$

$$\forall i, l+1 \leq i \leq r : \sum_{uv \in E(H)} q_{uv}^i = s - \sum_{v \in V(H)} [c(v) = i].$$

Clearly, there is a solution for Equitable Coloring if there is a solution to the ILP for one of the colorings c. Since an instance X of ILP with t variables can be solved in time $O(t^{2.5t+o(t)} \cdot |X|)$ [7], the overall running time is at most $O((32k^2)^{32k^2} + (256k^3)^{2.5 \cdot 256k^3 + o(k^3)} poly(n))$, where the polynomial is independent of k.

□

References

1. J. Chen, I. A. Kanj, and G. Xia. Improved parameterized upper bounds for vertex cover. In R. Kralovic and P. Urzyczyn, editors, *MFCS*, volume 4162 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2006.
2. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
3. V. Estivill-Castro, M. R. Fellows, M. A. Langston, and F. A. Rosamond. FPT is P-Time Extremal Structure I. In H. Broersma, M. J. 0002, and S. Szeider, editors, *ACiD*, volume 4 of *Texts in Algorithmics*, pages 1–41. King's College, London, 2005.
4. M. R. Fellows, F. V. Fomin, D. Lokshtanov, F. A. Rosamond, S. Saurabh, S. Szeider, and C. Thomassen. On the complexity of some colorful problems parameterized by treewidth. In A. W. M. Dress, Y. Xu, and B. Zhu, editors, *COCOA*, volume 4616 of *Lecture Notes in Computer Science*, pages 366–377. Springer, 2007.
5. M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.
6. J. Fiala, P. A. Golovach, and J. Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. In J. Chen and S. B. Cooper, editors, *TAMC*, volume 5532 of *Lecture Notes in Computer Science*, pages 221–230. Springer, 2009.
7. A. Frank and E. Tardos. An application of simultaneous diophantine approximation in combinatorial optimization. *Combinatorica*, 7(1):49–65, 1987.
8. D. West. *Introduction to graph theory*. Prentice Hall Inc., Upper Saddle River, NJ, 1996.

5 Appendix

Lemma 2. *Let $k \geq 2$ be an integer. Let P be a path with endpoints possibly colored by one of the colors $1, \ldots, k$. Let n be the number of uncolored vertices on the path, and $\forall i$ let $t(i) \in \{0, 1, 2\}$ be the number of endpoints colored by color i. Then P can be properly colored by the colors $1, \ldots, k$ such that there are $n_i + t(i)$ vertices of color i if and only if $\forall i : 0 \leq n_i \leq \lceil \frac{1}{2}(n - t(i)) \rceil$ and $\sum_i n_i = n$.*

Proof. The "only if" part is easy; the proof of the "if" part is by induction on n. The cases $n = 1$ and $n = 2$ are trivial. For $n \geq 3$, let us assume that $\lceil \frac{1}{2}(n - t(1)) \rceil - n_1 \leq \lceil \frac{1}{2}(n - t(2)) \rceil - n_2 \leq \lceil \frac{1}{2}(n - t(3)) \rceil - n_3 \leq \ldots \leq \lceil \frac{1}{2}(n - t(k)) \rceil - n_k$. It is then easy to check that, for $n \geq 3$, we always have $\lceil \frac{1}{2}(n - t(3)) \rceil - n_3 > 0$. In what follows we distinguish several cases. In each of them we can color the first uncolored vertex by one of the colors $1, 2$, and color the rest using the induction hypothesis for the path starting with the newly colored vertex, $n' := n - 1$, with the sizes n'_i of the color classes and the numbers of the colored enpoints $t'(i)$ set appropriately. To do so, it is enough to show that $\lceil \frac{1}{2}(n' - t'(1)) \rceil - n'_1 \geq 0$ and $\lceil \frac{1}{2}(n' - t'(2)) \rceil - n'_2 \geq 0$, since for $i \geq 3$ we have $\lceil \frac{1}{2}(n' - t'(i)) \rceil - n'_i \geq \lceil \frac{1}{2}(n - t(i)) \rceil - 1 - n_i \geq 0$. Note also that if we use the color $i \in \{1, 2\}$ then $\lceil \frac{1}{2}(n' - t'(i)) \rceil - n'_i = \lceil \frac{1}{2}(n - 1 - (t(i) - 1)) \rceil - (n_i - 1) = \lceil \frac{1}{2}(n - t(i)) \rceil - (n_i) \geq 0$.

- If one of the endpoints has color 1, then we color its neighbor by 2. Since $n'_1 = n_1$ and $t'(1) = t(1) - 1$, the induction hypothesis applies in this case.
- If one of the endpoints has color 2, then we color its neighbor by 1. Since $n'_2 = n_2$ and $t'(2) = t(2) - 1$, the induction hypothesis applies in this case.
- In any other case we color the first uncolored vertex by color 1. Note that in this case $t(1) = t(2) = 0$ and thus if $\lceil \frac{1}{2}(n - t(2)) \rceil - n_2 = 0$ then necessarily $n_1 = n_2 = \lceil \frac{1}{2}n \rceil$ and thus n must be even. But then $\lceil \frac{1}{2}n' \rceil = \lceil \frac{1}{2}n \rceil$ and hence the induction hypothesis also applies.

□