Parameterized Algorithms and Hardness for the Maximum Edge q-Coloring Problem

Rogers Mathew
Department of Computer Science and Engineering, IIT Hyderabad, India

Fahad Panolan
School of Computer Science, University of Leeds, UK

Seshikanth
Department of Computer Science and Engineering, IIT Hyderabad, India

Abstract

An *edge q-coloring* of a graph G is a coloring of its edges such that every vertex sees at most q colors on the edges incident on it. The *size* of an edge q-coloring is the total number of colors used in the coloring. Given a graph G and a positive integer t, the MAXIMUM EDGE q-COLORING problem is about whether G has an edge q-coloring of size t. Studies on this coloring problem were motivated by its application in the channel assignment problem in wireless networks.

Goyal, Kamat, and Misra (MFCS 2013) studied MAXIMUM EDGE 2-COLORING from the perspective of parameterized complexity. Given a graph on n vertices, they considered the standard parameter t, the number of colors in an optimal edge 2-coloring, and the dual parameter ℓ, where $n-\ell$ is the number of colors in an optimal edge 2-coloring. They designed FPT algorithms for MAXIMUM EDGE 2-COLORING parameterized by t and ℓ. In this paper, we revisit and study MAXIMUM EDGE 2-COLORING from the perspective of parameterized complexity and show the following results.

1. Let $\gamma(G)$ denote the maximum matching size in a given graph G. It is easy to see that a maximum edge 2-coloring of G is of size at least $\gamma(G)$. Goyal, Kamat, and Misra (MFCS 2013) had asked if there exists an FPT algorithm for MAXIMUM EDGE 2-COLORING parameterized by k, where $k :=$ (size of a maximum edge 2-coloring of G) $- \gamma(G)$. We show that MAXIMUM EDGE 2-COLORING parameterized by k is W[1] hard.
2. On the positive side, we show that there is an algorithm that, given a graph G on n vertices and a tree decomposition of width tw, runs in time $2^{O(q\mathsf{tw} \log q\mathsf{tw})}n$ and outputs a maximum edge q-coloring of G.

2012 ACM Subject Classification Theory of computation → Fixed parameter tractability

Keywords and phrases FPT algorithm, Edge coloring, Treewidth, W[1]-hardness

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2024.31

1 Introduction

Given a graph G and a positive integer q, an *edge q-coloring* is a coloring (not necessarily proper) of the edges of G such that every vertex sees at most q colors on the edges incident on it. The *size of an edge q-coloring* is the total number of colors used in the coloring. Assigning every edge of G the same color is indeed a valid edge q-coloring. However, we are interested in obtaining a *maximum edge q-coloring* of G, an edge q-coloring of G of the maximum possible size.

In 2005, Raniwala, Chiueh, and Gopalan in [12] and [11] proposed *Hyacinth*, a multichannel wireless mesh network architecture that uses 802.11 Network Interface Cards (NICs) at each node of the mesh network. They observed that two NICs on each node may improve network throughput by a factor of 6 to 7 compared to conventional single-channel ad hoc networks. A network can be modeled as a graph where each computer is a vertex. Assigning channels

44th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2024).
Editors: Siddharth Barman and Sławomir Lasota; Article No. 31; pp. 31:1–31:12
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to computers with two interface cards corresponds to an edge 2-coloring of the graph. The maximum number of colors in an edge 2-coloring represents the number of channels that can be used simultaneously in the network.

Maximum edge q-coloring of graphs has been studied from an algorithmic as well as a structural graph theoretic perspective. Adamaszek and Popa [1] proved that the decision version of the maximum edge q-coloring problem is NP-hard for every $q \geq 2$. Further they showed that, for every $q \geq 2$, the maximum edge q-coloring problem is APX-hard. For $q = 2$, the paper gives a 5/3-factor approximation algorithm for graphs having a perfect matching. For triangle-free graphs having a perfect matching, Chandran et al. [3] gave an 8/5-factor approximation algorithm for the maximum edge q-coloring problem. Later, in [2], Chandran et al. showed that the approximation factors of 5/3 and 8/5 can be improved under certain assumptions on the minimum degree of the graph under consideration. As for general graphs, Feng et al. in [8] gave a 2-factor approximation algorithm for the maximum edge 2-coloring problem. In the same paper, the authors showed that the maximum edge 2-coloring problem is polynomial-time solvable for trees and complete graphs. In [6], Dvorak et al. show that there is a PTAS known for the maximum edge q-coloring problem on minor-free graphs.

From a parameterized complexity perspective, Goyal, Kamat, and Misra in [10] gave fixed-parameter tractable algorithms for maximum edge 2-coloring of G for both the standard parameter (say t, if t is the size of an optimal edge 2-coloring of G) and the dual parameter (say ℓ, if $n - \ell$ is the size of an optimal edge 2-coloring, where n is the number of vertices in the input graph). It is known that the maximum number of colors used in an edge 2-coloring of a graph G is at most the number of vertices in G, and hence, with the dual parameter ℓ, the number of colors asked for is $n - \ell$. For the dual parameterization, the authors obtained a linear vertex kernel with $O(\ell)$ vertices and $O(\ell^2)$ edges.

The maximum edge q-coloring problem is related to another parameter called anti-Ramsey number, a concept introduced by Erdős, Simonovitz and Sós in 1975 [7]. Given a host graph G and a pattern graph H, the *anti-Ramsey number* $ar(G, H)$ is defined as the smallest positive integer k such that any coloring of the edges of G with k colors will have a rainbow subgraph (a subgraph no two of whose edges have the same color) isomorphic to H. In other words, $ar(G, H)$ is one more than the largest k for which there exists a coloring of the edges of G with k colors such that there is no rainbow subgraph isomorphic to H under this coloring. See [9] for a survey on anti-Ramsey numbers, a notion that has been extensively studied in extremal graph theory. From its definition, it is clear that the size of a maximum edge q-coloring of a graph G is one less than $ar(G, K_{1,q+1})$, where $K_{1,q+1}$ denotes the complete bipartite graph with 1 vertex in one part and $q + 1$ vertices in the other.

Our contributions

In Section 3 we give a fixed parameter tractable algorithm for Maximum Edge q-Coloring parameterized by the treewidth of the graph under consideration.

▶ **Theorem 1.** *There is an algorithm that, given an n-vertex graph G and its tree decomposition of width* tw*, runs in time $2^{O(\mathsf{tw} \cdot q \log(\mathsf{tw} \cdot q))} n$, and outputs a maximum edge q-coloring of G.*

To explain the significance of this result, let us recall the work of Goyal et al. [10] for Maximum Edge 2-Coloring with respect to the parameter k, the number of colors in the output edge 2-coloring of the graph. Let G be the input graph, and let $\gamma(G)$ denote the size of a maximum matching in G. They first observe that the size of a maximum edge 2-coloring of G is at least $\gamma(G)$ as coloring all the edges in a maximum matching with distinct colors and then coloring the remaining edges with a new color is indeed a valid edge 2-coloring of G.

Thus, the problem becomes challenging only when $\gamma(G) \leq k$. In this case, the vertex cover number of G is at most $2k$. Then, Goyal et al. [10] observed that maximum edge q-coloring can be expressed in Monadic Second Order (MSO_2) logic. Since the treewidth of G is at most its vertex cover number, which is upper bounded by $2k$, an application of Courcelle's theorem [4] provides a fixed parameter tractable (FPT) algorithm for the problem, but its running time will be impractical. Goyal et al. [10] gave a combinatorial algorithm with running time $2^{O(k \log k)} n^{O(1)}$, using the fact that the vertex cover number will be at most $2k$. As a corollary, one gets an FPT algorithm parameterized by the vertex cover number because the number of colors in an edge 2-coloring is at most two times the vertex cover number.

However, the case where the parameter is treewidth is not trivial. First of all, the edge 2-coloring number can be arbitrarily larger than the treewidth. Consider the graph, which is a path on n vertices. Here, the pathwidth (and hence the treewidth) is one, but coloring all the $n-1$ edges with distinct colors is a valid edge 2-coloring. Next, it is tempting to believe that since the problem is expressible in MSO_2, we get an FPT algorithm parameterized by treewidth. In fact, there is a caveat here. The length of the MSO_2 formula depends on the total number of colors used in the edge coloring (hence it is large), and the running time of the algorithm by the application of Courcelle's theorem depends exponentially on the length of the formula as well. Thus, it is important to design an algorithm for the problem when the parameter is treewidth. To the best of our knowledge, this is the first FPT algorithm for the problem when parameterized by treewidth.

Recall the the size of a maximum edge 2-coloring is at least $\gamma(G)$. Goyal et al. [10] asked if there exists an FPT algorithm for the maximum edge 2-coloring problem parameterized by k, where $k :=$ (size of a maximum edge 2-coloring of G) $- \gamma(G)$. In Section 4, we resolve this question by showing that the problem is W[1] hard.

▶ **Theorem 2.** *It is W[1]-hard parameterized by k to decide if the given graph G has an edge 2-coloring using at least $\gamma(G) + k$ colors.*

2 Preliminaries

We use $\mathbb{N}$ to denote the set of natural numbers. For $n \in \mathbb{N}$, we use $[n]$ to denote $\{1, \ldots, n\}$. For a function $f : A \to B$ and $A' \subseteq A$, $f(A') = \{f(a) : a \in A'\}$. For two functions $f : A \to B$ and $g : B \to C$, $g \circ f$ is the function from A to C defined as follows: $(g \circ f)(a) = g(f(a))$, for all $a \in A$. Let $f : A_1 \to B$ and $g : A_2 \to B$ be two functions. We use $f \oplus g$ to denote the function defined as follows: $(f \oplus g)(a) = f(a)$ if $a \in A_1$ and $(f \oplus g)(a) = g(a)$, if $a \in A_2 \setminus A_1$. If $f(a) = g(a)$ for all $a \in A_1 \cap A_2$, then $f \cup g$ denotes the the union of the functions f and g. Here, $(f \cup g)(a) = f(a)$ if $a \in A_1$ and $(f \cup g)(a) = g(a)$, otherwise. If $A_1 \cap A_2 = \emptyset$, then we may also write $f \uplus g$ instead of $f \cup g$.

Throughout this paper, we use simple, undirected graphs. We say a graph is connected if, for any pair of vertices in it, there is a path between them in the graph. Let G be a graph. We use $V(G)$ and $E(G)$ to denote its vertex and edge sets, respectively. We use $\{u, v\}$ as well as uv to denote the edge between vertex u and vertex v. For a vertex subset $U \subseteq V(G)$, $E_G(U)$ denotes the set of edges incident on U. That is, $E_G(U) = \{\{x, y\} \in E(G) : x \in U \vee y \in U\}$. For a vertex v, we use $E_G(v)$ to denote the set $E(\{v\})$. For a vertex subset $U \subseteq E(G)$, we use $E_G[U]$ to denote the set of edges with both endpoints in G. That is, $E_G[U] = \{\{u, v\} \in E(G) : u, v \in U\}$. For an edge subset $F \subseteq E(G)$, we use $V_G(F)$ to denote the set of endpoints of the edges in F. When the graph is clear from the context, we remove the subscript G in the above notations. For a vertex subset $U \subseteq V(G)$, we use $G[U]$ to denote the subgraph of G induced by U. That is, $V(G[U]) = U$ and $E(G[U]) = E_G[U]$. For an edge subset $F \subseteq E(G)$, we use $G[F]$ to denote the subgraph of G induced by F. That is, $V(G[F]) = V_G(F)$ and $E(G[F]) = F$. A *separation* of G is a pair (A, B) of vertex subsets such that $A \cup B = V(G)$, and there is no edge with one endpoint in $A \setminus B$ and the other in $B \setminus A$. The *separator* of this separation is $A \cap B$, and the *order* of the separation is $|A \cap B|$.

The following lemma is a folklore.

▶ **Lemma 3.** *Let G be a graph, $q \in \mathbb{N}$, and $d\colon E(G) \to \mathbb{N}$ be an edge q-coloring using the maximum number of colors. Then, for any color $r \in \mathbb{N}$ with $d^{-1}(r) \neq \emptyset$, $G[d^{-1}(r)]$ is connected.*

Proof. Suppose by contradiction, if $G[d^{-1}(r)]$ is disconnected, we have two disjoint components. It means we can give one new color (say $r' \neq r$) to any one component and strictly increase the number of colors used by one. This will be again a q-coloring of G. However, this is a contradiction as the given coloring uses the maximum number of colors. ◀

Tree decompositions. A tree decomposition of a graph G is a pair $\mathcal{T} = (\mathrm{T}, \{X_t\}_{t \in V(\mathrm{T})})$, where T is a tree and every node $t \in V(\mathrm{T})$ is assigned a vertex subset $X_t \subseteq V(G)$, called a bag, such that the following three conditions hold:

1. $\bigcup_{t \in V(\mathrm{T})} X_t = V(G)$.
2. For every $\{u, v\} \in E(G)$, there is a node $t \in V(\mathrm{T})$ such that $u, v \in X_t$.
3. For every $u \in V(G)$, the subgraph of T induced by $\mathrm{T}_u = \{t \in V(\mathrm{T}) : u \in X_t\}$ is connected.

The *width* of tree decomposition $\mathcal{T} = (\mathrm{T}, \{X_t\}_{t \in V(\mathrm{T})})$ is $\max_{t \in V(\mathrm{T})} |X_t| - 1$. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimum width among all tree decompositions of G. To explain dynamic programming in an easier way, we recall the definition of a nice tree decomposition. A *nice* tree decomposition if a $\mathcal{T} = (\mathrm{T}, \{X_t\}_{t \in V(\mathrm{T})})$ where T is a rooted tree and satisfies the following additional conditions. Let r be the root node in T.

1. $X_r = \emptyset$ and $X_\ell = \emptyset$ for every leaf node ℓ.
2. Each non-leaf node of T has one of the following three types:
   - **Introduce node:** A node t with exactly one child t' such that $X_t = X_{t'} \cup \{v\}$ and $v \notin X_{t'}$; we say that v is *introduced* at t.
   - **Forget node:** a node t with exactly one child t' such that $X_t = X_{t'} \setminus \{w\}$ for some vertex $w \in X_{t'}$; we say that w is *forgotten* at t.
   - **Join node:** a node t with two children t_1 and t_2 such that $X_t = X_{t_1} = X_{t_2}$.

▶ **Lemma 4** (Lemma 7.4 in [5]). *If a graph G admits a tree decomposition of width at most k, then it also admits a nice tree decomposition of width at most k. Moreover, given a tree decomposition $\mathcal{T} = (\mathrm{T}, \{X_t\}_{t \in V(\mathrm{T})})$ of G of width at most k, one can in time $O(k^2 \cdot \max(|V(T)|, |V(G)|))$ compute a nice tree decomposition of G of width at most k that has at most $O(k|V(G)|)$ nodes.*

Parameterized complexity. A parameterized problem P is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is the finite alphabet. Fixed parameter traceability of a problem P means whether we can decide the problem in $O(f(k) \cdot p(n))$ time, where k is the fpt parameter, n is the input size, $f(.)$ is some arbitrary function and $p(.)$ is a polynomial function.

▶ **Definition 5** (Parameterized reduction [5]). *Let $A, B \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A* parameterized reduction *from A to B is an algorithm that, given an instance (x, k) of A, outputs an instance (x', k') of B such that*

1. *$(x, k) \in A \iff (x', k') \in B$.*
2. *$k' \leq g(k)$ for some computable function $g(.)$, and*
3. *the running time is $f(k) \cdot |x|^{O(1)}$ for some computable function $f(.)$.*

3 FPT algorithm parameterized by treewidth

We give a dynamic programming algorithm for Maximum Edge q-Coloring when a tree decomposition of width at most k is given as part of the input. Without loss of generality, we assume that a nice tree decomposition is part of the input. Let $(G, \mathcal{T} = (T, \{X_t\}_{t \in V(T)}))$ be the input. Let $n = |V(G)|$, $m = |E(G)|$, and the width of the tree decomposition $\mathcal{T}$ is k. Our algorithm should output an edge q-coloring of G using the maximum number of colors.

Let us define some notations which we use in this section. Recall that T is a rooted tree. Let r be the root of T. For a node $t \in V(T)$, V_t is the union of the bags in the subtree rooted at t. That is, if T_t is the subtree of T rooted at t, then $V_t = \bigcup_{t' \in V(T_t)} X_{t'}$. We use G_t to denote the graph $G[V_t]$. Notice that for any $t \in V(T)$, $(V_t, (V(G) \setminus V_t) \cup X_t)$ is a separation of G of order $|X_t|$, which is upper bounded by k. Now consider the following lemma.

▶ **Lemma 6.** *Let G be a graph, $q \in \mathbb{N}$, and $d \colon E(G) \to \mathbb{N}$ be an edge q-coloring using a maximum number of colors. Let (A, B) be a separation of G and there is a color r such that $r \in d(E_G[A \setminus B])$ and $r \notin d(E_G[A] \setminus E_G[A \setminus B])$. Then, $r \notin E_G[B]$.*

Proof. Let r be the color specified in the lemma and let $\{u, v\} \in d^{-1}(r)$. From the definition of $E_G[A \setminus B]$, we have $u, v \in A \setminus B$. We are given that $r \notin d(E_G[A] \setminus E_G[A \setminus B])$. It means there is no edge with at least one endpoint in $A \cap B$ with color r. The reason is that $E_G[A \setminus B]$ is the set of edges with both the endpoints in $A \setminus B$ and $E_G[A] = \{\{x, y\} : x, y \in A \setminus B\} \cup \{\{x, y\} : x \in A \cap B \vee y \in A \cap B\}$. This implies that there is no edge from $E_G[A]$ with at least one endpoint in $A \cap B$ and colored with r. We need to prove that there is no edge in $E_G[B]$ colored with r. Now, for the sake of contradiction, say we have an edge with color r, and it is in $E_G[B]$. By Lemma 3, there should be a path from the edge $\{u, v\}$ (as defined above) to this edge in $E_G[B]$ and all the edges in this path are colored r. This path must pass through at least one vertex in $A \cap B$ as (A, B) is a separation of G. But it leads to contradiction as we are given that $r \notin d(E_G[A] \setminus E_G[A \setminus B])$. It means $r \notin E_G[B]$. ◀

Lemma 6 helps us to design a dynamic programming algorithm. Because of Lemma 6, at any node $t \in V(T)$, for any coloring of G_t, we do not need to remember the colors of the edges incident on the vertices other than X_t, as it will not be used to color the "future" edges. More formally, in the dynamic programming for any node t we compute and store a set $\mathcal{C}_t$ of edge q-colorings of G_t. Here, we use two disjoint sets of colors; the first set is $[qk] = \{1, 2, \ldots, qk\}$ and the other set is $\{a_1, a_2, \ldots, a_{nq}\}$. We use only the colors from $[qk]$ to color the edges incident on X_t. Other edges in G_t can be colored with any colors from $[qk] \cup \{a_1, a_2, \ldots, a_{nq}\}$. But, we make sure that if an edge in G_t is colored with a color c from $[qk]$, then at least one edge incident on X_t is colored with c. Consider two edges q-coloring g_1 and g_2 of G_t that satisfy the conditions mentioned above. We say that g_1 and g_2 are equivalent, denoted by $g_1 \sim_t g_2$, if the following conditions hold.

(i) $|g_1(E(G_t))| = |g_2(E(G_t))|$. That is, the number of colors used by both g_1 and g_2 are the same.

(ii) For all $u \in X_t$, $g_1(E_{G_t}(u)) = g_2(E_{G_t}(u))$. That is, the colors seen by the edges incident on u are the same in both the colorings g_1 and g_2, for any vertex $u \in X_t$.

Lemma 6 implies that if $g_1 \sim_t g_2$, and g_1 can be extended to a maximum edge q-coloring of G, then g_2 can be extended to a maximum edge q-coloring of G. This is formulated in the following lemma.

▶ **Lemma 7.** *Suppose $g_1 \sim_t g_2$. If there is an edge q-coloring f_1 of G such that $f_1|_{E(G_t)} = g_1$, then there is an edge q-coloring f_2 of G such that $f_2|_{E(G_t)} = g_2$ and $|f_1(E(G))| = |f_2(E(G))|$.*

Proof. We know that both g_1 and g_2 satisfy the following conditions.

(a) All the edges incident on X_t are colored using the colors from $[qk]$ by g_1 and g_2.

(b) All the edges in G_t are colored using colors from $[qk] \cup \{a_1, \ldots, a_{qn}\}$. Moreover, for each $i \in [2]$, if a color $c \in [qk]$ is used by g_i, then there is an edge $e \in E_{G_t}(X_t)$ such that $g_i(e) = c$.

Since $g_1 \sim_t g_2$ and condition (b) above implies that the number of colors from $\{a_1, \ldots, a_{qn}\}$ used by both g_1 and g_2 are same. Without loss of generality let $a_1, \ldots, a_\ell$ be the colors in $\{a_1, \ldots, a_{qn}\}$ used by the coloring g_2. As mentioned before, there are exactly ℓ colors from $\{a_1, \ldots, a_{qn}\}$ used by g_1. Let $i_1, \ldots, i_\ell$ be the distinct indices in $[qn]$ such that the colors from $\{a_1, \ldots, a_{qn}\}$ used by g_1 are $a_{i_1}, \ldots, a_{i_\ell}$. Now, we obtain a coloring $\widehat{f_1}$ from f_1 as follows. Let $\beta : \{a_1, \ldots, a_{qn}\} \cup [qk] \to \{a_1, \ldots, a_{qn}\} \cup [qk]$ be an arbitrary bijection such that $\beta(r) = r$ for all $r \in [qk]$ and $\beta(a_{i_j}) = a_j$ for all $j \in [\ell]$. Now the coloring $\widehat{f_1}$ is defined as follows. For each $e \in E(G_t)$, $\widehat{f_1}(e) = \beta(f_1(e))$ and for each $e \in E(G) \setminus E(G_t)$, $\widehat{f_1}(e) = f_1(e)$. Let $\widehat{f_1}|_{E(G_t)} = \widehat{g}_1$. Since β is a bijection as defined above, $\widehat{f_1}$ is an edge q-coloring of G, $\widehat{g}_1 \sim_t g_2$, $|f_1(E(G))| = |\widehat{f_1}(E(G))|$, and $\widehat{g}_1(E(G_t)) = g_2(E(G_t))$.

We define an edge q-coloring f_2 of G as below:

$$f_2(e) = \begin{cases} \widehat{f_1}(e), & \text{if } e \in E(G) \setminus E(G_t) \\ g_2(e), & \text{otherwise, i.e., } e \in E(G_t) \end{cases}$$

Since $g_2(E(G_t)) = \widehat{g}_1(E(G_t))$ and $\widehat{f_1}|_{E(G_t)} = \widehat{g}_1$, we have $|f_2(E(G))| = |\widehat{f_1}(E(G))| = |f_1(E(G))|$. Next we prove that indeed f_2 is an edge q-coloring of G. Towards that we need to prove that for all $u \in V(G)$, $|f_2(E_G(u))| \leq q$. Fix a vertex $u \in V(G)$. First, consider the case when $u \in V_t \setminus X_t$. Then, $f_2(E(u)) = g_2(E(u))$ and g_2 is an edge q-coloring of the induced subgraph G_t. Hence, $|f_2(E(u))| \leq q$. Now, consider the case $u \in X_t$. Since $\widehat{g}_1 \sim_t g_2$, $\widehat{g}_1(E_{G_t}(u)) = g_2(E(G_t(u))$. Thus, by the definition of f_2, we get that $f_2(E_G(u)) = \widehat{f_1}(E_G(u))$. This implies that $|f_2(E(u))| \leq q$. Finally, consider the case when $u \in V(G) \setminus V_t$. In this case, $f_2(E_G(u)) = \widehat{f_1}(E_G(u))$ and hence $|f_2(E(u))| \leq q$. ◀

Because of Lemma 7, it is enough to keep one coloring from an equivalence class of $\sim_t$. Let $\mathcal{S}_t$ be the set of all edge q-colorings of G_t such that

(a) all the edges incident on X_t are colored from the set $[qk]$,

(b) all other edges are colored from the set $[qk] \cup \{a_1, \ldots, a_{nq}\}$, and

(c) if an edge in G_t is colored with a color c from $[qk]$, then at least one edge incident on X_t is colored with c.

Two colorings in $\mathcal{S}_t$ are equivalent in $\sim_t$ if the conditions (i) and (ii) defined before, hold.

▶ **Lemma 8.** *The number of equivalence classes in $\sim_t$ is upper bounded by $\binom{qk}{q}^k \cdot q^{k+1} \cdot n$.*

Proof. From the definition of an equivalence class, the number of equivalence classes is determined by the product of the number of possibilities for conditions (i) and (ii) mentioned above in this section. For any n-vertex graph G, the number of colors used by any edge q-coloring is at most nq, because for any vertex, the number of colors used for the incident edges is at most q. Recall that $|X_t| \leq k$. Next we count the number of distict combination for the condition (2) (i.e., for all $u \in X_t$, $g_1(E_{G_t}(u)) = g_2(E_{G_t}(u))$). This number is upper bounded by $(\sum_{j=0}^{q} \binom{qk}{j})^k \leq q^k \binom{qk}{q}^k$, since for any vertex in X_t, we choose at most q out of qk colors for coloring the incident edges. Therefore, the total number of equivalence classes in $\sim_t$ is upper bounded by $\binom{qk}{q}^k \cdot n \cdot q^{k+1}$. ◀

Because of Lemmas 7 and 8, for each node t in T, we compute and store a family $\mathcal{C}_t \subseteq \mathcal{S}_t$ of edge q-colorings of G_t such that $|\mathcal{C}_t| \leq 2^{O(qk \log qk)} \cdot n$ and at least one coloring in it can be extended to a maximum edge q-coloring of G. Now, we explain how to compute $\mathcal{C}_t$ in a bottom-up fashion. Let t be a node in T and assume that we have computed $\mathcal{C}_{t'}$ for all node t' such that $t \neq t'$ and t' is a node in the subtree rooted at t.

Leaf Node. In this case $V(G_t) = \emptyset$, and hence $\mathcal{C}_t$ contains only one coloring which is an empty function.

Introduce Node. Suppose t is an introduce node with a child t' such that $X_t = X_{t'} \cup \{v\}$ for some $v \notin X_{t'}$. Recall that we have already computed $\mathcal{C}_{t'}$. Notice that v is adjacent to at most $k-1$ vertices in G_t. Moreover, $N_{G_t}(v) \subseteq X_t$ and $E(G_t) = E(G_{t'}) \uplus E_{G_t}(v)$. Now construct a set $\mathcal{D}_t$ as follows. Initially we set $\mathcal{D}_t := \emptyset$. Now for each coloring $f \in \mathcal{C}_{t'}$ and each coloring $g : E_{G_t}(v) \rightarrow [qk]$ such that $f \uplus g$ is an edge q-coloring of G_t, we add $f \uplus g$ to $\mathcal{D}_t$. Finally, construct a minimal subfamily $\mathcal{C}_t \subseteq \mathcal{D}_t$ such that for each non-empty equivalence class of $\sim_t$ in $\mathcal{D}_t$, there is exactly one such coloring in $\mathcal{C}_t$. The subfamily $\mathcal{C}_t$ is easy to compute. Initially, we set $\mathcal{C}_t := \mathcal{D}_t$, and while there are two colorings in $\mathcal{C}_t$ which are equivalent, delete one of them and repeat this step.

Forget Node. Suppose t is a forget node with a child t' such that $X_t = X_{t'} \setminus \{w\}$ for some $w \in X_{t'}$. Notice that here $G_t = G_{t'}$ and we have already computed $\mathcal{C}_{t'}$. But $X_t = X_{t'} \setminus \{w\}$. We want each of the colorings in $\mathcal{C}_t$ to have the following property. If a color is used only to color the edges from $E(G_t) \setminus E_{G_t}(X_t)$, then that color should be from $\{a_1, \ldots, a_{qn}\}$. For each coloring $f \in \mathcal{C}_{t'}$, we construct a coloring f' as follows. If a color $c \in [qk]$ is used to color an edge in $E(G_t) \setminus E_{G_t}(X_t)$ and not used to color any edge in $E_{G_t}(X_t)$, then recolor those edges with an unused color from $\{a_1, \ldots, a_{qn}\}$. The set $\mathcal{C}_t$ is the collection of such colorings f'. Clearly $|\mathcal{C}_t| = |\mathcal{C}_{t'}|$.

Join Node. Let t_1 and t_2 be the children of t. In this case, we have $X_t = X_{t_1} = X_{t_2}$ and G_t is the union of G_{t_1} and G_{t_2}. That is, $V(G_t) = V(G_{t_1}) \cup V(G_{t_2})$ and $E(G_t) = E(G_{t_1}) \cup E(G_{t_2})$. Also, we have already computed $\mathcal{C}_{t_1}$ and $\mathcal{C}_{t_2}$. We want to construct $\mathcal{C}_t$ by combining functions from $\mathcal{C}_{t_1}$ and $\mathcal{C}_{t_2}$. Initially we set $\mathcal{D}_t := \emptyset$. For a function $f_1 \in \mathcal{C}_{t_1}$ and a function $f_2 \in \mathcal{C}_{t_2}$, we construct an edge coloring of G_t, which is described below. Let $A_j = f_j(E_{G_{t_j}}) \cap \{a_1, \ldots, a_{nq}\}$ and $\ell_j = |A_j|$, for all $j \in \{1, 2\}$. Let $\beta_1 : A_1 \rightarrow \{a_1, \ldots, a_{\ell_1}\}$ and $\beta_2 : A_2 \rightarrow \{a_{\ell_1+1}, \ldots, a_{\ell_1+\ell_2}\}$ be two arbitrary fixed bijections. Notice that since f_1 and f_2 are edge q-colorings from $\mathcal{C}_{t_1} \subseteq \mathcal{S}_{t_1}$ and $\mathcal{C}_{t_2} \subseteq \mathcal{S}_{t_2}$, respectively, and $(V_{t_1} \cap V_{t_2}) \setminus X_t = \emptyset$, we have that $\ell_1 + \ell_2 \leq nq$, and hence β_1 and β_2 are well defined. Now, for all $j \in \{1, 2\}$, $f'_j : V_{t_j} \rightarrow [qk] \cup \{a_1, \ldots, a_{nq}\}$ be the edge q-colorings defined as follows: $f'_j(e) = f_j(e)$, if $f_j(e) \in [qk]$, and $f'_j(e) = \beta_j(f_j(e))$ if $f_j(e) \in \{a_1, \ldots, a_{nq}\}$. It is easy to see that $f_j \sim_{t_j} f'_j$ and f'_j is indeed an edge q-coloring of G_{t_j}. Moreover $f'_1(E(G_{t_1})) \cap f'_2(E(G_{t_2})) \subseteq [qk]$. Now define an edge coloring $f'_1 \oplus f'_2$ of G_t as follows.

$$f'_1 \oplus f'_2(e) = \begin{cases} f'_1(e), & \text{if } e \in E(G_{t_1}) \\ f'_2(e), & \text{otherwise, i.e., } e \in E(G_{t_2}) \setminus E(G_{t_1}) \end{cases}$$

Notice that each edge in G_t gets exactly one color in $f'_1 \oplus f'_2$, and hence $f'_1 \oplus f'_2$ is edge coloring of G_t. If $f'_1 \oplus f'_2 \in \mathcal{S}_t$, then we include $f'_1 \oplus f'_2$ in $\mathcal{D}_t$. Notice that when $f'_1 \oplus f'_2 \in \mathcal{S}_t$, $f'_1 \oplus f'_2$ is an edge q-coloring of G_t. Finally, construct a minimal subfamily $\mathcal{C}_t \subseteq \mathcal{D}_t$ such that for each non-empty equivalence class of $\sim_t$ in $\mathcal{D}_t$, there is exactly one such coloring in $\mathcal{C}_t$.

This completes the construction of $\mathcal{C}_t$. Recall that r is the root of T. Finally, we output a coloring from $\mathcal{C}_r$ that uses maximum number of colors.

Correctness. For the correctness proof, it is enough to prove the following statement. For any maximum edge q-coloring h of G, any node $t \in V(T)$, and any injective function $\beta : h(E(G)) \to [qk] \cup \{a_1, \ldots, a_{nq}\}$ such that $\beta \circ h(E_{G_t}(X_t)) \subseteq [qk]$, there is an edge q-coloring $f \in \mathcal{C}_t$ such that $f \sim_t (\beta \circ h)|_{E(G_t)}$. We prove the statement using mathematical induction, where the base case is when t is a leaf node. The base case is trivially true, because $V(G_t) = \emptyset$ for a leaf node t.

Consider the case when t is labelled as an introduce node. Let t' be the child of t and $X_{t'} = X_t \setminus \{v\}$. Let h be a maximum edge q-coloring of G and $\beta : h(E(G)) \to [qk] \cup \{a_1, \ldots, a_{nq}\}$ be an injective function such that $(\beta \circ h)(E_{G_t}(X_t)) \subseteq [qk]$. Let $f = (\beta \circ h)|_{E(G_{t'})}$ and $g = (\beta \circ h)|_{E_{G_t}(v)}$. By induction hypothesis, there is $f' \in \mathcal{C}_{t'}$ with $f' \sim_{t'} f = (\beta \circ h)|_{E(G_{t'})}$. Then, $f' \uplus g \in \mathcal{D}_t$ and there is a function $f'' \in \mathcal{C}_t$ such that $f'' \sim_t f' \uplus g$. Since, $f' \sim_{t'} f$, we get that $f'' \sim_t (\beta \circ h)|_{E(G_t)}$.

Consider the case when t is labelled as forget node and t' be the child of t. Notice that in this case for any coloring f in $\mathcal{C}_{t'}$, we changed some colors that are not used to color the edges in $E_{G_t}(X_t)$ to some other unused colors. Hence, the proof is simple and we omit here.

Consider the case when t is labelled as a join node. Let t_1 and t_2 be the children of t. Here, we have $X_t = X_{t_1} = X_{t_2}$. Let h be a maximum edge q-coloring of G and $\beta : h(E(G)) \to [qk] \cup \{a_1, \ldots, a_{nq}\}$ be an injective function such that $(\beta \circ h)(E_{G_t}(X_t)) \subseteq [qk]$. Let $g_1 = (\beta \circ h)|_{E(G_{t_1})}$ and $g_2 = (\beta \circ h)|_{E(G_{t_2})}$. By induction hypothesis, there are function $f_1 \in \mathcal{C}_{t_1}$ and $f_2 \in \mathcal{C}_{t_2}$ such that $f_1 \sim_{t_1} g_1$ and $f_2 \sim_{t_2} g_2$. Recall the functions f_1' and f_2' constructed in the algorithm. Since $g_1 \oplus g_2 = g_1 \cup g_2$ is an edge q-coloring of G_t, $f_1 \sim_{t_1} g_1$ and $f_2 \sim_{t_2} g_2$, we get that $f_1' \oplus f_2'$ is an edge q-coloring of G_t. Moreover, by the construction of f_1' and f_2', the number of colors used by $f_1' \oplus f_2'$ is same as the number of colors used by $g_1 \cup g_2$. Also, since $f_1 \sim_{t_1} g_1$ and $f_2 \sim_{t_2} g_2$, we get that $(f_1' \oplus f_2') \sim_t (g_1 \cup g_2) = (\beta \circ h)|_{E(G_t)}$. This completes the correctness proof.

Runtime analysis. Lemma 8 implies that for all $t \in V(T)$, $|\mathcal{C}_t| \leq \binom{qk}{q}^k \cdot q^{k+1} \cdot n$. There are $O(kn)$ nodes in T and the bottleneck in the computation is the computation of $\mathcal{C}_t$ for a join node. This running time is upper bounded by $O(|\mathcal{C}_{t_1}| \cdot |\mathcal{C}_{t_2}| \cdot n)$, where t_1 and t_2 are the children of t. This is upper bounded by $2^{O(kq \log kq)} n^3$. Thus the total running time is upper bounded by $2^{O(kq \log kq)} n^4$.

Improving the running time. Recall the definition of $\sim_t$. Two functions g_1 and g_2 are equivalent under $\sim_t$, if the following conditions hold.

(i) $|g_1(E(G_t))| = |g_2(E(G_t))|$. That is, the number of colors used by both g_1 and g_2 are the same.

(ii) For all $u \in X_t$, $g_1(E_{G_t}(u)) = g_2(E_{G_t}(u))$. That is, the colors seen by the edges incident on u are the same in both the colorings g_1 and g_2.

Instead of this, we may define the equivalence only when condition (ii) is satisfied. Then, in the computation we may store one from the equivalence class that uses maximum number of colors. This will reduce the number of equivalence classes to $\binom{qk}{q}^k \cdot q^k$ and thus the total running time to $2^{O(kq \log kq)} n$.

4 Hardness result

A *matching* M in a graph G is a collection of pairwise vertex disjoint edges, and the *size* of M is the number of edges in M. A *maximum matching* is a matching of the largest size. We shall use $\gamma(G)$ to denote the size of a maximum matching in G. It is known that every graph G has an edge 2-coloring of size at least $\gamma(G)$ as assigning every edge in a maximum matching a distinct color and the rest of the edges another new color is indeed a valid edge 2-coloring. We thus consider the following above-guarantee version of the maximum edge 2-coloring problem.

Above-Guarantee Edge 2-coloring	**Parameter:** k
Input: An undirected graph G and $k \in \mathbb{N}$.	
Question: Does G have an edge 2-coloring of size $\gamma(G) + k$?	

An *independent set* I in a graph G is a subset of its vertices such that no two vertices in I are adjacent to each other in G. The *size* of I is its cardinality. A *maximum independent set* in a graph G is an independent set of largest size. We shall use $\alpha(G)$ to denote the size of a largest independent set in G. It is known [5] that the following problem on whether a graph G has an independent set of size ℓ parameterized by ℓ is W[1]-hard.

Independent Set	**Parameter:** ℓ
Input: An undirected graph G and $\ell \in \mathbb{N}$	
Question: Does G have an independent set of size ℓ?	

In this section, we prove that the Above-Guarantee Edge 2-coloring problem is W[1]-hard by giving an parameterized reduction from the Independent Set problem to the former.

4.1 Construction

Let (H, ℓ) be an instance of the Independent Set problem. Let $V(H) = \{1, \dots, n\}$. From H we construct a bipartite graph G as described below:

1. For each vertex $i \in V(H)$, we have an edge $u_i u_i'$ in G. Let $U = \{u_i : i \in V(H)\} \cup \{u_i' : i \in V(H)\}$.
2. For each edge $ij \in E(H)$, we have a vertex $x_{i,j}$ that is adjacent to u_i and u_j. Let $X = \{x_{i,j} : ij \in E(H)\}$.
3. Finally, we have an edge ww' in G with w adjacent to every vertex in X. Let $W = \{w, w'\}$.

See Figure 1 for an example of the construction of the graph G from $H = K_4$. Here, $V(H) = [4]$ and $E(H) = \{\{1,2\}, \{1,3\}, \{1,4\}, \{2,3\}, \{2,4\}\})$.

Thus, $V(G) = U \cup X \cup W$, and the set of edges of G is as defined above. In the rest of this section, we shall use U-X *edges* to denote the set of edges having one endpoint in U and the other in X. In a similar way, we define X-W edges. Finally, for any vertex v in G, we shall use v-U (resp., v-X, v-W) edges to denote the set of edges from v to U (resp., X, W).

4.2 The proof

Throughout this section, we assume that (i) H is a graph on n vertices, and (ii) G is the graph constructed from H as described in Section 4.1.

▶ **Proposition 9.** $\gamma(G) = n + 1$.

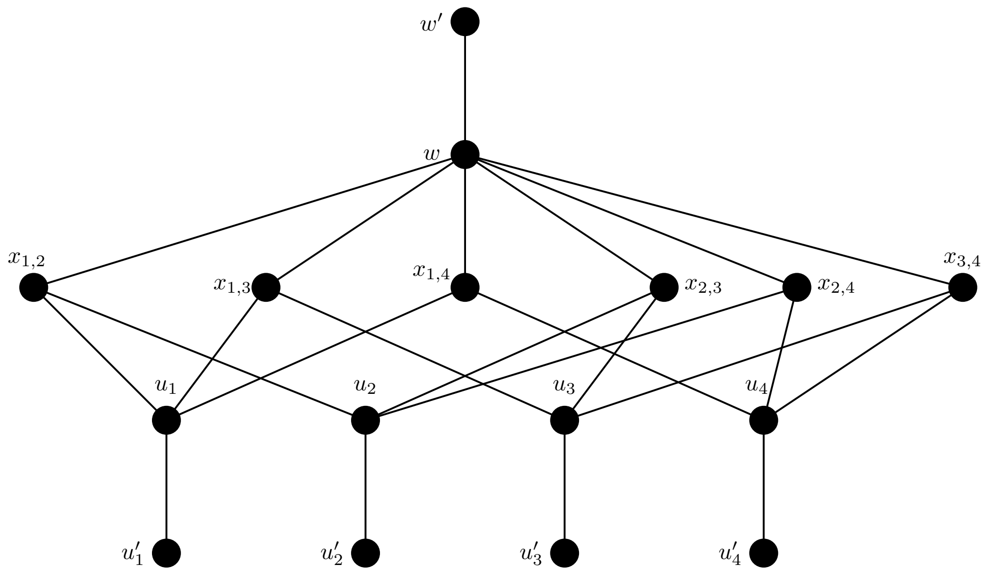

Figure 1 Graph G constructed from $H = K_4$.

Proof. The set $M = \{u_iu'_i \,:\, i \in [n]\} \cup \{ww'\}$ is a matching of size $n+1$. Thus, $\gamma(G) \geq n+1$. To show that $\gamma(G) \leq n+1$, consider any matching A of G. We construct a matching A' out of A with $|A'| = |A|$ below. If A contains any w-X edge, replace it with ww' in A'. If A contains any $x_{i,j}u_i$ edge, replace it with $u_iu'_i$ in A'. Thus, every edge in A' is a pendant edge of G. Since the number of pendant edges in G equals $n+1$, we have $|A| = |A'| \leq n+1$. ◀

▶ **Lemma 10.** *Given any edge 2-coloring f of G, one can obtain another edge 2-coloring f' of G such that*
1. *Every w-X edge in G has color c_F under f',*
2. *For every u_i, all u_i-X edges are of the same color under f'.*
3. *f' uses the same number of colors as f*

Proof. Below, we construct the coloring f' from f. Without loss of generality, assume that c_F is a color seen by the vertex w under f. If w sees only one color under f, then do nothing. Suppose w sees two colors, say c and c_F. In that case, replace every occurrence of c in the given coloring of G with c_F and finally assign the color c to the edge ww'. The resultant coloring is a valid edge 2-coloring as every vertex continues to see the same number of colors. Further, we have managed to satisfy Condition 1. The size of the new coloring obtained is the same as that of f. Now, consider each vertex u_i, $1 \leq i \leq n$, individually. If u_i sees only one color under f, then do nothing.

Suppose u_i sees two colors. In that case, (i) if c_F and, without loss of generality, c are the two colors seen by u_i, then replace every occurrence of c in the given coloring of G with c_F and finally assign the color c to the edge $u_iu'_i$, or (ii) if, without loss of generality, c and c' are the two colors seen by u_i, then replace every occurrence of say c' in the given coloring with c and finally assign the color c' to the edge $u_iu'_i$. Let us call the resultant coloring f'. Note that f' satisfies Condition 2. Lastly, note that the size of f' is the same as that of f. ◀

▶ **Lemma 11.** *Any maximum edge 2-coloring of G is of size $n + \alpha(H) + 2$.*

Proof. Assign color c_i to every $u_iu'_i$ edge, color c_0 to the ww' edge, and color c_F to every w-X edge. Let $S \subseteq V(H)$ be a maximum independent set in H. For each $i \in S$, assign the color c'_i to every u_i-X edge. For all the remaining X-U edges, assign the color c_F. Note that in the above coloring, (i) the vertex w sees the colors c_0 and c_F, (ii) every $u_i \in S$ sees colors c_i and c'_i, and (iii) every $u_i \in U \setminus S$ sees colors c_F and c_i. Now consider a vertex $x_{i,j} \in X$. Since S is an independent set in H, S won't contain both i and j. If S contains neither, then $x_{i,j}$ sees only color c_F. Without loss of generality, assume $i \in S$. Then, $x_{i,j}$ sees the colors c'_i and c_F. Thus, the above coloring is a valid edge 2-coloring. Note that the size of the above coloring is $n + \alpha(H) + 2$. We have thus shown that the size of a maximum edge 2-coloring of G is at least $n + \alpha(H) + 2$.

We now show that any edge 2-coloring of G is of size at most $n + \alpha(H) + 2$. Let f be an edge 2-coloring of G. Apply Lemma 10 to obtain the coloring f' from f. Let $P_{f'}$ be the set of colors seen by the pendant vertices of G under f'. Since G has $n + 1$ pendant vertices, $|P_{f'}| \leq n + 1$. We know that, under f', every w-X edge is assigned the color c_F, and for every u_i, all u_i-X edges are of the same color. Let $R_{f'}$ denote the set of all colors used by the coloring f' that are not present in $P_{f'} \cup \{c_F\}$. Observe that every color in $R_{f'}$ is used to color U-X edges (as the color of every other edge is present in $P_{f'} \cup \{c_F\}$). Let $U_R \subseteq U$ be a set of size $r := |R_{f'}|$ such that (i) for any two distinct $u_i, u_j \in U_R$, the color of u_i-X edges is different from the color of u_j-X edges, and (ii) the set of colors used to color the U_R-X edges is equal to $R_{f'}$. Without loss of generality, assume $U_R = \{u_1, \ldots, u_r\}$. We claim that the set $\{1, \ldots, r\}$ is an independent set in H. Suppose not. Then $ij \in E(H)$, for some $1 \leq i < j \leq r$. However, this would mean that the vertex $x_{i,j}$ sees three distinct colors (on its edges $x_{i,j}u_i, x_{i,j}u_j$, and $x_{i,j}w$) which is a contradiction to the fact that f' is a valid edge 2-coloring. Hence, $\{1, \ldots, r\}$ is an independent set in H and therefore, $r \leq \alpha(H)$. The set of colors used by f' is $P_{f'} \cup \{c_F\} \cup R_{f'}$ which is at most $(n+1)+(1)+(\alpha(H)) = n+\alpha(H)+2$. ◀

▶ **Theorem 12.** *Above-Guarantee Edge 2-coloring is W[1]-hard.*

Proof. Let (H, ℓ) be an instance of Independent Set. We construct G from H as described in Section 4.1. Note that this construction can be done in $O(n^2)$ time. Let $k = \ell + 1$. We claim that $(H, \ell) \in$ Independent Set if and only $(G, k) \in$ Above-Guarantee Edge 2-coloring. If H has an independent set of size ℓ, then, by Lemma 11, G has an edge 2-coloring of size $n + \alpha(H) + 2 \geq \gamma(G) + k$ (because $\alpha(H) \geq \ell$ and from Proposition 9, we have $\gamma(G) = n + 1$). From such a coloring, it is easy to obtain a valid edge 2-coloring of size exactly $\gamma(G) + k$ (see Proposition 1 in [10] for a proof of this statement). To prove the converse, suppose H does not have any independent set of size ℓ. Then, by Lemma 11, G has no edge 2-coloring of size $n + \ell + 2 = (n + 1) + (\ell + 1) = \gamma(G) + k$. This completes the reduction. Note that this is a parameterized reduction from Independent Set to Above-Guarantee Edge 2-coloring as it satisfies all the three conditions of Definition 5. Since Independent Set is known to be W[1]-hard, we have the theorem. ◀

5 Concluding remarks

In this work, we resolve an open question of Goyal et al. [10]. Further, we give an FPT algorithm of running time $2^{O(\mathsf{tw} \cdot q \log(\mathsf{tw} \cdot q))}n$ for Maximum Edge q-Coloring, where tw is the treewidth of the input graph. It is natural to ask if this running time is optimal. We would like to mention that as a corollary of a result of Goyal et al. [10] (as well as our result above), one gets an FPT algorithm of running time $2^{O(\mathsf{vc} \log(\mathsf{vc}))}n$ for Maximum Edge 2-Coloring, where vc is the vertex cover number of the input graph. It would be interesting to obtain a single exponential FPT algorithm and a polynomial kernel even when the parameter is vertex cover number.

References

1 Anna Adamaszek and Alexandru Popa. Approximation and hardness results for the maximum edge q-coloring problem. *Journal of Discrete Algorithms*, 38:1–8, 2016. doi:10.1016/J.JDA.2016.09.003.

2 L. Sunil Chandran, Talha Hashim, Dalu Jacob, Rogers Mathew, Deepak Rajendraprasad, and Nitin Singh. New bounds on the anti-ramsey numbers of star graphs via maximum edge q-coloring. *Discret. Math.*, 347(4):113894, 2024. doi:10.1016/J.DISC.2024.113894.

3 L. Sunil Chandran, Abhiruk Lahiri, and Nitin Singh. Improved approximation for maximum edge colouring problem. *Discret. Appl. Math.*, 319:42–52, 2022. doi:10.1016/J.DAM.2021.05.017.

4 B Courcelle. The monadic second-order theory of graphs i: Recognizable sets of finite graphs. *Information and Computation*, 1990.

5 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 5(4). Springer, 2015. doi:10.1007/978-3-319-21275-3.

6 Zdeněk Dvořák and Abhiruk Lahiri. Maximum edge colouring problem on graphs that exclude a fixed minor. In Daniël Paulusma and Bernard Ries, editors, *Graph-Theoretic Concepts in Computer Science*, pages 291–304, Cham, 2023. Springer Nature Switzerland.

7 Pál Erdős, Miklós Simonovits, and Vera T Sós. Anti-ramsey theorems. In *Infinite and finite sets: To Paul Erdős on his 60th birthday*. North-Holland Publishing Company, 1975.

8 Wangsen Feng, Hanpin Wang, et al. Approximation algorithm for maximum edge coloring. *Theoretical computer science*, 410(11):1022–1029, 2009. doi:10.1016/J.TCS.2008.10.035.

9 Shinya Fujita, Colton Magnant, and Kenta Ozeki. Rainbow generalizations of ramsey theory: a survey. *Graphs and Combinatorics*, 26:1–30, 2010. doi:10.1007/S00373-010-0891-3.

10 Prachi Goyal, Vikram Kamat, and Neeldhara Misra. On the parameterized complexity of the maximum edge 2-coloring problem. In *Mathematical Foundations of Computer Science 2013: 38th International Symposium, MFCS 2013, Klosterneuburg, Austria, August 26-30, 2013. Proceedings 38*, pages 492–503. Springer, 2013. doi:10.1007/978-3-642-40313-2_44.

11 Ashish Raniwala and Tzi-cker Chiueh. Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network. In *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 3, pages 2223–2234. IEEE, 2005. doi:10.1109/INFCOM.2005.1498497.

12 Ashish Raniwala, Kartik Gopalan, and Tzi-cker Chiueh. Centralized channel assignment and routing algorithms for multi-channel wireless mesh networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 8(2):50–65, 2004. doi:10.1145/997122.997130.