Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 28, no. 1, pp. 129–147 (2024)
DOI: 10.7155/jgaa.v28i1.2931

The maximum 2-edge-colorable subgraph problem and its fixed-parameter tractability

Vahan Mkrtchyan

Gran Sasso Science Institute,
School of Advanced Studies, L'Aquila, Italy

Submitted: November 2023	Accepted: April 2024	Published: May 2024
Article type: Regular paper	Communicated by: Md. Saidur Rahman	

Abstract. A k-edge-coloring of a graph is an assignment of colors $\{1, ..., k\}$ to edges of the graph such that adjacent edges receive different colors. In the maximum k-edge-colorable subgraph problem we are given a graph and an integer k, the goal is to find a k-edge-colorable subgraph with maximum number of edges together with its k-edge-coloring. In this paper, we consider the maximum 2-edge-colorable subgraph problem and present some results that deal with the fixed-parameter tractability of this problem.

Keywords: Edge-coloring; maximum 2-edge-colorable subgraph; exact algorithm; fixed-parameter tractability

1 Introduction

In this paper, we consider finite, undirected graphs that do not contain loops or parallel edges. The set of vertices and edges of a graph G is denoted by V and E, respectively. $d_G(u)$ denotes the degree of a vertex u of G. Let $\delta(G)$ and $\Delta(G)$ be the minimum and maximum degree of vertices of G. Let $rad(G)$ and $diam(G)$ be the radius and diameter of G.

A matching in a graph G is a subset of E such that no vertex of G is incident to two edges from it. A maximum matching is a matching that contains the largest possible number of edges.

For $k \geq 0$, a graph G is k-edge colorable, if its edges can be assigned colors from a set of k colors so that adjacent edges receive different colors. The smallest k, such that G is k-edge-colorable is called chromatic index of G and is denoted by $\chi'(G)$. The classical theorem of Shannon states that for any multi-graph G, $\Delta(G) \leq \chi'(G) \leq \left\lfloor \frac{3\Delta(G)}{2} \right\rfloor$ [31, 35]. Moreover, the classical theorem of Vizing states that for any multi-graph G, $\Delta(G) \leq \chi'(G) \leq \Delta(G) + \mu(G)$ [35, 36]. Here $\mu(G)$

The work of the author has been partially supported by the Italian MIUR PRIN 2017 Project ALGADIMAR "Algorithms, Games, and Digital Markets."

E-mail address: vahan.mkrtchyan@gssi.it (Vahan Mkrtchyan)

denotes the maximum multiplicity of an edge of G. A multi-graph G is class I, if $\chi'(G) = \Delta(G)$, otherwise it is class II.

If $k < \chi'(G)$, we cannot color all edges of G with k colors. Therefore, it is natural to investigate the maximum number of edges that one can color with k colors. A subgraph H of G is called maximum k-edge-colorable, if H is k-edge-colorable and contains maximum number of edges among all k-edge-colorable subgraphs of G. For $k \geq 0$ and a graph G let

$$\nu_k(G) = \max\{|E(H)| : H \text{ is a } k\text{-edge-colorable subgraph of } G\}.$$

Clearly, a k-edge-colorable subgraph is maximum if it contains exactly $\nu_k(G)$ edges. Observe that $\nu_1(G)$ is the size of a maximum matching of G. We will shorten this notation to $\nu(G)$.

One may think if we have a maximum k-edge-colorable subgraph of a graph, then by adding some edges to it, we can get a maximum $(k+1)$-edge-colorable subgraph. The tree from Figure 1 shows that this is not the case. It has a unique perfect matching, a matching covering all vertices of the graph, which contains the edge joining the two degree three vertices. However, a maximum 2-edge-colorable subgraph of it contains all its eight edges except the edge joining the two degree three vertices.

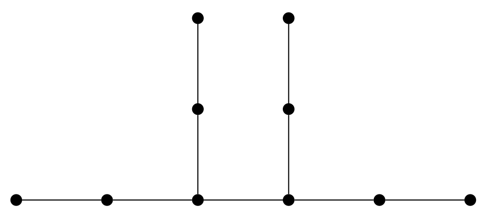

Figure 1: A tree in which the largest matching is not a subset of a maximum 2-edge-colorable subgraph.

In this paper, we deal with the exact solvability of the maximum k-edge-colorable subgraph problem. Its precise formulation is the following:

Problem 1 *Given a graph G and an integer k, find a k-edge-colorable subgraph with maximum number of edges together with its k-edge-coloring.*

We investigate this problem from the perspective of fixed-parameter tractability. Recall that an algorithmic problem Π is fixed-parameter tractable with respect to a parameter θ, if there is an exact algorithm solving Π, whose running-time is $f(\theta) \cdot poly(size)$. Here f is some (computable) function of θ, $size$ is the length of the input and $poly$ is a polynomial function. A (parameterized) problem is paraNP-hard, if it remains NP-hard even when the parameter is constant.

In this paper, we focus on the maximum 2-edge-colorable subgraph problem which is the restriction of the problem to the case $k = 2$. We present some results that deal with the fixed-parameter tractability of this problem with respect to various graph-theoretic parameters. Parameterized complexity theory forms a very active research area. The ideas and concepts in it allow us to deepen our understanding of the hardness of algorithmic problems. They strengthen the results that the classical complexity theory provides. The main idea presented in this area is that the algorithms solving our algorithmic problems should depend not only on the size of the input (like we have in classical complexity theory), but also one or more parameters of the input. More details on parameterized complexity theory can be found in [11].

The main contributions of this paper are the following:

- ParaNP-hardness of the problem with respect to the radius, diameter and $|V| - MaxLeaf(G)$,
- Fixed-parameter tractability of our problem with respect to $|V| - \delta$, the dimension of the cycle space and $MaxLeaf(G)$-the maximum number of leaves in a spanning tree of G.
- Fixed-parameter tractability of two related problems with respect to the budget ℓ.

For the notions, facts and concepts that are not explained in the paper the reader is referred to [11, 15, 38].

The history behind [5] and preprints in arXiv. During the review of the present paper, the editor handling paper and two referees have requested to clarify the history behind the paper [5] and the preprints in arXiv [4]. The preprints in arXiv, are just unpublished versions of our results, with the sole exception of results presented in [5]. This paper presents current author's contributions. The only result that was obtained by our combined efforts is a previous version of Lemma 3. We agreed that the author will publish the current version of the lemma where we have edge weights, and the other author will publish his even more general version later.

2 Motivation and related work

There are many papers where the ratio $\frac{\nu_k(G)}{|E|}$ has been investigated. [7, 18, 27, 28, 37] prove lower bounds for this ratio in case of regular graphs and $k = 1$. For regular graphs of high girth the bounds are improved in [14]. Albertson and Haas investigated the problem in [1, 2] when G is a cubic graph. See also [24], where it is shown that for every cubic multigraph G, $\nu_2(G) \geq \frac{4}{5}|V|$ and $\nu_3(G) \geq \frac{7}{6}|V|$. Moreover, [6] proves that for any cubic multigraph G, $\nu_2(G) + \nu_3(G) \geq 2|V|$, and in [24, 25] Mkrtchyan et al. showed that for any cubic multigraph G, $\nu_2(G) \leq \frac{|V|+2\nu_3(G)}{4}$. Finally, in [21], it is shown that the sequence ν_k is convex in the class of bipartite multigraphs. Rizzi in [29] has shown that the above-mentioned $\frac{7}{6}|V|$ bound for cubic multigraphs can be significantly improved for graphs (without parallel edges) G of maximum degree three. For such graphs G, it can be shown that $\nu_3(G) \geq \frac{6}{7} \cdot |E|$ [29].

Bridgeless cubic graphs that are not 3-edge-colorable are called snarks [9], and the ratio for snarks is investigated by Steffen in [33, 34]. This lower bound has also been investigated in the case when the graphs need not be cubic in [16, 20, 29]. Kosowski and Rizzi have investigated the problem from the algorithmic perspective [22, 29]. The problem of finding a maximum k-edge-colorable graph in an input graph is NP-complete for every fixed $k \geq 2$. For example, when G is cubic and $k = 2$, we have that $\nu_2(G) = |V|$ if and only if G contains two edge-disjoint perfect matchings. The latter condition is equivalent to saying that G is 3-edge-colorable, which is an NP-complete problem as Holyer has demonstrated in [19]. Thus, it is natural to investigate the (polynomial) approximability of the problem. In [13] for each $k \geq 2$ an approximation algorithm for the problem is presented. There for each fixed value of $k \geq 2$, algorithms are proved to have certain approximation ratios and these ratios are tending to 1 as k goes to infinity. In [22], two approximation algorithms for the maximum 2-edge-colorable subgraph and maximum 3-edge-colorable subgraph problems are presented whose performance ratios are $\frac{5}{6}$ and $\frac{4}{5}$, respectively. Finally, note that the results of [13] are improved for $k = 3, ..., 7$ in [20].

Some structural properties of maximum k-edge-colorable subgraphs of graphs are proved in [6, 26]. There it is shown that every set of disjoint cycles of a graph with $\Delta = \Delta(G) \geq 3$ can be extended to a maximum $\Delta(G)$-edge colorable subgraph. Moreover, there it is shown that any maximum $\Delta(G)$-edge colorable subgraph of a graph is always class I. Observe that this statement is not true when G is a multigraph. If one considers a triangle in which each edge is of multiplicity three, then the maximum degree in it is six. An example of a maximum 6-edge-colorable in this graph will be the triangle in which each edge is of multiplicity two. Observe that it has maximum degree four and chromatic index six. Thus, it is class II. Finally, in [26] it is shown that if G is a graph of girth (the length of the shortest cycle) $g \in \{2k, 2k+1\}$ $(k \geq 1)$ and H is a maximum $\Delta(G)$-edge colorable subgraph of G, then $\frac{|E(H)|}{|E|} \geq \frac{2k}{2k+1}$, The bound is best possible as there is an example attaining it. See the recent paper [8], where some new results about partitioning arbitrary multi-graphs into class I subgraphs are presented.

In [17] the k-edge-coloring problem is considered, which is formulated as follows:

Problem 2 *Given a graph G and an integer k, check whether G is k-edge-colorable.*

There it is shown that for each fixed k, the k-edge-coloring problem is fixed-parameter tractable with respect to the number of maximum degree vertices of the input graph. Observe that the maximum k-edge-colorable subgraph problem is harder than k-edge-coloring, as if we can construct a maximum k-edge-colorable subgraph H_k of the input graph G, then in order to see that whether G is k-edge-colorable, we just need to check whether $E(H_k) = E$. If one considers the edge-coloring problem, where for an input graph G, we need to find a $\chi'(G)$-edge-coloring of G, then in [23] it is stated that a major challenge in the area is to find an exact algorithm for this problem whose running-time is $2^{O(n)} = O(c^n)$. Observe that the maximum k-edge-colorable subgraph problem is harder than edge-coloring. If we are able to solve the maximum k-edge-colorable subgraph problem in time $O(f(size))$, then we can solve the Edge-Coloring problem in time $O(f(size)) \cdot \log(|V|)$. In order to see this, just observe that we can do a binary search on $k = 1, 2, ..., |V|$, solve the maximum k-edge-colorable problem and find an edge-coloring of G with the smallest number of colors. Here we used the fact that any graph G is $|V|$-edge-colorable. See [3] for some new results on this problem.

3 Some auxiliary results

In this section, we present some results that will be used in obtaining the main results of the paper. Below we assume that $\mathbb{N}$ is the set of natural numbers.

Lemma 1 *([30]) Let Π be an algorithmic problem, and let k_1 and k_2 be some parameters. Assume that there is a (computable) function $g : \mathbb{N} \to \mathbb{N}$, such that for any instance I of Π, we have $k_1(I) \leq g(k_2(I))$. Then if Π is FPT with respect to k_1, then it is FPT with respect to k_2.*

In [19], Holyer has shown that checking whether a cubic graph is 3-edge-colorable is an NP-complete problem. For a cubic graph G, let $r_3(G)$ be defined as:

$$r_3(G) = |E| - \nu_3(G).$$

This parameter is introduced and investigated in [34]. In particular, there it is observed there that $r_3(G) \neq 1$ for any cubic graph G. This means that $r_3(G)$ can be zero or at least two, and the 3-edge-coloring problem in cubic graphs amounts to deciding which of these two cases holds. For our purposes we will consider the following restriction of 3-edge-coloring problem in cubic graphs:

Problem 3 *For a fixed integer $l \geq 1$, consider a decision problem, whose input is a cubic graph G, in which $r_3(G)$ is from the set $\{0, l, l+1, l+2, ...\}$. The goal is to check whether G is 3-edge-colorable, that is, whether $r_3(G) = 0$.*

Lemma 2 *For each fixed $l \geq 1$, Problem 3 is NP-complete.*

Proof: The case when $l \leq 2$ corresponds to the usual 3-edge-coloring problem in cubic graphs. Thus, we can assume that $l \geq 3$. We reduce the 3-edge-coloring problem of cubic graphs to this problem. Let G be any cubic graph. Consider a cubic graph H obtained from l vertex disjoint copies of G. Observe that $|V(H)| = l \cdot |V|$, hence H can be constructed from G in linear time. Now, it is easy to see that G is 3-edge-colorable if and only if H is 3-edge-colorable. Moreover, $r_3(H) = l \cdot r_3(G)$. Hence, $r_3(H)$ is either zero or at least l. The proof is complete. □

We will also need the following result obtained in [24, 25]:

Theorem 1 *For any cubic graph G $\nu_2(G) \leq \frac{|V|+2\nu_3(G)}{4}$.*

4 Main results

In this section, we present the first part of our main results about the maximum 2-edge-colorable subgraph problem. If m is the number of edges of the input graph G, then clearly we can generate all 2^m subgraphs/subsets of E, and check each of them for 2-edge-colorability. In great contrast with k-edge-colorability with $k \geq 3$, checking 2-edge-colorability can be done in polynomial time. A subgraph F of G is 2-edge-colorable if and only if it has maximum degree at most two, and it contains no component that is an odd cycle. Clearly this can be checked in polynomial time. The running time of this trivial, brute-force algorithm is $O^*(2^m)$. We will refer to this algorithm as trivial or brute-force algorithm.

The first parameter with respect to which we will investigate our problem is the radius of the graph.

Theorem 2 *The maximum 2-edge-colorable subgraph problem is paraNP-hard with respect to the $rad(G)$.*

Proof: We present a reduction from Problem 3 with $l \geq 6$. By Lemma 2 it is NP-complete. Let us take an arbitrary cubic graph G with $r_3(G)$ either zero or at least l. Take a new vertex z, who is joined to every vertex of G. Let G' be the resulting graph (Figure 2).

Let us show that $\nu_2(G') \geq |V|$ if and only if G is 3-edge-colorable. Let G be a 3-edge-colorable. Then it admits a pair of edge-disjoint perfect matchings. Hence, these perfect matchings form a 2-edge-colorable subgraph in G'. Thus, $\nu_2(G') \geq |V|$. Now, assume that G is not 3-edge-colorable, hence $r_3(G) \geq l \geq 6$. By Theorem 1:

$$\nu_2(G') \leq 2 + \nu_2(G) \leq 2 + \frac{|V| + 2 \cdot \nu_3(G)}{4} = 2 + |V| - \frac{r_3(G)}{2} \leq |V| - 1,$$

since $r_3(G) \geq l \geq 6$. Hence, if $\nu_2(G') \geq |V|$, then G is 3-edge-colorable.

Observe that in graphs G' that we obtained from G, we have $rad(G') = 1$ (z is of distance one from any other vertex). Thus, checking whether $\nu_2(G') = |V|$ is an NP-complete problem even when the radius is one. Thus the problem is paraNP-hard with respect to the radius. The proof is complete. □

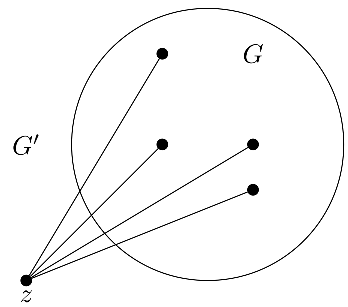

Figure 2: G' is obtained from G by adding a vertex z that is joined to every vertex of G.

Remark 1 *The maximum 2-edge-colorable subgraph problem is paraNP-hard with respect to the* $diam(G)$.

This follows from Theorem 2, Lemma 1 and the fact that in any graph G, we have

$$rad(G) \leq diam(G) \leq 2 \cdot rad(G).$$

In [17], it is shown that for each fixed k, the k-edge-coloring problem is FPT with respect to the number of maximum degree vertices of the input graph. As we have mentioned previously, the maximum k-edge-colorable subgraph problem is harder than k-edge-coloring. Thus, one can try to parameterize the latter with respect to the number of vertices of maximum degree. As the following theorem states, if $P \neq NP$, this is impossible.

Theorem 3 *The maximum 2-edge-colorable subgraph problem is paraNP-hard with respect to the number of maximum-degree vertices.*

Proof: Consider the class of graphs G' from the proof of Theorem 2. Observe that if G is the complete graph on four vertices then G' has five vertices of degree four, which have maximum degree in G'. On the other hand, if $|V| \geq 6$, then z is the only vertex of maximum degree. Thus, the problem is NP-hard when the number of maximum degree vertices is at most five. The proof is complete. □

Remark 2 *Observe that in the above proof, there is no need for us to join z to all the vertices of G. Since G is cubic we can join z to five vertices of G. This will lead to the graph G', where z is the only vertex of degree five, which is maximum. All other vertices are of degree four or three. Thus, the problem remains hard even when the number of maximum degree vertices is one and the maximum degree is five.*

Holyer's result [19] implies that it is NP-hard to find a maximum 2-edge-colorable subgraph in cubic graphs. Thus, the maximum 2-edge-colorable subgraph problem is paraNP-hard with respect to $\Delta(G)$ and $\delta(G)$. Moreover, in the proof of Theorem 2, we have $|V(G')| = |V| + 1$ and $\Delta(G') = d(z) = |V|$, hence $|V(G')| - \Delta(G') = 1$ in these graphs G'. Thus, one can say that it is paraNP-hard with respect to $|V| - \Delta$, too. On the positive side, it turns out that

Proposition 1 *The maximum 2-edge-colorable subgraph problem is FPT with respect to* $|V| - \delta$.

Proof: Let G be any graph. If $|V| - \delta(G) \geq \frac{|V|}{2}$, then

$$|V| \leq 2 \cdot (|V| - \delta(G)).$$

Thus,

$$|E| \leq |V|^2 \leq 4 \cdot (|V| - \delta(G))^2.$$

Now, if we run the trivial algorithm, its running-time will depend solely on $|V| - \delta$, as we have bounded the number of edges in terms of it. On the other hand, if $|V| - \delta(G) \leq \frac{|V|}{2}$, then

$$\delta(G) \geq \frac{|V|}{2}.$$

Thus, by Ore's classical theorem [38], G has a Hamiltonian cycle C. Now, if $|V|$ is even, then C is a 2-edge-colorable subgraph in G. Since in any graph G, $\nu_2(G) \leq |V|$, we have that C is a maximum 2-edge-colorable subgraph in G. On the other hand, if $|V|$ is odd, then any matching in G has at most $\frac{|V|-1}{2}$ edges, hence $\nu_2(G) \leq |V| - 1$. Now, if we remove any edge from C, then the resulting Hamiltonian path will be a 2-edge-colorable subgraph with $|V| - 1$ edges. Hence it will be a maximum 2-edge-colorable subgraph in G. The proof is complete. □

Remark 3 *Let us note that the proof of Ore's theorem represents a polynomial time algorithm which actually finds the Hamiltonian cycle. Thus, in the second case of the previous proof, the algorithm will run in polynomial time.*

Observe that in any graph G, we have the following relationship among vertex, edge connectivity and minimum degree:

$$\kappa(G) \leq \kappa'(G) \leq \delta(G).$$

Since, the maximum 2-edge-colorable subgraph problem is paraNP-hard with respect to δ, Lemma 1 implies that the problem is paraNP-hard with respect to $\kappa(G)$ and $\kappa'(G)$. Moreover, since in any graph

$$|V| - \delta(G) \leq |V| - \kappa'(G) \leq |V| - \kappa(G),$$

Proposition 1 and Lemma 1 imply that the problem is FPT with respect to $|V| - \kappa(G)$ and $|V| - \kappa'(G)$.

It is a simple corollary of Courcelle's theorem that the maximum 2-edge-colorable subgraph problem is fixed-parameter tractable with respect to treewidth. Since the latter is bounded by $\nu(G)$ [30], we have the following:

Corollary 1 *The maximum 2-edge-colorable subgraph problem is FPT with respect to $\nu(G)$.*

Let $\alpha'(G)$ be the smallest number of edges of G such that any vertex of G is incident to at least one of these edges. By the classical Gallai theorem [38], we have that if the graph G has no isolated vertices, then

$$\nu(G) + \alpha'(G) = |V|.$$

Since $\nu(G) \leq \frac{|V|}{2}$, we have

$$\nu(G) \leq \frac{|V|}{2} \leq \alpha'(G).$$

Thus, Corollary 1 and Lemma 1 imply that the maximum 2-edge-colorable subgraph problem is FPT with respect to $\alpha'(G)$. Observe that isolated vertices play no role in the maximum k-edge-colorable subgraph problem, thus we can assume that the input graph contains none of them.

Also, observe that the parameterization with respect to $\alpha'(G)$ can be interpreted as parameterization with respect to $|V|-\nu(G)$. One may wonder, whether we can strengthen this result, by showing that the maximum 2-edge-colorable subgraph problem is FPT with respect to $|V|-2\cdot\nu(G)$? The answer to this question is negative unless $P=NP$. If a cubic graph G is 3-edge-colorable, then it must be bridgeless. Thus, by Holyer's result the maximum 2-edge-colorable subgraph problem is NP-hard for bridgeless cubic graphs. By the classical Petersen theorem [38], bridgeless cubic graphs have a perfect matching. Thus, in this class we have $|V|-2\nu(G)=0$. Hence, the maximum 2-edge-colorable subgraph problem is paraNP-hard with respect to $|V|-2\nu(G)$.

One can consider the decision version of the maximum 2-edge-colorable subgraph problem, where for a given graph G and an integer t, one needs to check whether $\nu_2(G)\geq t$. It turns out that this problem is FPT with respect to t. In order to see this, just observe that if in the input graph G $\nu(G)\geq t$, then clearly $\nu_2(G)\geq\nu(G)\geq t$, hence the instance is a "yes" instance. On the other hand, if $\nu(G)\leq t$, then the FPT algorithm with respect to $\nu(G)$ (Corollary 1) will in fact be an FPT algorithm with respect to t (Lemma 1).

Below we will parameterize the maximum 2-edge-colorable subgraph problem with respect to the dimension of the cycle space of a graph. Before we start, we recall some concepts. Let $B=\{0,1\}$ be the binary field. If K is a subset of edges of a graph G, then we can consider an $|E|$-dimensional vector x_K whose coordinates are zero and one, and for any edge e of G, we have $e\in E'$, if and only if $x_K(e)=1$. Here $x_K(e)$ denotes the coordinate of x_K corresponding to e. The vector x_K is usually called the characteristic vector of K. Observe that the characteristic vectors of all subsets of E form a $|E|$-dimensional linear space over B. Now, let us consider the cycle space $C(G)$ of G, which is defined as the linear hull of all characteristic vectors that correspond to simple cycles of G. Clearly, the cycle space is a linear subspace of all characteristic vectors. A classical result in the area states that if G is any graph with d components, then the dimension of $C(G)$ is given by the following formula:

$$dim(C(G))=|E|-|V|+d.$$

An alternative way of looking at $dim(C(G))$ is the following: a subset F of edges of a graph G is called a feedback edge-set, if $G-F$ contains no cycles. In other words, any cycle of G contains an edge from F. It turns out that $dim(C(G))$ represents the size of a smallest feedback edge-set of G. Moreover, there is a polynomial time algorithm that finds such a subset of edges for any input graph G.

For our parameterization, we will require the following lemma that we will state in a very general form. It extends the approach of [5] to the case when edges have weights, and we are seeking a maximum weighted k-edge-colorable subgraph. If $H\subseteq G$ is a subgraph of G and $p:E(G)\to\mathbb{N}$ is an edge-weight function, then its p-weight is defined as

$$p(H)=p(E(H))=\sum_{e\in E(H)}p(e).$$

In the maximum weighted k-edge-colorable subgraph problem we are searching a k-edge-colorable subgraph H maximizing $p(H)$ together with the k-edge-coloring of H.

Lemma 3 *Let $k\geq 1$ and G be an edge-weighted forest with $p:E\to\mathbb{N}$. Suppose $W:V\to 2^{\{1,\dots,k\}}$ is a function that assigns each vertex u a subset $W(u)\subseteq\{1,...,k\}$ of admissible colors. Then, there is a $O((k+1)\cdot 2^{2k}\cdot|V|)$-time algorithm that finds a largest weighted k-edge-colorable subgraph (with respect to p) with the constraint that around every vertex v appear* **only** *colors from $\{0\}\cup W(v)$.*

Remark 4 *We explicitly note that around a vertex multiple edges of color 0 can appear, however at most one edge of colors 1, 2,..., k can appear.*

Remark 5 *Note that our lemma includes the case when W is given only for some subset of V, as for the vertices x outside the subset we can always define $W(x) = \{0, 1, ..., k\}$.*

Proof: Clearly, we can assume that G is a tree, as otherwise we can solve the problem on each component of the forest, and then take the union of these solutions.

Let $G(V, E)$ be an edge-weighted tree with $n = |V|$ vertices and $|E| = n - 1$ edges. Assume that $W: V \to 2^{\{0,1,\ldots,k\}}$ is an assignment of the available colors for each vertex in V. Let $q: \{0, 1, ..., k\} \to \{0, 1\}$ be a function that is equal to 0, if the input is the dummy color 0, and is 1, otherwise. Let r be the root of G, and let $G(u)$ be the subgraph of G induced by u and all the descendants of u in G.

Now, we describe a dynamic programming algorithm that finds a largest weighted k-edge-colorable subgraph (with respect to p) of $G = (V, E)$, that respects our constraints W. We call this problem $\mathcal{PR}$.

In the algorithm, we compute $f(u, A)$ for each vertex $u \in V$, and for every $A \subseteq W(u)$, which is equal to the optimum value of $\mathcal{PR}$ restricted to the subgraph $G(u)$, and with the additional constraint where the colors incident to u are those in A. Since we can always take the empty subgraph as a k-edge-colorable subgraph, if there is no solution, we set $f(u, A) = 0$. In particular, the algorithm starts from the leaves and goes up to the root r, and the optimal value of $\mathcal{PR}$ is the maximum $f(r, A)$, for every $A \subseteq W(r)$.

If u is a leaf, and A a specific color subset of $W(u)$, we proceed as follows: $f(u, A) = 0$, if $A = \emptyset$; and we set $f(u, A) = 0$ if $A \neq \emptyset$. In fact, since there are no edges in $G(u)$ when u is a leaf, there cannot be any color incident to u in $G(u)$.

For each internal vertex u, we suppose $A \neq \emptyset$, as it is always possible to use the dummy color 0.

If u is an internal vertex with t sons $\{v_1, \ldots, v_t\}$, we compute $f(u, A)$ for any subset $A \subseteq W(u)$ by using the values $f(v_i, A)$ for every u's son v_i. For every $i \in \{1, \ldots, t\}$, denote by V_i the set containing the vertices in the subgraph $G(u)$ minus the vertices in each subgraph $G(v_j)$, with $j \in \{i+1, \ldots, t\}$. Let $G(V_i)$ be the subgraph induced by the vertices in V_i. For every $i \in \{1, \ldots, t\}$ we compute $h(u, V_i, A)$, which equals the maximum value of $\mathcal{PR}$ restricted to the subgraph $G(V_i)$, and with the additional constraint that the colors incident to u are those in A. If the empty subgraph is the only solution, we set $h(u, V_i, A) = 0$. Notice that $h(u, V_t, A)$ is equivalent to $f(u, A)$. Now, we see how to compute $h(u, V_i, A)$ for every $i \in \{1, \ldots, t\}$, recursively.

If $i = 1$, there is only one edge incident to u in $G(V_1)$, i,.e., (u, v_1). So, we can set $A = \{\theta(u, v_1)\}$, where A contains only the color $\theta(u, v_1)$ from $W(u)$ assigned to (u, v_1). We compute $h(u, V_1, A)$ solving the following problem:

$$\begin{array}{ll} \max_{C \subseteq W(v_1)} & f(v_1, C) + q(\theta(u, v_1)) \cdot p((u, v_1)) \\ \text{subject to:} & C \cap \{\theta(u, v_1)\} \subseteq \{0\} \\ & C \subseteq W(v_1) \end{array} \tag{1}$$

In fact, for a specific color $\theta(u, v_1)$, we get the best value $f(v_1, C)$ for every $C \subseteq W(v_1)$ that is compatible with $\theta(u, v_1)$. The compatibility is guaranteed by the constraint that does not allow to choose a subset C (the problem's variable) that contains $\theta(u, v_1)$ if it is not the color 0.

If $i \geq 2$, we calculate $h(u, V_i, A)$ by using the values $h(u, V_{i-1}, B)$. We solve the following maximisation problem for every non empty subset $B \subseteq W(u)$ and any color $\theta(u, v_i) \in W(u)$ for the edge (u, v_i), such that $A = B \cup \{\theta(u, v_i)\}$, and $B \cap \{\theta(u, v_i)\} \subseteq \{0\}$.

$$\begin{array}{lll} \max_{B\subseteq W(u),C\subseteq W(v_i)} & h(u,V_{i-1},B)+f(v_i,C)+q(\theta(u,v_i))\cdot p((u,v_i)) & \\ \text{subject to} & C\cap\{\theta(u,v_i)\}\subseteq\{\mathit{0}\} & (2) \\ & C\subseteq W(v_i) & \end{array}$$

The idea is that, for every $B \subseteq W(u)$, and for every $\theta(u, v_i)$ that are compatible, we search the subset $C \subseteq W(v_i)$ (the problem's variable), compatible with $\theta(u, v_i)$, which maximises the weight of edges with true colors in $G(V_i)$, that is the objective function. In the objective function, $f(v_i, C) + q(\theta(u, v_i)) \cdot p((u, v_i))$ refers to the subgraph $G(v_i) \cup (u, v_i)$, while $h(u, V_{i-1}, B)$ is the value already computed.

The time complexity to compute $h(u, V_i, A)$ for all the t sons of an internal vertex u is $O((k+1) \cdot 2^{2k} \cdot t)$. In fact, for $t = 1$ we solve (1) for every $\theta(u, v_i) \in W(u)$, so at most $(k+1)$ times; at each of the t steps, with $t \geq 2$, we solve (2) at most for every subset $B \subseteq W(u)$, and for every color $\theta(u, v_i) \in W(u)$. Since there are less than 2^k non empty subsets B, at most $(k+1)$ possible colors for $\theta(u, v_i)$, and at most 2^k subsets C (the problems' variable), the time complexity for computing $h(u, V_i, A)$ for every u's sons, is $O((k+1) \cdot 2^{2k} \cdot t)$. Then, this is the time complexity to compute $f(u, A)$ for an internal vertex u with t sons.

In conclusion, starting from the leaves, we can compute $f(v, A)$ for every internal node v, from the lowest level of the tree until we reach r. Since we need

$$O((k+1) \cdot 2^{2k} \cdot t_v)$$

time for each internal node v with t_v sons, the total running-time will be

$$\sum_{v\in V} O((k+1) \cdot 2^{2k} \cdot t_v)$$

which is

$$O((k+1) \cdot 2^{2k} \cdot |V|),$$

as the number of edges in a tree is $|V| - 1$. The proof is complete. □

We are ready to obtain the next result.

Theorem 4 *The maximum 2-edge-colorable subgraph problem is FPT with respect to the dimension of the cycle space.*

Proof: Let $t = dim(C(G))$. In polynomial time, we can find t edges whose removal leave a forest. Let F be this set of t edges, and let $T = G - F$. Any maximum 2-edge-colorable subgraph of G colors some subset of edges of F. Thus, we can guess this subset. The number of choices is 3^t (each edge of F is either of color 1 or 2, or 0 meaning that it is uncolored). Now, consider any of these guesses. If it contains at least three edges adjacent to the same vertex, or two edges of the same color incident to the same vertex, then we do not consider it. If it contains two edges e and f incident to the same vertex z such that edges have different color, we remove z and forbid the corresponding color on the other end-point of e and f in T. If an edge is not adjacent to any other edge in the guess, we simply remove it and forbid its color in its end-points on T. Having done this, we get an instance of the forest problem with constraints on vertices. By Lemma 3, we can find a largest 2-edge-colorable subgraph respecting the constraints in polynomial time. Thus, we can compare the sizes of all these 2-edge-colorable subgraphs and get a maximum 2-edge-colorable subgraph of G in polynomial time. Thus, the total running-time of our algorithm is $3^t \cdot poly(size)$. The proof is complete. □

Remark 6 *Theorem 4 can be deduced as a consequence of our observation on treewidth (see Corollary 1) and Lemma 1, since by Lemma 1 of [32], the size of the smallest feedback edge-set is an upper bound for the size of the smallest feedback set, which in its turn is an upper bound for the treewidth.*

Despite this "negative" remark, we believe that our proof is interesting, as it relies on Lemma 3 which can be useful in other situations. Also note that our proof allows us to obtain an explicit expression for the running-time of the algorithm. As it is stated in the end of Section 7.4.2 of [11], obtaining the exact expression for the running-time of algorithms arising from Courcelle's theorem could be a non-trivial task. The reader is invited to take a look at the end of Section 7.4.2 of [11] for further details on this.

The strategy of the proof of Theorem 4 implies the following corollary:

Corollary 2 *Let $G = (V, E)$ be a connected graph with $|E| \leq |V| + \log |V|$ edges. Then the maximum 2-edge-colorable subgraph problem can be solved in polynomial time for this type of graphs.*

Proof: The proof is the same. Start with any spanning tree T of G. Observe that the number of edges of G outside T is at most $\log |V|$. Guess all possible assignments of 2-colors to these edges. Since their number is at most $\log |V|$, we have that the total number of guesses in polynomial in $|V|$. For each of the guesses, via Lemma 3, we find a largest 2-edge-colorable subgraph respecting the constraints arising from the guesses in polynomial time. Thus, we can compare the sizes of all these 2-edge-colorable subgraphs and get a maximum 2-edge-colorable subgraph of G in polynomial time. The proof is complete. $\square$

Using this corollary, one can show that our problem is FPT with respect to $MaxLeaf(G)$. Recall that for a connected graph G, $MaxLeaf(G)$ is defined as the maximum number of leaves in a spanning tree of G. In order to derive this result, we will use the following

Theorem 5 *([12]) Let G be a simple connected graph with $|E| \geq |V| + \frac{t(t-1)}{2}$ edges and $|V| \neq t+2$. Then $MaxLeaf(G) > t$ and the bound is best possible.*

We are ready to prove:

Proposition 2 *The maximum 2-edge-colorable subgraph problem is FPT with respect to $MaxLeaf(G)$.*

Proof: Clearly, we can assume that the input graph G is connected. If $|E| \leq |V| + \log |V|$, then Corollary 2 implies that we can find a maximum 2-edge-colorable subgraph in polynomial time. Thus, without loss of generality, we can assume that $|E| > |V| + \log |V|$. Observe that if we choose $t = \lfloor \sqrt{2 \log |V|} \rfloor$, then

$$\log |V| \geq \frac{t^2}{2},$$

hence

$$|E| > |V| + \log |V| \geq |V| + \frac{t^2}{2} > |V| + \frac{t(t-1)}{2}.$$

Since $|V| \neq t + 2 = \lfloor \sqrt{2 \log |V|} \rfloor + 2$ (this is true for sufficiently large $|V|$, we can solve small instances with the brute force algorithm directly), Theorem 5 implies that

$$MaxLeaf(G) > t = \lfloor \sqrt{2 \log |V|} \rfloor > \sqrt{2 \log |V|} - 1,$$

or

$$|V| < 2^{\frac{(MaxLeaf(G)+1)^2}{2}}.$$

In other words, $|V|$ is bounded in terms of $MaxLeaf(G)$. Thus the trivial algorithm will solve the problem in FPT time with respect to $MaxLeaf(G)$. The proof is complete. □

One may wonder, whether our problem is FPT with respect to the complementary parameter $|V| - MaxLeaf(G)$? Observe that $|V| - MaxLeaf(G) = 1$ if and only if $MaxLeaf(G) = |V| - 1$. The latter condition is equivalent to the statement that the graph under consideration contains a spanning star. However, the latter condition is the same as having $rad(G) = 1$. Thus, combined with Theorem 2, we get:

Proposition 3 *The maximum 2-edge-colorable subgraph problem is paraNP-hard with respect to $|V| - MaxLeaf(G)$.*

5 The maximum 2-edge-colorable subgraph problem and the method of iterative compression

In this section, we consider two problems related to the maximum 2-edge-colorable subgraph problem. Our main goal here is to show that these two problems are FPT with respect to the budget ℓ using the method of iterative compression as it is described in Section 4 of [11]. After obtaining these results, we present two branching algorithms and compare their running-time with the ones obtained from the method of iterative compression.

The first problem that we will consider is the following:

Problem 4 *Given a graph G and an integer ℓ, is there $X \subseteq E(G)$, such that $|X| \leq \ell$ and $G - X$ is 2-edge-colorable?*

Observe that if G is a cubic graph, then in order to get a 2-edge-colorable subgraph, for each vertex v of G we have to remove at least one edge incident to v. Thus, we have to remove at least $\frac{|V|}{2}$ edges. Now, observe that in cubic graphs there is a set of size $\frac{|V|}{2}$ whose removal leaves a 2-edge-colorable subgraph if and only if G is 3-edge-colorable. Thus, combined with [19], we have that Problem 4 is NP-complete even for cubic graphs.

We continue with the following lemma.

Lemma 4 *Consider the following decision problem: given a bipartite graph H with $\Delta(H) \leq 2$, two subsets of edges $E_1, E_2 \subseteq E(H)$, and an integer ℓ. The goal is to check whether there is a subset $X \subseteq E(H)$, such that $|X| \leq \ell$ and $H - X$ has a proper 2-edge-coloring $f : E(H)\backslash X \to \{1, 2\}$, such that if $e \in E_1 - X$, then $f(e) = 1$, and $e \in E_2 - X$, then $f(e) = 2$. There is a polynomial time algorithm for solving this problem.*

Proof: We will actually prove that the minimization version of this problem is polynomial time solvable. Thus, we can find the smallest size of a feasible set X can compare it with ℓ. Clearly, when solving the minimization problem we can focus solely on connected graphs. Thus, H is a path or a cycle. Let us start with paths. Let $E_1 \cup E_2 = \{e_1, ..., e_r\}$. Assume that the labelling is done so that when you look at the path from left to right, the edges appear in this order. Let us define the notion of a conflict. A pair of consecutive edges e_j, e_{j+1} forms a conflict if the length of the path between them is even and they belong to the same E_1 (or the same E_2) (roughly speaking their colors should be the same) or they belong to different E_js and their distance is odd. Observe that this definition is meaningful even when e_j and e_{j+1} are incident to the same vertex. In this case the distance is zero, hence we have a conflict if they must have the same color.

Now the critical observation is that if one has a smallest X that overcomes the conflicts, then we can always assume that $X \subseteq \{e_1, ..., e_r\}$. This follows from the observation that if you have removed an edge f overcoming a certain conflict, then you can remove the closest right edge from $\{e_1, ..., e_r\}$. Observe that in the latter case this edge from $E_1 \cup E_2$ will remove the conflict next to it, too. Thus, if we assume that e_{i_1}, e_{i_1+1},...,e_{i_q}, e_{i_q+1} form consecutive conflicts, then by removing $X = \{e_{i_1+1}, e_{i_2+1}, ..., e_{i_q+1}\}$ we will get rid of all conflicts. Moreover, the feasible set X will be the smallest.

Now, let us consider the case of cycles C. Again, we can assume that $E_1 \cup E_2 = \{e_1, ..., e_r\}$. Moreover, we will assume the same way of labelling the edges. The conflicts will be defined in the same way (we assume some circumference order on the cycle C). Now, let us show that any two consecutive edges e and e' from $E_1 \cup E_2$ must form a conflict. Assume not. Let e and e' be two consecutive edges such that there is no conflict between them. Consider the subpath of the cycle C starting from e and ending on e'. Solve the optimization problem in this subpath P. Clearly we can extend the 2-edge-coloring of $P - X$ to that of $C - X$.

Thus, we are left with the assumption that e_1, e_2, ..., e_{r-1}, e_r and e_r, e_1 form conflicts. Now, if r is even then by taking $X = \{e_2, e_4, ..., e_r\}$ we will have that $G - X$ is without conflicts and clearly X is smallest. On the other hand, if r is odd, then $X = \{e_2, e_4, ..., e_{r-1}\} \cup \{e_r\}$ is a smallest feasible set. The proof is complete. □

As in [11] (see Section 4), this lemma implies that the disjoint version of the problem is FPT with respect to ℓ.

Lemma 5 *In the disjoint version of the problem we are given a graph G, integer ℓ and $W \subseteq E(G)$, such that $G - W$ is 2-edge-colorable and $|W| = \ell + 1$. The goal is to check whether there is $X \subseteq E(G) \backslash W$ such that $|X| \leq \ell$ and $G - X$ is 2-edge-colorable. This problem can be solved in time $2^\ell \cdot poly(size)$.*

Proof: The proof is similar to the ones given in Section 4 of [11]. Since the edges of W cannot deleted in $G - X$, it is necessary that $G[W]$ is 2-edge-colorable. Let us consider all possible 2-edge-colorings f_W of W. Clearly, their number is at most $2^{|W|} = 2^{\ell+1}$. Now for each of them let E_1^W and E_2^W be the color classes of f_W. Let E_1 be the set of edges of $G - W$ that are adjacent to an edge from E_2^W. Similarly, let E_2 be the set of edges of $G - W$ that are adjacent to an edge from E_1^W. Observe that any edge of E_1 either has to be deleted or colored with 1. Similarly, any edge of E_2 either has to be deleted or colored with 2. Thus, in order to solve this problem we have to solve the instance of the problem from Lemma 4 for $(G - W, E_1, E_2, \ell)$. Observe that the edges outside W which are incident to degree-two vertices of $G[W]$, must be deleted (they must be taken in X). Also observe that by the definition of W, $G - W$ satisfies the conditions of the Lemma 4. According to the lemma, each of these $2^{|W|} = 2^{\ell+1}$ instances can be solved in time $poly(size)$. Thus, we have the desired running time. The proof is complete. □

Since the disjoint version of our problem can be solved in time $2^\ell \cdot poly(size)$, we immediately have the following result as a consequence of the method of iterative compression (Section 4 of [11]):

Theorem 6 *Problem 4 is FPT with respect to ℓ and it can be solved in time $3^\ell \cdot poly(size)$.*

Proof: This just follows from the method of iterative compression. See Section 4.1.1 of [11] The proof is complete. □

Remark 7 *There is a simple branching algorithm that leads to the running time from Theorem 6: Consider the current subgraph of the input graph. If it has a maximum degree equal to two then delete one edge in each odd cycle. This can be done in polynomial time. Otherwise, pick a vertex with the largest degree* $\Delta \geq 3$*. For this vertex, we need to delete almost all (at least* $\Delta - 2$*) edges except two. We branch on several cases in which edges need to be deleted. It can be easily seen that the worst case is for degree* $\Delta = 3$*, which leads to an algorithm with running-time* 3^{ℓ}*.*

Now, we turn to the vertex-set removal version of the problem.

Problem 5 *Given a graph* G *and an integer* ℓ*, is there* $X \subseteq V(G)$*, such that* $|X| \leq \ell$ *and* $G - X$ *is* 2*-edge-colorable?*

It can be shown that Problem 5 is NP-complete, too. This just follows from Lemma 1,2 and Theorem 1 from [10]. The key observation from Theorem 1 there is that when the authors find an odd cycle transversal (a subset $V' \subseteq V(H)$, such that $H - V'$ is bipartite) of the line graph $L(G)$, they actually have that it is also gives an even 2-factor. In other words, $L(G) - V'$ is 2-edge-colorable.

We continue with the following lemma.

Lemma 6 *Let* K *be a graph with* $\Delta(K) \leq 2$*. Consider a graph* H *obtained from* K *by attaching maximum one pendant edge to some degree-two vertices of* K*. Assume that the resulting degree three vertices are independent, moreover, on pendant edges we have a color from* $\{1, 2\}$ *which must be satisfied. Consider the following problem: find a smallest subset* J *of degree two vertices of* H *(we are not allowed to take degree-one or degree-three vertices), such that* $H - J$ *admits a 2-edge-coloring respecting the constraints on pendant edges. This problem can be solved in polynomial time.*

Proof: Since we are solving a minimization problem, we can assume that H is connected. Let us start with the case when K is a path. Look at the path from left to right, and take the first two degree one vertices with constraints. Let w_1 and w_2 be these two degree one vertices. First, assume that the unique neighbor of w_1 is of degree two. Then consider the vertices u, v, the neighbors of the degree three vertex adjacent to w_2. Assume u is between w_1 and w_2. Observe that we have to remove at least one of u or v. Now, if the path between w_1 and w_2 does not create a conflict (see the proof of Lemma 4 for the definition of a conflict), then clearly we can just remove v and solve the resulting smaller instance. On the other hand, if the path between w_1 and w_2 creates a conflict, then we need to remove a vertex between them, thus it is safe to remove u and solve the remaining smaller instance. Now, assume that the neighbor of w_1 is of degree three. Then on its left there is no other conflicting edge. Thus, it suffices to remove the neighbor of the neighbor of w_1 that is between w_1 and w_2, and solve the resulting smaller instance. This allows us to consecutively solve the case of paths.

Now, assume that we have a cycle in K and some pendant edges with constraints are added to it in order to obtain H. If there are two consecutive degree one vertices such that the path between them is not creating a conflict, then let this path be P. Observe that $|V(P)| \geq 3$. First assume that P contains at least four vertices. Consider the unique neighbors of neighbors of w_1 and w_2, respectively, that lie outside P. Clearly we can remove these two vertices (because we have to remove at least one around w_1 and w_2), and solve the resulting problem for the resulting path, and of course we can find the coloring of P satisfying the constraints because it is conflict free. On the other hand, if $|V(P)| = 3$, we solve two cases of the path problem: first we remove the unique neighbor of degree-three vertices (the unique degree-two vertex of P) and find the smallest set of vertices for the remaining path problem. Next, we remove the two vertices adjacent to the degree-three vertices that differ from the common vertex between them and solve the resulting path problem. Then we take the smaller of the two. Observe that the path instances are not reduced to multiple instances. Hence we get just two instances here.

Thus we are left with the case, that any two consecutive pendant edges of the cycle form a conflict. Now for each of the pendant edges, remove, for example, the right degree two vertex adjacent to the degree vertex. Clearly this will be a smallest set, as otherwise, if we assume that there is a smaller one, then clearly there are two consecutive pendant edges such that between them no vertex is removed, hence there can be no 2-coloring extending the constraints. The proof is complete. □

Our next lemma works with the following extension of the previous problem.

Lemma 7 *Let K be a graph with $\Delta(K) \leq 2$. Consider a graph H obtained from K by adding new vertices w, and joining ws to some vertices of K with edges and adding a color from $\{1, 2\}$ as a constraint. All edges adjacent to the same w have the same color as a constraint. Consider the following decision problem: for this type of graph H and an integer ℓ, check whether there is a subset $X \subseteq V(K)$ (we are not allowed to take the vertices w in X), such that $|X| \leq \ell$ and $G - X$ admits a 2-edge-coloring that satisfies the constraints on edges incident to w. This problem is FPT with respect to ℓ and it can be solved in time $2^{\ell(\ell+1)} \cdot poly(size)$.*

Proof: Let Q_1 be the number of those ws that are adjacent to exactly one edge. Similarly, let $Q_{\geq 2}$ be the number of those ws that are adjacent to at least 2 edges. Observe that we can assume that $Q_{\geq 2} \leq \ell$ as for each such vertex we have to remove at least one neighbor from K, thus if their total number is greater than ℓ, the instance is a no-instance. Thus, $Q_{\geq 2} \leq \ell$. Now observe that each fixed w of degree at least two, must have degree at most $\ell + 1$, as if its degree is at least $\ell + 2$, then at least $\ell + 1$ neighbors should be removed, hence we have a no-instance. Let us call these neighbors of ws as roots. Thus, each w is adjacent to at most $k + 1$ roots. Hence, the total number of roots is at most $Q_1 + \ell(\ell + 1)$.

Now let us guess all subsets of those $\ell(\ell + 1)$ roots that are not counted in Q_1. Their number is at most $2^{\ell(\ell+1)}$. Let R be such a guess. Then in the graph $H - R$ we need to check whether $d_{H-R}(w) \leq 1$ for any w. Also we need to have $|R| \leq \ell$. If one of these conditions is not satisfied then the guess is wrong. Now, if these conditions are satisfied then clearly we cannot have adjacent degree three vertices, as in the solution at least one of adjacent degree three vertices must be removed, and hence R is not the correct guess. Thus, if this condition is also satisfied for R, we need to solve the instance of the problem from previous lemma and check whether a smallest subset of size at most $\ell' = \ell - |R|$ exists. Lemma 6 guarantees that each of these instances can be solved in polynomial time. Thus, the total running-time of our algorithm will be $2^{\ell(\ell+1)} \cdot poly(size)$. The proof is complete. □

Now, we solve the disjoint version of our problem.

Lemma 8 *In the disjoint version of our problem, we are given a graph G, integer ℓ and $W \subseteq V(G)$, such that $G - W$ is 2-edge-colorable and $|W| = \ell + 1$. The goal is to check whether there is $X \subseteq V(G) \backslash W$ such that $|X| \leq \ell$ and $G - X$ is 2-edge-colorable. This problem can be solved in time $2^{(\ell+1)^2} \cdot poly(size)$.*

Proof: Observe that if the solution exists, $G[W]$ must be 2-edge-colorable. Thus, we can guess all its 2-edge-colorings. Their total number is at most $2^{\ell+1}$. Now, for each of those guesses f_W, we define the set E_1 as those edges that are adjacent to an edge of W of color 2, and similarly, let

E_2 be those edges of G that are adjacent to an edge W of color 1. If an edge is both from E_1 and E_2 then we must delete its neighbor outside W. Now, in order to answer our problem, we need to solve the instance of the problem from the previous lemma for the graph $G - W$ with constraints on E_1 and E_2 and the parameter ℓ. Observe that we are not allowed to touch the vertices in W. According to the previous lemma, each such instance can be solved in time $2^{\ell(\ell+1)} \cdot poly(size)$. Thus the total running time for the disjoint version of our problem is

$$2^{\ell+1} \cdot 2^{\ell(\ell+1)} \cdot poly(size) = 2^{(\ell+1)^2} \cdot poly(size).$$

The proof is complete. □

Since the disjoint 2-vertex-coloring problem can be solved in time $2^{(\ell+1)^2} \cdot poly(size)$, we immediately have the following result as a consequence of the method of iterative compression:

Theorem 7 *Problem 5 is FPT with respect to ℓ and can be solved in time $2^{(\ell+1)^2+\ell} \cdot poly(size)$.*

Proof: This just follows from the method of iterative compression (see Section 4.1.1 of [11]). The actual expression that needs to be bounded is the following one multiplied with $poly(size)$:

$$\sum_{i=0}^{\ell} \binom{\ell+1}{i} 2^{(\ell-i+1)^2} \leq 2^{(\ell+1)^2} \cdot \sum_{i=0}^{\ell} \binom{\ell+1}{i} \leq 2^{(\ell+1)^2} \cdot 2^{\ell+1} = 2^{(\ell+1)^2+\ell+1}.$$

Thus, the running time will be $2^{(\ell+1)^2+\ell} \cdot poly(size)$. The proof is complete. □

Remark 8 *The running time in Theorem 7 is not close to the optimal. An algorithm with a smaller running-time can be obtained as follows. As in Remark 7 consider a vertex v of the largest degree $\Delta \geq 3$ and branch on several possibilities which vertices in a closed neighborhood need to be deleted. If we delete v we decrease the parameter by one. Otherwise, we need to delete $\Delta - 2$ vertices in $N(v)$. So if $T(\ell)$ is the running of this algorithm, we will have the following recurrence for it:*

$$T(\ell) \leq T(\ell-1) + \frac{\Delta(\Delta-1)}{2} T(\ell - \Delta + 2)$$

Note that this leads to a c^{ℓ} algorithm with $c \leq 2$ instead of $2^{(\ell+1)^2+\ell}$ presented in Theorem 7.

6 Conclusion and future work

In this paper, we considered the maximum 2-edge-colorable subgraph problem. Our main goal was to address this problem from the perspective of fixed-parameter tractability. Our results state that this problem is paraNP-hard with respect to radius, diameter and $|V| - MaxLeaf$. On the positive side, it is fixed-parameter tractable with respect to $|V| - \delta$, the size of largest matching, the dimension of the cycle space and $MaxLeaf$. Moreover, it is polynomial time solvable for the graphs containing at most $|V| + \log |V|$ edges.

From our perspective the following line of research is suitable for future research. For a graph G, let $\tau(G)$ be the size of the smallest vertex cover of G. Since in any graph $\nu(G) \leq \tau(G)$, Corollary 1 and Lemma 1 imply that the maximum 2-edge-colorable subgraph problem is FPT with respect to $\tau(G)$. We would like to ask:

Question 1 *Is the maximum 2-edge-colorable subgraph problem FPT with respect to $\tau(G) - \nu(G)$?*

The classical 2-approximation algorithm for the vertex cover problem and its analysis imply that for any graph G, we have $\tau(G) \leq 2 \cdot \nu(G)$. This inequality means that in any graph G, $\tau(G) - \nu(G) \leq \nu(G) \leq \tau(G)$. Thus, a positive answer to Question 1 will strengthen Corollary 1 and its consequence for $\tau(G)$.

Acknowledgement

The author would like to thank Dr. Zhora Nikoghosyan for useful discussions on Hamiltonian graphs. He also would like to thank Dr. Kenta Ozeki for pointing him out the paper [12].

References

[1] M. Albertson and R. Haas. Parsimonious edge coloring. *Discrete Mathematics*, 148:1–7, 1996. doi:10.1016/0012-365X(94)00254-G.

[2] M. Albertson and R. Haas. The edge chromatic difference sequence of a cubic graph. *Discrete Mathematics*, 177:1–8, 1997. doi:10.1016/S0012-365X(96)00368-8.

[3] A. Aloisio. Fixed-parameter tractability for branchwidth of the maximum-weight edge-colored subgraph problem. In L. Barolli, editor, *AINA 2024, Advanced Information Networking and Applications*, volume 204 of *Lecture Notes on Data Engineering and Communications Technologies*, pages 86–95. Springer, Cham., 2024. doi:10.1007/978-3-031-57942-4_10.

[4] A. Aloisio and V. Mkrtchyan. On the fixed-parameter tractability of the maximum 2-edge-colorable subgraph problem. *CoRR*, abs/1904.09246, 2019. URL: http://arxiv.org/abs/1904.09246, arXiv:1904.09246.

[5] A. Aloisio and V. Mkrtchyan. Algorithmic aspects of the maximum 2-edge-colorable subgraph problem. In L. Barolli, I. Woungang, and T. Enokido, editors, *AINA 2021, Advanced Information Networking and Applications*, volume 227 of *Lecture Notes in Networks and Systems*, pages 232–241. Springer, Cham., 2021. doi:10.1007/978-3-030-75078-7_24.

[6] D. Aslanyan, V. Mkrtchyan, S. Petrosyan, and G. Vardanyan. On disjoint matchings in cubic graphs: Maximum 2-edge-colorable and maximum 3-edge-colorable subgraphs. *Discrete Applied Mathematics*, 172:12–27, 2014. doi:10.1016/j.dam.2014.03.001.

[7] B. Bollobás. *Extremal Graph Theory.* Academic Press, London, New York, San Francisco, 1978.

[8] Y. Cao, G. Jin, R. Luo, V. Mkrtchyan, C.-Q. Zhang, and Y. Zhao. Decomposition of class II graphs into two class I graphs. *Discrete Mathematics*, 346(12):113610, 2023. doi:10.1016/j.disc.2023.113610.

[9] A. Cavicchioli, M. Meschiari, B. Ruini, and F. Spaggiari. A survey on snarks and new results: Products, reducibility and a computer search. *J. Graph Theory*, 28(2):57–86, 1998. doi:10.1002/(SICI)1097-0118(199806)28:2<57::AID-JGT1>3.0.CO;2-D.

[10] N. Chiarelli, T. Hartinger, M. Johnson, M. Milanic, and D. Paulusma. Minimum connected transversals in graphs: New hardness results and tractable cases using the price of connectivity. *Theor. Comput. Sci.*, 705:75–83, 2018. doi:10.1016/j.tcs.2017.09.033.

[11] M. Cygan, F. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized algorithms.* Springer International Publishing, 2015. doi:10.1007/978-3-319-21275-3.

[12] G. Ding, T. Johnson, and P. Seymour. Spanning trees with many leaves. *Journal of Graph Theory*, 37(4):189–197, 2001. doi:10.1002/jgt.1013.

[13] U. Feige, E. Ofek, and U. Wieder. Approximating maximum edge coloring in multigraphs. *Lecture Notes in Computer Science*, 2462:108–121, 2002. doi:10.1007/3-540-45753-4_11.

[14] A. D. Flaxman and S. Hoory. Maximum matchings in regular graphs of high girth. *The Electronic Journal of Combinatorics*, 14(1):1–4, 2007. doi:10.37236/1002.

[15] F. V. Fomin and D. Kratsch. *Exact Exponential Algorithms.* Springer-Verlag, Berlin, Heidelberg, 2010. doi:10.1007/978-3-642-16533-7.

[16] J.-L. Fouquet and J.-M. Vanherpe. On parsimonious edge-coloring of graphs with maximum degree three. *Graphs and Combinatorics*, 29(3):475–487, 2013. doi:10.1007/s00373-012-1145-3.

[17] E. Galby, P. T. Lima, D. Paulusma, and B. Ries. On the parameterized complexity of k-edge coloring. 2019. URL: https://arxiv.org/abs/1901.01861.

[18] M. A. Henning and A. Yeo. Tight lower bounds on the size of a maximum matching in a regular graph. *Graphs and Combinatorics*, 23(6):647–657, 2007. doi:10.1007/s00373-007-0757-5.

[19] I. Holyer. The np-completeness of edge-coloring. *SIAM J. Comp.*, 10(4):718–720, 1981. doi:10.1137/021005.

[20] M. J. Kamiński and L. Kowalik. Beyond the Vizing's bound for at most seven colors. *SIAM J. Discrete Math.*, 28(3):1334–1362, 2014. doi:10.1137/120899765.

[21] L. Karapetyan and V. Mkrtchyan. On maximum k-edge-colorable subgraphs of bipartite graphs. *Disc. Appl. Math.*, 257(1):226–232, 2019. doi:10.1016/j.dam.2018.10.013.

[22] A. Kosowski. Approximating the maximum 2- and 3-edge-colorable problems. *Disc. Appl. Math.*, 157(17):3593–3600, 2009. doi:10.1016/j.dam.2009.04.002.

[23] L. Kowalik, J. Lauri, and A. Socała. On the fine-grained complexity of rainbow coloring. *SIAM J. Disc. Math.*, 32(3):1672–1705, 2018. doi:10.1137/16M1102690.

[24] V. Mkrtchyan, S. Petrosyan, and G. Vardanyan. On disjoint matchings in cubic graphs. *Discrete Mathematics*, 310:1588–1613, 2010. doi:10.1016/j.disc.2010.02.007.

[25] V. Mkrtchyan, S. Petrosyan, and G. Vardanyan. Corrigendum to "On disjoint matchings in cubic graphs". *Discrete Mathematics*, 313(21):2381, 2013. doi:10.1016/j.disc.2013.06.020.

[26] V. Mkrtchyan and E. Steffen. Maximum Δ-edge-colorable subgraphs of class II graphs. *J. Graph Theory*, 70(4):473–482, 2012. doi:10.1002/jgt.20629.

[27] T. Nishizeki. On the maximum matchings of regular multigraphs. *Discrete Mathematics*, 37:105–114, 1981. doi:10.1016/0012-365X(81)90143-6.

[28] T. Nishizeki and I. Baybars. Lower bounds on the cardinality of the maximum matchings of planar graphs. *Discrete Mathematics*, 28:255–267, 1979. doi:10.1016/0012-365X(79)90133-X.

[29] R. Rizzi. Approximating the maximum 3-edge-colorable subgraph problem. *Discrete Mathematics*, 309(12):4166–4170, 2009. doi:10.1016/j.disc.2008.11.017.

[30] R. Sasák. Comparing 17 graph parameters. Master thesis, University of Bergen, 2010.

[31] C. E. Shannon. A theorem on coloring the lines of a network. *J. Math. Physics*, 28:148–151, 1949. doi:10.1002/sapm1949281148.

[32] Y. Song and M. Yu. On the treewidths of graphs of bounded degree. *PLoS ONE*, 10(04):e0120880, 2015.

[33] E. Steffen. Classifications and characterizations of snarks. *Discrete Mathematics*, 188:183–203, 1998. doi:10.1016/S0012-365X(97)00255-0.

[34] E. Steffen. Measurements of edge-uncolorability. *Discrete Mathematics*, 280:191–214, 2004. doi:10.1016/j.disc.2003.05.005.

[35] M. Stiebitz, D. Scheide, B. Toft, and L. M. Favrholdt. *Graph Edge Coloring*. John Wiley and Sons, 2012. doi:10.1002/9781118032497.

[36] V. Vizing. On an estimate of the chromatic class of a p-graph. *Diskret Analiz*, 3:25–30, 1964.

[37] J. Weinstein. Large matchings in graphs. *Canadian Journal of Mathematics*, 26(6):1498–1508, 1974. doi:10.4153/CJM-1974-145-8.

[38] D. West. *Introduction to Graph Theory*. Prentice-Hall, Englewood Cliffs, 1996.