



**HAL**  
open science

# Minimum colored Maximum Matchings in vertex-colored Graphs

Johanne Cohen, Yannis Manoussakis, Jonas Sénizergues

► **To cite this version:**

Johanne Cohen, Yannis Manoussakis, Jonas Sénizergues. Minimum colored Maximum Matchings in vertex-colored Graphs. 2024. hal-02157745v3

**HAL Id: hal-02157745**

**<https://hal.science/hal-02157745v3>**

Preprint submitted on 9 Jul 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - ShareAlike 4.0 International License

# Minimum colored Maximum Matchings in vertex-colored Graphs

Johanne Cohen<sup>1</sup>, Yannis Manoussakis<sup>1</sup>, and Jonas S enizergues<sup>2</sup>

<sup>1</sup> LISN-CNRS, Universit e Paris-Saclay, France

<sup>2</sup> LaBRI, Universit e de Bordeaux, France

**Abstract.** We deal with three aspects of the complexity of the problem of finding a maximum matching that minimizes the number of colors in a vertex-colored graph. We first prove that the problem is  $W[2]$ -hard with the number of color of the solution as parameter. Next we establish that it is hard to approximate within logarithmic ratio of the number of internal nodes (nodes of degree 2 or more). Finally, we show that the problem is fixed-parameter tractable with the size of a maximum matching as parameter.

## 1 Introduction

Graphs are a powerful modelization tool, whose uses are widespread. However, when dealing with complex systems, it is often necessary to incorporate additional information alongside the structural relationships they represent. Many works have explored labelled graphs, that add a such new layer of information on the edges of the graph.

In this work, we focus on adding information to the vertices, specifically by using vertex coloration. For example, this approach can model scenarios such as the Web, where each vertex color represents the type of content a page holds. It is of great use in the field of bioinformatics where vertex-colored graphs emerge in genome mapping [1], protein-protein interactions [3], phylogenetic analysis [12], and metabolic network analysis [9], among others.

Our study addresses the variation of the Maximum Matching problem that aims to minimize the number of colors used. This work builds on a previous study of a related variation of the problem, where the maximum matching was termed tropical [4], a concept first introduced in [5].

Before going further, let us introduce some conventions, definitions, and notations. In this paper, the word *graph* refers to *simple non-oriented graph*.

**Definition 1.1.** A vertex-colored graph is a couple  $G^c = (G, c)$  where  $G = (V, E)$  is a simple undirected graph and  $c$  a coloring on  $V$  (i.e. a function giving a color to each vertex in  $V$ ).

Observe that it does not need to be a *proper* coloration: Two adjacent vertices can have the same color.

**Definition 1.2.**  $H^{c'}$  is said to be a (vertex-colored) subgraph of  $G^c$  a vertex-colored graph when  $H$  is a subgraph of  $G$  and  $c'$  is  $c$  restricted to  $V(H)$ .

Given the definition,  $H^{c'}$  can be alternatively written  $H^c$  or  $H$ , when it is clearly stated that it is a subgraph of  $G^c$ .

We will also use the following notations concerning vertex-sets and edge-sets when it is convenient :

- For  $G$  a graph,  $V(G)$  denotes its vertex-set,  $E(G)$  its edge-set
- For  $M$  a set of edges of  $G$ ,  $V(M)$  denotes the vertex-set of the subgraph induced by  $M$
- Given  $x$  and  $y$  two vertices,  $xy$  denotes the edge, if any, between  $x$  and  $y$  in  $G$ .

While, in a graph  $G^c$  where  $G = (V, E)$ , for  $x$  a vertex in  $V$ ,  $c(x)$  is already well-defined, we will also use the following notations :

- For  $A \subset V$ ,  $c(A)$  denotes direct image of  $A$  by  $c$  (the set of the colors of  $A$ )
- For  $H$  a subgraph of  $G^c$ ,  $c(H)$  denotes the direct image by  $c$  of its vertex-set
- For  $M$  a set of edges, subset of  $E$ ,  $c(M)$  denotes  $c(V(M))$ .

**Definition 1.3.** In a vertex-colored graph  $(V, E)^c$ , a set of vertices  $A \subset V$  is said to be *tropical* when the set of colors used on  $A$  is exactly the same as the set of colors used in the entire graph (i.e. when  $c(A) = c(V)$ ).

By extension, a set of edges  $M$  is said to be *tropical* when the vertex-set of its induced subgraph is tropical (i.e. when  $c(M) = c(V)$ ).

In our constructions, we will use the following concept to formalize arbitrary choices:

**Definition 1.4.** Given a set  $S$ ,  $f$  is said to be a choice function on  $S$  if for any nonempty subset  $S'$  of  $S$ ,  $f(S') \in S'$ .

The problem of finding a maximum matching is a classical one. Here is a proper definition to extend it to the colored case.

A *matching*  $M$  is a subset of edges of  $E(G^c)$  such that no two edges in the matching share a common incident vertex. The vertices incident to an edge of  $M$  are said to be *matched* or *covered* by  $M$ . A *maximal* matching is a matching that is maximal (under inclusion), while a *maximum* matching is a matching with highest cardinality among all possible matchings. Note that a maximum matching is, by definition, always a maximal matching.

The decision problem associated with this optimization problem is known to be polynomial [8], but what happens when we add some constraint on the colors to the problem? For example, one could think about the *tropical* version of the problem :

TROPICAL MAXIMUM MATCHING

- Input:*         $A$  vertex-colored graph  $G^c$   
*Output:*       $A$  tropical maximum matching  $M$  of  $G^c$ , if any

We can observe that a perfect matching is always tropical. Consequently, the above question is interesting for maximum (but not perfect) matchings. In [4], the authors address this case and provide a polynomial-time algorithm. Using their Theorem 2.2, an immediate corollary is that we still have a polynomial time algorithm when we replace *tropical* with *maximum colored* :

MAXIMUM (VERTEX-)COLORED MAXIMUM MATCHING

*Input:*        *A vertex-colored graph  $G^c$*

*Output:*       *A maximum matching  $M$  of  $G^c$  with maximum number of colors*

Another natural variation is to consider the minimization of the number of colors instead of maximizing it :

MINIMUM (VERTEX-)COLORED MAXIMUM MATCHING (MCMM)

*Input:*        *A vertex-colored graph  $G^c$*

*Output:*       *A maximum matching  $M$  of  $G^c$  with minimum number of colors*

That problem, however, is not as easy to solve as we will prove. The corresponding decision problem is indeed NP-hard, as a direct consequence of the reduction used for Theorem 2.1, which states that one cannot expect to easily solve instances of the *Minimum Colored Maximum Matching* (MCMM) problem where the solution has few colors. When parameterized by the number of colors in the solution, it is at least as hard as the Minimum Dominating Set problem parameterized by the size of the solution, as we will demonstrate in Section 2.

For an approximate solution to MCMM, since finding a regular maximum matching is easy, one might consider using the number of colors of such matching as weight to evaluate the quality of a maximum matching as an approximation. As we will prove in Section 3, MCMM, parameterized in such fashion, is nearly as hard as the Set Cover problem regarding approximation.

However, as hard as the problem can be with the most natural parameter, there is another sensible one -the size of a maximum matching- with which the problem becomes Fixed-Parameter Tractable (FPT), as we prove in Section 4.

## 2 NP-hardness and W[2]-hardness of MCMM

In this section, we prove the following hardness result.

**Theorem 2.1.** *Minimum colored Maximum Matching (MCMM) is W[2]-hard on trees considering the total number of colors of the solution as parameter.*

This section is devoted to proving this theorem.

Recall that a *dominating set* of a graph  $G = (V, E)$  is a subset of vertices  $S \subseteq V$  such that every vertex of the graph is either in  $S$  or has at least a neighbor in  $S$ . The classical optimization problem is then to minimize the size of such a subset.

## MINIMUM DOMINATING SET

*Input:*        A graph  $G$   
*Output:*       A dominating set  $S$  of minimum size

The natural corresponding parameterized problem, where the parameter is the size of a solution, is known to be  $W[2]$ -complete [7]. The proof of the theorem is then based on a linear reduction from the Dominating Set problem, using the following construction and lemmas. In particular, it will be an immediate consequence of Lemma 2.7.

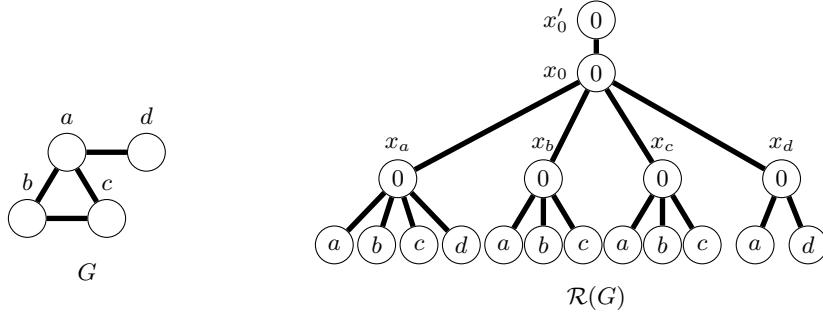
We now introduce a construction of an instance of MCMM from an instance of Dominating Set:

Given a connected simple non-colored graph  $G = (V, E)$ , we define a vertex-colored tree  $T^c$  as follows:

- $V(T) = \{x_u \mid u \in V\} \cup \{x_{u,v} \mid u \in V, v \in N[u]\} \cup \{x'_0, x_0\}$ ,
- $E(T) = \{x_0x_u \mid u \in V\} \cup \{x_u x_{u,v} \mid u \in V, v \in N[u]\} \cup \{x'_0x_0\}$ .

Then, we color the vertices of  $T$  using  $V \uplus \{0\}$  as set of colors:

- $c(x'_0) = c(x_0) = 0$ ,
- For every  $u \in V$ ,  $c(x_u) = 0$ ,
- For each  $(u, v) \in V \times N[u]$ ,  $c(x_{u,v}) = v$ .



**Fig. 1.** A graph  $G$ , its transformed version  $\mathcal{R}(G)$  (colors depicted in the nodes).

Notice that  $|V(T)| = 2|V| + 2|E| + 2$ ,  $|E(T)| = 2|V| + 2|E| + 1$ , and that we can build  $T^c$  from  $G$  in polynomial time. Notice also that there are  $|V| + 1$  internal vertices (vertices of degree 2 or more).

To make discussions easier, we let  $\mathcal{R}$  denote the function that, given  $G$  as input, returns  $T^c$ . The following series of lemmas explores the properties of  $\mathcal{R}$ .

**Lemma 2.1.**  $\{x'_0x_0\} \cup \{x_u x_{u,u} \mid u \in V\}$  is a maximum matching of  $\mathcal{R}(G)$ .

*Proof.* One can easily see that  $\{x'_0x_0\} \cup \{x_u x_{u,u} \mid u \in V\}$  is a matching and that there is no augmenting path since all paths between an unmatched vertex to another one are of length 4. Thus, this matching is a maximum one.

An immediate consequence of the previous lemma is that the size of any maximum matching of  $\mathcal{R}(G)$  is  $|V| + 1$ .

**Lemma 2.2.** *If  $M$  is a matching of  $\mathcal{R}(G)$  and  $M \cap \{x_0x_u \mid u \in V\} \neq \emptyset$ , then  $M$  is not a maximum one.*

*Proof.* Let  $M$  be a matching of  $\mathcal{R}(G)$ .

Assume that  $x_0x_u \in M$  for some  $u \in V$ . Then since  $M$  is a matching,  $x'_0x_0$  and  $x_u x_{u,u}$  are not in  $M$ , so  $x'_0, x_0, x_u, x_{u,u}$  is an augmenting path and  $M$  is not maximum.

**Lemma 2.3.** *Let  $u$  be a vertex in  $G$ . Any maximum matching of  $\mathcal{R}(G)$  uses exactly one edge in  $\{x_u x_{u,v} \mid v \in N[u]\}$  and contains the edge  $x'_0x_0$ .*

*Proof.* Assume that a matching  $M$  of  $\mathcal{R}(G)$  has no edge in  $\{x_u x_{u,v} \mid v \in N[u]\}$ . Since  $x_0x_u \notin M$  by Lemma 2.2,  $x_u$  is unmatched in  $M$ . Then  $M \cup \{x_u x_{u,u}\}$  is a matching greater than  $M$ , and  $M$  is not maximum. Thus any maximum matching must contain at least one edge in  $\{x_u x_{u,v} \mid v \in N[u]\}$ , and thus contains exactly one as they all have  $x_u$  as an end.

By the same argument,  $x'_0x_0$  must be in any maximum matching, which concludes the proof.

Let  $M$  be a maximum matching of  $\mathcal{R}(G)$ . We then define a function  $g$  by  $g(M) = \{v \mid \exists u \in V, x_u x_{u,v} \in M\}$ .

**Lemma 2.4.** *If  $M$  is a maximum matching of  $\mathcal{R}(G)$ , then  $g(M)$  is a dominating set of  $G$ .*

*Proof.* Let  $u$  be a vertex of  $G$ . As  $M$  is a maximum matching of  $\mathcal{R}(G)$ , by Lemma 2.3,  $M$  has one edge in  $\{x_u x_{u,u}, x_u x_{u,v} : v \in N[u]\}$ , say  $x_u x_{u,v}$ .

By the definition of  $g(M)$ ,  $v \in g(M)$ , which ensures that  $u$  is dominated by  $v$  and also by  $g(M)$ .

**Lemma 2.5.** *If  $M$  is maximum matching of  $\mathcal{R}(G)$  with  $k+1$  colors, then  $g(M)$  is a dominating set of  $G$  of size  $k$ .*

*Proof.* Let  $M$  be a maximum matching of  $\mathcal{R}(G)$  with  $k+1$  colors.

By Lemma 2.3,  $x_0$  of color 0 is covered by  $M$ . Thus  $M$  has  $k$  other colors in  $V$ . If  $c(M)$  contains the color  $v \in V$ , then by construction of  $\mathcal{R}(G)$ , there is some  $u$  such that  $x_u x_{u,v} \in M$ . The definition of the function  $g$  implies that  $v \in g(M)$ . Thus  $|g(M)| \geq k$ .

Conversely, if  $M$  does not contain a color  $v \in V$ , by construction of  $\mathcal{R}(G)$ , there is no vertex  $u$  such that  $x_u x_{u,v} \in M$ . Moreover, by definition of  $g$ ,  $v \notin g(M)$ . Thus,  $|g(M)| \leq k$ .

We conclude that  $|g(M)| = k$ , and since  $g(M)$  is a dominating set of  $G$  by Lemma 2.4,  $g(M)$  is then a dominating set of  $G$  of size  $k$ .

**Lemma 2.6.** *Graph  $G$  admits a dominating set of size  $k$  if and only if  $\mathcal{R}(G)$  admits a maximum matching with  $k + 1$  colors.*

*Proof.* By Lemma 2.5, if  $\mathcal{R}(G)$  admits a maximum matching with  $k + 1$  colors,  $G$  admits a dominating set of size  $k$ .

Conversely, assume that  $S$  is a dominating set of size  $k$  in  $G$ .

Let  $\alpha$  be an arbitrary injective valuation on  $V$ . For each  $u \in V$  we define a function  $\varphi$  by

$$\varphi(u) = \begin{cases} u, & \text{if } u \in N_G[u] \cap S \\ \min_\alpha(N_G[u] \cap S), & \text{otherwise} \end{cases}$$

Since  $S$  is a dominating set of  $G$ , for each  $u \in V$ ,  $N_G[u] \cap S$  is not empty, and  $\varphi$  is then well-defined.

Then we define  $M = \{x'_0x_0\} \cup \{x_u x_{u,v} \mid v = \varphi(u)\}$ .  $M$  is a matching by construction, and is maximum since it is of size  $|V| + 1$ . Furthermore, any vertex covered by  $M$  is of color either 0 or  $u \in S$ , and each of those  $k + 1$  colors appears at least once (if  $u \in S$  then by construction  $x_u x_{u,u} \in M$ , and  $x_{u,u}$  has  $u$  as a color). Consequently,  $M$  is  $k + 1$ -colored, which concludes the proof.

To prove  $W[2]$ -hardness with our reduction  $\mathcal{R}$ , we need to show that it is in fact a FPT reduction, that is:

1.  $\mathcal{R}$  is a reduction from Dominating Set to MCMM.
2.  $\mathcal{R}$  is computable with a FPT algorithm.
3. A computable function  $g$  must exist such that the parameter of MCMM (the number of colors of the optimal maximum matching) in  $\mathcal{R}(G)$  is less than  $g$  applied to the parameter (the size of an optimal solution) of Dominating Set in  $G$ .

**Lemma 2.7.**  *$\mathcal{R}$  is a FPT-reduction from the Dominating Set problem with parameter size of the optimal solution to the MCMM problem on trees with parameter number of colors of the optimal solution.*

*Proof.* Point 1 and Point 3 are proven in Lemma 2.6.

The computation of  $\mathcal{R}$  is polynomial in the size of  $G$  (see the construction). It is as such also FPT, and we have Point 2.

Thus,  $\mathcal{R}$  is a FPT-reduction.

*Proof (Proof of Theorem 2.1).* It is an immediate consequence of Lemma 2.7 using the fact that the Dominating Set problem is known to be  $W[2]$ -complete [7].

**Theorem 2.2.** *Minimum colored maximum matching is NP-complete on trees.*

*Proof.* It is enough to see that  $\mathcal{R}$  is also a polynomial reduction from the Dominating Set problem to the MCMM problem on trees.

### 3 Hardness of approximating MCMM

We consider as candidate for approximating MCMM any maximum matching, with the weight function being the number of colors used. For that definition, we prove the following inapproximability result.

**Theorem 3.1.** *MCMM cannot be approximated on trees with an approximation ratio better than  $\log(N-1)(1-\varepsilon)$  (with  $0 < \varepsilon < 1$ ), where  $N$  is the number of internal vertices (vertices with degree at least 2) of  $G$ , unless  $P=NP$ .*

The proof of this theorem is based on a reduction from the Set Cover problem, which is known not to be approximable beyond a certain logarithmic ratio [6].

MINIMUM SET COVER

*Input:* A finite set  $U$ , and  $\mathcal{F} \subset \mathcal{P}(U)$  such that  $U = \bigcup_{F \in \mathcal{F}} F$

*Output:*  $\Xi \subset \mathcal{F}$  such that  $U = \bigcup_{F \in \Xi} F$  with minimum cardinality

As it is more convenient for us, we will use the equivalent following form of the problem :

MINIMUM SET COVER (BIPARTITE GRAPH)

*Input:* A bipartite graph  $G = (U, V, E)$  such that no  $u \in U$  is isolated and no two  $v, v'$  distinct vertices of  $V$  have the same neighborhood

*Output:*  $\Xi \subset V$  such that  $U = \bigcup_{v \in \Xi} N(v)$  with minimum cardinality

The proof of Theorem 3.1 uses the following construction and lemmas below. Note that the construction, and hence the following lemmas and proofs, are very close to what was done in the previous part. It should not be very surprising, given the proximity between the Dominating Set and the Set Cover problems.

Given an instance of Set Cover  $G = (U, V, E)$ , we define a vertex-colored tree  $T^c$  as follows:

- $V(T) = \{x'_0, x_0\} \cup \{x_u \mid u \in U\} \cup \{x_{u,v} \mid u \in U, uv \in E\}$ ,
- $E(T) = \{x_u x_{u,v} \mid u \in U, uv \in E\} \cup \{x_u x_0 \mid u \in U\} \cup \{x'_0 x_0\}$ .

Then we color the vertices of  $T$  with  $n+1$  colors so that :

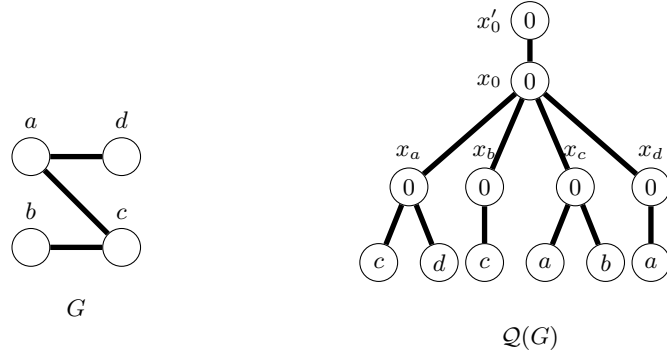
- $c(x'_0) = 0$ ,  $c(x_0) = 0$ , and for each  $u \in U$ ,  $c(x_u) = 0$ ,
- For each  $uv \in E$ ,  $c(x_{u,v}) = v$ .

Note that  $|V(T)| = |U| + |E| + 2$ ,  $|E(T)| = |U| + |E| + 1$ , and that we can obtain  $T^c$  in polynomial time from  $G$ . Note also that there are  $|U| + 1$  internal vertices.

We use  $\mathcal{Q}$  to denote the function that, given  $G$  as input, returns  $T^c$  (see Figure 2 for an example). The following lemmas explore the properties of  $\mathcal{Q}$  to prove that it is indeed a reduction of Set Cover to MCMM on trees.

**Lemma 3.1.**  $\{x'_0 x_0\} \cup \{x_u x_{u,v} \mid u \in U, uv \in E\}$  is a maximum matching of  $G$ .





**Fig. 2.** A bipartite graph  $G$ , and its transformed version  $\mathcal{Q}(G)$  (colors depicted in the nodes).

*Proof.* One can easily see that there is no augmenting path since all paths from an unmatched vertex to another are of length 4.

Moreover, no maximum matching can use an edge that does not cover a leaf since this would create an augmenting path.

**Lemma 3.2.** *If  $M$  is a matching of  $\mathcal{Q}(G)$  and  $M \cap \{x_0x_u \mid u \in U\} \neq \emptyset$ , then  $M$  is not a maximum matching.*

*Proof.* Let  $M$  be a matching of  $\mathcal{Q}(G)$ .

Let us suppose that  $x_0x_u \in M$  for some  $u \in U$ . Since there is no isolated vertex in  $G$ , there exists  $v \in V$  such that  $uv \in E$ . Then since  $M$  is a matching,  $x'_0x_0$  and  $x_u x_{u,v}$  are not in  $M$ , so  $x'_0x_0x_u x_{u,v}$  is an augmenting path and  $M$  is not maximal therefore not maximum.

**Lemma 3.3.** *Consider  $u \in U$ . Any maximum matching of  $\mathcal{Q}(G)$  uses exactly one edge in  $\{x_u x_{u,v} \mid v \in N(u)\}$ , and contains the edge  $x'_0x_0$ .*

*Proof.* Let  $M$  be a matching of  $\mathcal{Q}(G)$ . Suppose that there exists  $u \in U$  such that  $M \cap \{x_u x_{u,v} \mid v \in N(u)\} = \emptyset$ . Since  $x_0x_u \notin M$  by Lemma 3.2,  $x_u$  is unmatched in  $M$ . Then  $M \cup \{x_u x_{u,v}\}$  would be a matching of greater size, and  $M$  cannot be maximum. Thus, any maximum matching must contain at least one edge in  $\{x_u x_{u,v} \mid v \in N(u)\}$ , and thus contains exactly one as they all have  $x_u$  as an end.

By the same argument, any maximum matching must contain  $x'_0x_0$ , which concludes the proof.

Given a maximum matching  $M$  of  $\mathcal{Q}(G)$ , we then define  $g$  by  $g(M) = \{v \in V \mid \exists u, x_u x_{u,v} \in M\}$ .

**Lemma 3.4.** *If  $M$  is a maximum matching of  $\mathcal{Q}(G)$ , then  $g(M) \cup \{0\} = c(M)$  and  $g(M) = c(M) \setminus \{0\}$ .*

*Proof.* Let  $M$  be a maximum matching of  $\mathcal{Q}(G)$ .

For  $v \in g(M)$ , by definition of  $g(M)$ , there is  $u \in U$  such that  $x_u x_{u,v} \in M$ , thus  $v \in c(M)$ . Since we have by Lemma 3.3,  $x'_0 x_0 \in M$ , we have also  $0 \in c(M)$ , thus  $g(M) \cup \{0\} \subset c(M)$ .

Conversely, for  $v \in c(M) \setminus \{0\}$ , there must be  $u \in U$  such that  $x_u x_{u,v} \in M$  as only vertices  $x_{u,v}$  have color  $v$ . By definition of  $g(M)$ ,  $v \in g(M)$ . Thus,  $c(M) \subset g(M) \cup \{0\}$ .

Therefore, we have  $g(M) \cup \{0\} = c(M)$ , and the second equality follows immediately, as  $0 \notin g(M)$  by definition.

**Lemma 3.5.** *If  $M$  is a maximum matching of  $\mathcal{Q}(G)$ , then  $g(M)$  is a set cover of  $G$  (i.e. a subset of  $U$  whose union of neighborhoods gives  $V$ ).*

*Proof.* Let  $M$  be a maximum matching of  $\mathcal{Q}(G)$  and  $u$  be a vertex from  $U$ . As  $M$  is a maximum matching of  $\mathcal{Q}(G)$ , by Lemma 3.3 there exists  $v$  such that  $x_u x_{u,v}$  is in  $M$ , which ensures that  $u$  is covered by  $g(M)$ .

**Lemma 3.6.** *If  $M$  is a  $k+1$ -colored maximum matching of  $\mathcal{Q}(G)$ , then  $g(M)$  is a set cover of  $G$  of size  $k$ .*

*Proof.* Let  $M$  be a maximum matching of  $\mathcal{Q}(G)$ .

By Lemma 3.5,  $g(M) = c(M) \setminus \{0\}$ , so we have  $|g(M)| = |c(M)| - 1 = k$  (since  $0 \in c(M)$  by direct corollary of Lemma 3.3). By Lemma 3.5,  $g(M)$  is also a set cover, which concludes the proof.

**Lemma 3.7.** *A bipartite graph  $G$  admits a minimal set cover of size  $k$  if and only if  $\mathcal{Q}(G)$  admits a minimally colored maximum matching (i.e., a matching whose set of colors is minimal but could not be minimum) with  $k+1$  colors.*

*Proof.* Let  $\alpha$  be a choice function on  $V$  (i.e. a function which, for any non-empty subset of  $V$ , gives an element of the said subset).

By Lemma 3.6, if  $\mathcal{Q}(G)$  admits a minimally-colored maximum matching  $M$  with  $k+1$  colors, then  $G$  admits a set cover  $g(M)$  of size  $k$ . Assume that  $g(M)$  was not minimal, i.e. that there exists  $v_0 \in g(M)$  such that  $g(M) \setminus \{v_0\}$  is a set cover of size  $k-1$ . For  $u \in U$ , let us write:

$$\varphi(u) = \alpha(\{v \mid uv \in E, v \in g(M) \setminus \{v_0\}\})$$

which is well-defined since  $g(M) \setminus \{v_0\}$  is a set cover of  $G$ . We can then define  $M' = \{x_u x_{u,\varphi(u)} \mid u \in U\} \cup \{x_0 x'_0\}$ . Notice that  $M'$  is a maximum matching since it is a matching of size  $|U| + 1$ . By construction, its color set is included in  $(g(M) \cup \{0\}) \setminus \{v_0\}$ , which contradicts the minimality of the color set of  $M$ .

Conversely, let  $S$  be a minimal set cover of  $G$  of size  $k$ . For  $u \in U$ , let us denote  $\psi(u) = \alpha(\{v \mid uv \in E, v \in S\})$  (which is well-defined since  $S$  is a set cover of  $G$ ). Then we define  $M = \{x'_0 x_0\} \cup \{x_u x_{u,\psi(u)} \mid u \in U\}$ . This matching  $M$  is of the same size as the one presented in Lemma 3.1. Thus it is a maximum matching with at most  $k+1$  colors since all colors used are in  $S \cup \{0\}$ . It remains to prove that  $M$  has  $k+1$  colors and is minimally colored. If it is false, that would mean either that it is not minimally colored, or that  $M$  has not  $k+1$  colors.

- If  $M$  was not minimally-colored, there would be a maximum matching  $M'$  of  $\mathcal{Q}(G)$  such that  $c(M') \subsetneq c(M) \subset S \cup \{0\}$ .
- If  $M$  had not  $k + 1$  colors, then we would have  $c(M) \subsetneq S \cup \{0\}$ .

In both case, there exists a matching  $M^\circ$  such that  $c(M^\circ) \subsetneq S \cup \{0\}$  which is equivalent to  $g(M^\circ) \subsetneq S$ . But  $g(M^\circ)$  is a set cover of  $G$  of size at most  $k - 1$  (by Lemma 3.7), which contradicts the minimality of  $S$ .

### Proof of Theorem 3.1

From every not minimal set cover, one can extract in polynomial time a minimal set cover that is smaller than the previous one. Then, without loss of generality, we only consider minimal set covers as approximation candidates for the Minimum Set Cover problem.

Let us suppose that MCMM is approximable with a ratio  $f(N)$  where  $N$  is the number of internal vertices (vertices of degree at least 2) of the MCMM instance.

Given an instance  $G$  of the Set Cover problem  $(U, V, E)$  with universe  $U$  of size  $k$ , we use  $\mathcal{Q}$  to compute in polynomial time an instance of MCMM (of polynomial-size in  $|U|$  and  $|V|$ ), with  $k + 1$  internal vertices. By the above hypothesis, we can compute an  $f(k + 1)$ -approximation of that instance of MCMM. Then we use  $g$  to build in polynomial time a set cover which is, by Lemma 3.7, of the same size as the approximate solution to MCMM, thus providing at most an  $f(k + 1)$ -approximation of the solution of the Minimum Set Cover on  $G$ .

Then, if  $f(N)$  was asymptotically smaller than  $\log(N - 1)(1 - \varepsilon)$ , the corresponding approximation ratio for *Set Cover* would be better than  $\log(k)(1 - \varepsilon)$ , contradiction unless  $P=NP$  [6].  $\square$

## 4 MCMM is FPT when parameterized by the maximum size of a matching in the input graph

This section is devoted to proving the following result:

**Theorem 4.1.** *MCMM is FPT when parameterized by the size of a maximum matching in the input.*

To show this, we adapt the construction of an exploration tree presented in [10] for the labelled version of the problem.

Let  $G^c = (V, E, c)$  be a vertex-colored graph with maximum matching size  $k$ . We consider an arbitrary maximum matching  $M_0$  of  $G$  (which can be built in polynomial time). It will be used as a reference to decompose other matchings. In order to do so, we write  $I_0 = V(G) \setminus V(M_0)$ , and  $G[M_0]$  the subgraph induced by  $V(M_0)$  in  $G$ .

If we consider a maximum matching  $M^*$ , each edge of  $M^*$  has at least one shared vertex extremity with  $M_0$  (otherwise  $M_0$  would not be a maximum matching). Thus, we can split the edges of  $M^*$  into two parts, the one included in  $G[M_0]$

and the remaining ones. We use that property to decompose the search for an optimal solution.

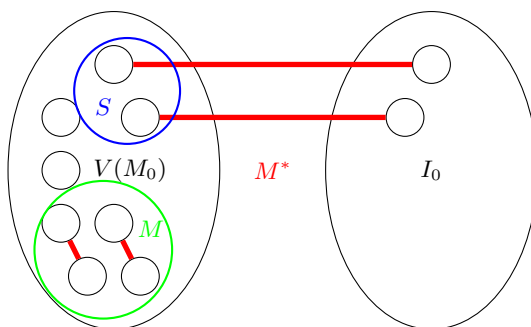
Similarly, we use other “natural” splits to decompose the configuration space we want to explore (*i.e.*, the set of every possible maximum matching). For the first splits, we remain exhaustive (as these do not incur significant costs). Then, we make choices that break exhaustivity, and we will have to prove afterward that even if some optimal solutions are missed, we cannot miss all of them.

Formally, we do so by building a rooted exploration tree, where each node represents the subset of matchings compatible with the choices made along the path from the root to the node. As such, every descendant of a node will represent a subset of matchings of those of its ancestors. The construction is performed as follows:

We create the root  $\omega_0$  which represents all possible maximum matchings, since no choice has been made so far. Every other vertex of the exploration tree will be given a label that contains the choices made at that level.

From the root we branch for every possible selection  $(M, S)$  where  $M \subset E(G[M_0])$  is a matching,  $S \subset V(G[M_0]) \setminus V(M)$ , and  $|M| + |S| = k$ , by adding a child  $\omega_{M,S}$  labeled  $(M, S)$ .

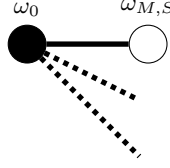
The exploration tree vertex  $\omega_{M,S}$  represents the set of all maximum matchings compatible with the choice of the sets  $M$  and  $S$ . Each of these matchings contains  $M$ , and every other edge of those matchings has one end in  $S$  (and the other in  $I_0$ ) (see Figure 3).



**Fig. 3.** Decomposition of a matching  $M^*$  according to the structure of  $M_0$ .

Note that the condition  $|M| + |S| = k$  arises from the fact that we are searching for a matching of size  $k$ , which includes all edges in  $M$  and an edge for each vertex in  $S$ . In this branch, and for every future branching under it, we will write  $C = c(M) \cup c(S)$  for readability.

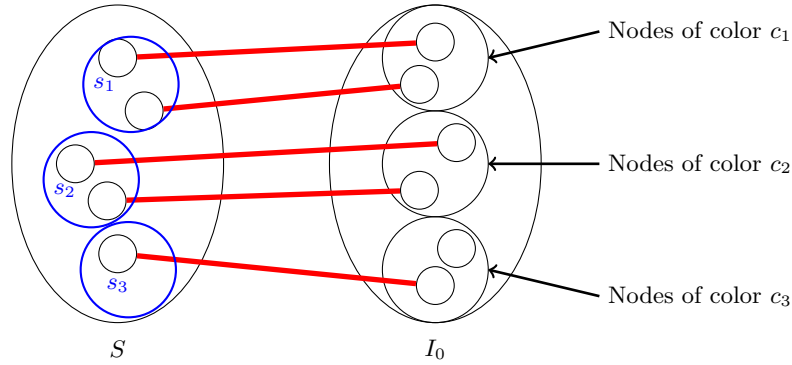
Note that since we branched for every possible choice, the sets of matchings represented by the children of the root form a partition of the set of all possible matchings.



**Fig. 4.** The exploration tree after adding the first layer of vertices.

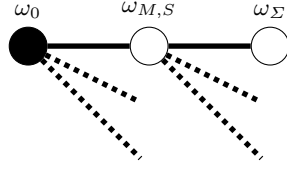
*Observation:* At this point, we have created at most  $T_k \binom{2k}{k}$  new leaves where  $T_i$  is the  $i$ -th telephone number (the number of possible matchings in a clique of size  $i$ ). This enumeration can be completed in time  $O(k \times T_k 2^{2k})$  ( $O(k)$  for each distinct choice).

Then, we want to consider the partition of  $S$  according to the color of the matching vertex in  $I_0$  (see Figure 5). To capture every potential such partition, for every leaf  $\omega_{M,S}$  labeled  $(M, S)$  we branch for every partition  $\Sigma$  of  $S$  by adding a child  $\omega_\Sigma$  labeled  $(\Sigma)$ .



**Fig. 5.** The partition  $\Sigma = \{s_1, s_2, s_3\}$  of  $S$  by matching color on the “exterior side” ( $I_0$ ). Red edges are the edges of a matching compatible with the partition  $\Sigma$ .

The set of matchings represented by  $\omega_\Sigma$  is a subset of the one of its father  $\omega_{M,S}$ . It only keeps from its father the matchings that have  $\Sigma$  as a partition of  $S$  when you partition  $S$  with respect to the colors elements of  $S$  are associated with on the  $I_0$  side by the matching. Again, as we branched for every possible choice of  $\Sigma$ , the sets represented by the children of  $\omega_{M,S}$  form a partition of the set represented by their father.



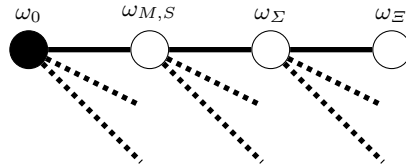
**Fig. 6.** The exploration tree after adding the second layer of vertices.

*Observation :* For each leaf of the exploration tree at the previous step (leaves in Figure 4), we have created the  $B_{|S|} \leq B_k$  possible partitions of  $S$ , where  $B_i$  is the  $i$ -th Bell number (the number of possible partitions of a set with  $i$  elements). They can be enumerated in time  $O(kB_k)$  ( $O(k)$  by distinct partition).

We have to assign a different color to every part of  $\Sigma$ . It will either be a color already in  $C$ , or a new color.

To cover the possible combination of those two options, for every leaf  $\omega_\Sigma$  son of  $\omega_{M,S}$ , we branch for every possible choice of partial injective coloration of nonempty parts of  $\Sigma$  by colors of  $C$ ,  $\Xi$ , by adding a child  $\omega_\Xi$  labeled  $\Xi$ . Parts of  $\Sigma$  that are attributed the value 0 will be attributed a new color (*i.e.* not in  $C$ ) later in constructing the exploration tree.

We formally define  $\Xi$  as a function  $\Xi : \Sigma \rightarrow C \uplus \{0\}$  injective on  $\Sigma \setminus \Xi^{-1}(0)$ .



**Fig. 7.** The exploration tree after adding the third layer of vertices.

The set of matchings represented by  $\omega_\Xi$  is a subset of the one of its father  $\omega_{M,S}$ . It only keeps the matchings that have, for every  $s \in \Sigma$ , the vertices of  $s$  matched with vertices of  $I_0$  of color  $\Xi(s)$  if  $\Xi(s) \neq 0$ , and matched with vertices of the same color not in  $C$  otherwise.

*Observation :* For each leaf of the exploration tree at the previous step (leaves in Figure 6), we have created at most

$$\sum_{i=0}^{\min(|C|, |\Sigma|)} i! \binom{|C|}{i} \leq \min(|C|, |\Sigma|)! \times 2^{|C|} \leq k! \times 2^k$$

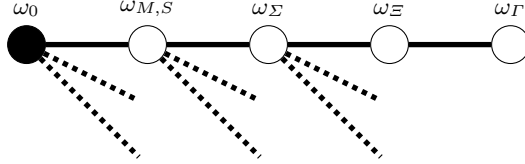
possible partial injective colorations of  $\Sigma$  (and that many new leaves), which can be enumerated in time  $O(k \times k! \times 2^k)$  ( $O(k)$  by distinct coloration).

We want to build partial matchings for every  $s$  of  $\Sigma$ , between  $s$  and  $I_0$ , where every vertex on the  $I_0$  side has the same color.  $\Xi(s)$  if  $\Xi(s) \neq 0$ , any color not in  $C$  otherwise. The goal is to be able to choose one partial matching for every part of the partition, with distinct colors, to compute a maximum matching. Note that if no partial matching exists for some  $s \in \Sigma$ , the choices already made above in the tree do not lead to the construction of a valid maximum matching.

Formally, for every leaf  $\omega_{\Xi}$  produced at the previous step, we compute matchings for every part of the partition  $s \in \Sigma$  (the values of  $M, S$  and  $\Sigma$  are those that appear in the branch from the root to the said leaf):

- If  $\Xi(s) \neq 0$ , we compute, if any,  $\mu$  a matching between  $s$  and vertices of  $I_0$  of color  $\Xi(s)$ , and we write  $\Gamma(s) = \{\mu\}$ . If no such matching exists,  $\Gamma(s) = \emptyset$ .
- If  $\Xi(s) = 0$  then for every color  $c_0 \in c(V)$ , we compute, if any,  $\mu$  a matching between  $s$  and vertices of  $I_0$  of color  $c_0$ , and denote by  $\Gamma(s)$  the set of those matchings truncated at  $k + 1$  (we stop the computation when we already have  $k + 1$  such matchings).

Then we add exactly one child  $\omega_{\Gamma}$  labeled  $\Gamma$  to  $\omega_{\Xi}$ .

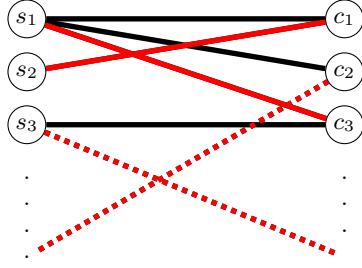


**Fig. 8.** The exploration tree after the addition of the fourth layer of vertices.

The set of matchings represented by  $\omega_{\Gamma}$  is a subset of the one of its father. It only keeps the matchings whose restriction to the edges with an end in  $s \in \Sigma$  is in  $\Gamma(s)$  for every  $s \in \Sigma$ . Do note that we do not maintain exhaustivity here, as some matchings may have been lost in the process.

*Observation :* For each leaf of the exploration tree at the previous step (leaves in Figure 7), for every color, we compute at most a maximum matching, each one being computed in  $O(k^{5/2})$  [2].

Now that we have those partial matchings, we can build the bipartite graph with the elements of  $\Sigma$  on the left side and colors on the right. There is an edge between  $s \in \Sigma$  and a color  $c$  if there is a partial matching with color  $s$  on the  $I_0$  side in  $\Gamma(s)$ . To compute a maximum matching that satisfies the constraints already chosen, it is enough to find a maximum matching of the graph we have just built, then take the union of the partial matchings corresponding to the edges of the maximum matching (see Figure 9).



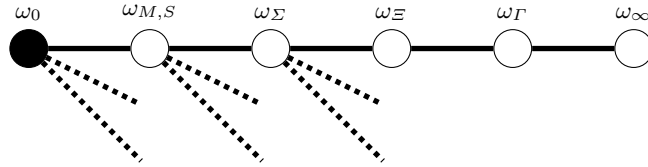
**Fig. 9.** A matching in the bipartite graph with parts  $\Sigma$  to the left and colors of available partial matching to the right.

Formally, for each leaf  $\omega_\Gamma$ , we compute a maximum matching  $\gamma$  on the bipartite graph

$$\left( \Sigma, c \left( \bigcup_{(s, \mu) \in \Sigma \times \Gamma(s)} V(\mu) \cap I_0 \right), \{s\mathcal{C}(\mu) \mid s \in \Sigma, \mu \in \Gamma(s)\} \right),$$

where  $\mathcal{C}(\mu)$  denotes the only color in  $c(V(\mu) \cap I_0)$ . (The values of  $M, S, \Sigma$ , and  $\Xi$  are those that appear in the branch from the root to the said leaf.)

Then we add a child  $\omega_\infty$  to the said leaf. If  $|\gamma| = |\Sigma|$ , we define  $M_\infty = \bigcup_{s\mathcal{C}(\mu) \in \gamma} \mu$ , and we label the child with  $M_\infty$ . Else, it means that we failed to construct a maximum matching with the set of constraints we have, and we label it  $\perp$ .



**Fig. 10.** The completed exploration tree.



Here, the new leaf represents either exactly 1 maximum matching, or the empty set.

*Observation :* For each leaf of the exploration tree at the previous step (leaves in Figure 8), the computation of the auxiliary matching can be done in  $O(k^{5/2})$  [2], the following computation of a matching of  $G$  takes  $O(k^2)$ , and finally the computation of its number of colors  $|c(M_\infty)|$  in  $O(k)$ . It is then a  $O(k^{5/2})$ .

**Lemma 4.1.** *The exploration tree described above can be computed in time  $O(k^4 T_k B_k k! 2^{3k} |V|)$  from a given maximum matching on  $G$ .*

*Proof.* From the analysis boxed between steps of the tree construction, we have that the tree can be computed in

$$O(k^{1/2} 2^{2k} T_k \times (k + B_k(k + k! \times 2^k(k + k \times |c(G)| \times k^{5/2} + k^{5/2})))$$

Which is then  $O(k^4 T_k B_k k! 2^{3k} |V|)$ , (by taking  $|c(G)| = O(|V|)$ ).

*Remark 4.1.* To better visualize that complexity, one can note that for any  $\varepsilon > 0$ , the above is a  $O((\frac{k}{e})^{(3/2+\varepsilon)k} |V|)$ .

**Lemma 4.2.** *There exists a leaf in the exploration tree labeled by a maximum matching whose number of colors is minimal.*

*Proof.* Let  $M_{opt}$  be a minimum colored maximum matching. Let us decompose it relatively to  $M_0$  into  $M_{opt} = M_{in} \uplus M_{out}$  where :

- $M_{in} = M_{opt} \cap E(G[V(M_0)])$
- $M_{out} = M_{opt} \setminus M_{in}$

Note that every edge in  $M_{out}$  must have an end in  $V(M_0)$ , since it would otherwise contradict the maximality of  $M_0$ .

We define  $S_{out} = V(M_{out}) \cap V(M_0)$ . Note that  $S_{out} \cup V(M_{in}) = V(M_0)$ . We then go in the exploration tree to the vertex  $\omega_{M_{in}, S_{out}}$  labeled  $(M_{in}, S_{out})$  which exists since we branched exhaustively on the possible values of  $M$  and  $S$ .

We compute the following partition of  $S_{out}$  :

$$\Sigma = \{\{u \in S_{out} \mid uv \in M_{out}, c(v) = c_0\} \mid c_0 \in c(G)\} \setminus \{\emptyset\}$$

and search among the children of  $\omega_{M_{in}, S_{out}}$  for the child labeled  $\Sigma$ ,  $\omega_\Sigma$ , which exists since we branched on all the possible partitions of  $S_{out}$ .

Then we define  $\Xi(s)$  as the only color in  $c(\{v \mid u \in s, uv \in M_{out}\}) \cap (c(M_{in} \cup c(S_{out}))$  if it is nonempty (there cannot be more than one color in that set from the construction of  $\Sigma$ ), and as 0 otherwise. From the construction of  $\Sigma$ ,  $\Xi$  is injective on  $\Sigma \setminus \Xi^{-1}(0)$ . We search for  $\omega_\Xi$  the child of  $\omega_\Sigma$  labeled  $\Xi$ , which exists since we branched on all possible partial permutations of already chosen colors on the parts of the partition. By construction  $\omega_\Sigma$  have one child  $\omega_\Gamma$ . Let us consider the maximum matching computed in the fifth step in creating the

exploration tree between the parts of the partition  $\Sigma$  and the colors. Recall that, formally, it is a matching in the following bipartite graph:

$$\Omega = (\Sigma, c(\bigcup_{(s,\mu) \in \Sigma \times \Gamma(s)} V(\mu) \cap I_0), \{s\mathcal{C}(\mu) \mid s \in \Sigma, \mu \in \Gamma(s)\}),$$

where  $\mathcal{C}(\mu)$  denotes the only color in  $c(V(\mu) \cap I_0)$

We know that the computed maximum matching is of size  $|\Sigma|$ , since we can construct the following matching :

- For all  $s \in \Sigma$  that have  $k$  or less edges in  $\Omega$ , we take the edge corresponding to the color attributed to  $s$  in  $M_{OPT}$  (that is the only color in  $c(\{u \mid uv \in M_{OPT}, v \in s\})$ ). That color appears in  $\Omega$  since we exhaustively enumerated possible colors for  $s \in \Sigma$  with less than  $k$  possible color to match. Let us denote by  $n_{matched}$  the number of  $s \in \Sigma$  in this situation.
- Then, there are at most  $k - n_{matched}$  parts of  $\Sigma$  that still needs to be matched. For every one of those  $s$ ,  $\Gamma(s)$  contains  $k + 1$  matchings of different colors on the  $I_0$  side, that is for every one of those  $s$ , it has edges to  $k + 1$  colors in  $\Omega$ , at least  $k + 1 - |n_{matched}| > k - |n_{matched}|$  of them not being already matched. We can then choose greedily a different color to match every remaining  $s \in \Sigma$ .

The described matching is of size  $|\Sigma|$ , so the maximum matching computed when constructing the exploration tree must have size  $|\Sigma|$ . Then, by construction of the exploration tree, the only child of  $\omega_\Gamma$  cannot be labeled  $\perp$ , and is labeled with a maximum matching  $M_\infty$ . In that matching, the parts of  $\Sigma$  that are matched to colors already in  $M_{in}$  or in  $S_{out}$  are the same as in  $M_{OPT}$  (since there is only one edge from those in  $\Omega$ ). Every other part of  $\Sigma$  is matched in  $M_\infty$  to a different new color (a color not appearing in  $M_{in}$  or  $S_{out}$ ) as it is the case in  $M_{OPT}$  by construction of  $\Sigma$ . Thus  $M_\infty$  has the same number of colors as  $M_{OPT}$ , which concludes the proof.

#### Proof of Theorem 4.1

Since we supposed that  $G$  admits a maximum matching of size  $k$ , we search for such a matching  $M_0$  in time  $O(|E|\sqrt{|V|})$  [11]. We construct the exploration tree described above and then search for a leaf not labeled  $\perp$  with minimum number of colors in the matching of its label in time  $O(k^4 T_k B_k k! 2^k |V|)$  (from Lemma 4.1). From Lemma 4.2, such a matching is a minimum-colored maximum one of  $G^c$ . Thus, the algorithm above runs in time  $O(|E|\sqrt{|V|}) + O(k^4 T_k B_k k! 2^k |V|)$ . Therefore, Theorem 4.1 holds.  $\square$

## 5 APX-completeness on collections of $P_1$ and $P_2$

In this section, we prove that MCMM restricted to collections of  $P_1$  and  $P_2$  (paths of length 1 and 2 respectively) is APX-complete.

APX-completeness is defined under *approximation-preserving reductions* (AP-reductions). In the case of minimization optimization problems, *linear reductions* (L-reductions) happens to be also AP-reduction. As they are easier to handle, we will work here with linear reductions. Let us then define linear reductions.

For an optimization problem  $A$ , we denote by  $c_A$  its cost function, and when  $x$  is an instance of problem  $A$ ,  $OPT_A(x)$  is the minimum cost of a solution of problem  $A$  on  $x$ .

Then if  $A$  and  $B$  are optimization problems, and we have  $\alpha > 0$  and  $\beta > 0$  two constants, a linear reduction of ratios  $(\alpha, \beta)$  from  $A$  to  $B$  is a pair of functions  $(f, g)$  such that:

1.  $f$  and  $g$  are computable in polynomial time,
2. If  $x$  is an instance of  $A$ , then  $f(x)$  is an instance of  $B$ ,
3. If  $y$  is a solution to problem  $B$  on  $f(x)$ ,  $g(y)$  is a solution to problem  $A$  on  $x$ ,
4.  $OPT_B(f(x)) \leq \alpha OPT_A(x)$ ,
5.  $|OPT_A(x) - c_A(g(y))| \leq \beta |OPT_B(f(x)) - c_B(y)|$ .

Here, we will only consider linear reductions of ratios  $(1, 1)$ .

As there is a natural reduction from this problem to MCMM restricted to collections of  $P_2$  and  $P_3$  to this problem, we introduce a variation of Minimum Vertex Cover: *Minimum Vertex Cover with mandatory vertices*.

MINIMUM VERTEX COVER WITH MANDATORY VERTICES

*Input:* A graph  $G = (V, E)$ , and  $V_0 \subset V$

*Output:* A minimal vertex cover  $C$  of  $G$  such that  $V_0 \subset C$  with minimum cardinality

This problem is equivalent to the regular Minimum Vertex Cover problem under linear reductions, as we will show.

One direction of this equivalence is trivial as one is a restricted version of the other. The following lemma proves the other direction.

**Lemma 5.1.** *There is a linear reduction from Minimum Vertex Cover with mandatory vertices to Minimum Vertex Cover.*

*Proof.* Let  $G = (V, E)$ ,  $V_0 \subset V$ , be an instance of Minimum Vertex Cover with mandatory vertices. We construct an instance of Minimum Vertex Cover  $G' = (V', E')$  by taking  $V' = V \uplus V_0$  where we will denote  $\bar{v}$  the new copy of  $v \in V_0$  added in  $V'$ , and  $E' = E \cup \{v\bar{v} \mid v \in V_0\}$ . It is trivial to observe that any minimal vertex cover  $S$  of  $G'$  either uses all vertices of  $V_0$  or can be modified into a vertex cover of same size or less that does not use the new vertices by replacing all the  $\bar{v} \in S$  by their corresponding  $v$ . Moreover, for any such minimal vertex cover, it is of exact same size, and we have a linear reduction (with ratios 1 / 1).

Then we can use the existence of those reductions to prove that Minimum Vertex Cover with mandatory vertices is APX-complete.

**Lemma 5.2.** *Minimum Vertex Cover with mandatory vertices is APX-complete.*

*Proof.* By Lemma 5.1, we have a linear reduction in one direction, and there is a trivial linear reduction from Minimum Vertex Cover to Minimum Vertex Cover with mandatory vertices: we take the same graph, and take  $V_0 = \emptyset$ . Minimum Vertex Cover being APX-complete, so is Minimum Vertex Cover with mandatory vertices.

**Proposition 5.1.** *MCMM on collections of  $P_1$  and  $P_2$  is APX-complete.*

*Proof.* On one hand, there is a linear reduction from Minimum Vertex Cover to MCMM on collections of  $P_1$  and  $P_2$ . For an instance of Minimum Vertex Cover  $G = (V, E)$ , we construct  $G' = (V', E')$  colored with  $c$  by taking:

- $V' = \{x_{u,uv}, x_{uv}, x_{v,uv} \mid uv \in V\}$ ,
- $E' = \{x_{u,uv}x_{uv}, x_{uv}x_{v,uv} \mid uv \in V\}$ ,
- $c(x_{u,uv}) = u, c(x_{uv}) = 0$ , for every  $u$  and  $v$  in  $V$  such that those nodes exist ( $c(G) = V \uplus \{0\}$ ).

If  $M$  is a maximum matching of  $G'^c$ ,  $S = \{u \in V \mid \exists v \in V, x_{u,uv}x_{uv} \in M\}$  is a vertex cover of  $G$ . If it was not, there would be  $uv \in E$  not covered by  $S$ , which would mean that neither  $x_{u,uv}x_{uv}$  nor  $x_{v,uv}x_{uv}$  are in  $M$ , and thus  $M$  would not be a maximum matching. Thus any maximum matching  $M$  of  $G'^c$  correspond to a vertex cover  $S$  of  $G$  such that  $|c(M)| = |S|$ . We have then a linear reduction (with ratios 1 / 1).

We construct an instance of Vertex Cover with mandatory Vertices  $G' = (V', E')$  with

- $V' = c(V)$ ,
- $E' = \{c(x_{i,1})c(x_{i,3}) \mid i \in \llbracket 1, k \rrbracket\}$ ,
- The mandatory set  $V_0 = \{c(x_{i,2}) \mid i \in \llbracket 1, k \rrbracket\} \cup \{c(y_{i,1}), c(y_{i,2}) \mid i \in \llbracket 1, \ell \rrbracket\}$ .

Then, if  $S$  is a minimal (not necessarily minimum) solution of Vertex Cover with mandatory vertices on  $G'$  with mandatory vertex set  $V_0$ , let us define  $\phi(i)$  as 1 if  $c(x_{i,1}) \in S$ , and 3 otherwise. Observe that when  $\phi(i) = 3$ , as  $S$  is a vertex cover and  $c(x_{i,1})c(x_{i,3}) \in E'$ , we must have  $c(x_{i,3}) \in S$ . We can then define  $M = \{x_{i,\phi(i)}x_{i,2}\} \cup \{y_{i,1}y_{i,2} \mid i \in \llbracket 1, \ell \rrbracket\}$ , and it is a maximum matching since it is of cardinality  $k + \ell$ . We have  $c(M) = S$ , as if it was not the case it would imply that  $S$  is not minimal. Thus, we have a linear reduction (with ratio 1 / 1).

## 6 Conclusion

We have shown that MCMM is NP-hard, W[2]-hard when parameterized by the number of colors in the optimal solution, FPT when parameterized by the size of a maximum matching, and hard to approximate.

These results directly raise several questions. The size of a maximum matching is indeed a “big” parameter, and thus it is not very surprising that MCMM

is FPT with respect to it. However, the classical “small” parameter, treewidth, does not help here, as MCMM is hard even on trees. Is there a sensible parameter, smaller than the size of a maximum matching, for which MCMM is FPT? Additionally, we have given an inapproximability result that gives a lower bound for achievable approximation ratios, but no approximation algorithm that would set an upper bound on the best approximation achievable.

Both questions may lead to further study around the MCMM problem.

## References

- [1] Hans L Bodlaender and Babette de Fluiter. “Intervalizing  $k$ -colored graphs”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 1995, pp. 87–98.
- [2] Bala G. Chandran and Dorit S. Hochbaum. “Practical and theoretical improvements for bipartite matching using the pseudoflow algorithm”. In: *CoRR* abs/1105.1569 (2011).
- [3] Benny Chor, Michael Fellows, Mark A Ragan, Igor Razgon, Frances Rosamond, and Sagi Snir. “Connected coloring completion for general graphs: algorithms and complexity”. In: *Computing and Combinatorics: 13th Annual International Conference, COCOON 2007, Banff, Canada, July 16–19, 2007. Proceedings 13*. Springer. 2007, pp. 75–85.
- [4] J. Cohen, Y. Manoussakis, H.P. Phong, and Zs. Tuza. “Tropical matchings in vertex-colored graphs”. In: *Electronic Notes in Discrete Mathematics* 62 (2017), pp. 219–224.
- [5] J.-A. Angl es d’Auriac, Cs. Bujt as, A. El Maftouhi, M. Karpinski, Y. Manoussakis, L. Montero, N. Narayanan, L. Rosaz, J. Thapper, and Zs. Tuza. “Tropical dominating sets in vertex-coloured graphs”. In: *Journal of Discrete Algorithms* 48 (2018), pp. 27–41.
- [6] Irit Dinur and David Steurer. “Analytical Approach to Parallel Repetition”. In: *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing*. STOC ’14. New York, New York: ACM, 2014, pp. 624–633.
- [7] Rod G. Downey and Michael R. Fellows. “Fixed-Parameter Tractability and Completeness I: Basic Results”. In: *SIAM Journal on Computing* 24.4 (1995), pp. 873–921.
- [8] Jack Edmonds. “Paths, Trees, and Flowers”. In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467.
- [9] Michael R Fellows, Guillaume Fertin, Danny Hermelin, and St ephane Vialette. “Upper and lower bounds for finding connected motifs in vertex-colored graphs”. In: *Journal of Computer and System Sciences* 77.4 (2011), pp. 799–811.
- [10] Michael R. Fellows, Jiong Guo, and Iyad Kanj. “The parameterized complexity of some minimum label problems”. In: *Journal of Computer and System Sciences* 76.8 (2010), pp. 727–740.

- [11] S. Micali and V. V. Vazirani. “An  $O(\sqrt{|V||E|})$  algorithm for finding maximum matching in general graphs”. In: *21st Annual Symposium on Foundations of Computer Science (sfcs 1980)*. 1980, pp. 17–27.
- [12] Shlomo Moran and Sagi Snir. “Convex recolorings of strings and trees: Definitions, hardness results and algorithms”. In: *Workshop on Algorithms and Data Structures*. Springer. 2005, pp. 218–232.