Complexity issues in Vertex-Colored Graph Pattern Matching

Riccardo Dondi, Guillaume Fertin, Stéphane Vialette

Complexity Issues in Vertex-Colored Graph Pattern Matching[☆,☆☆]

Riccardo Dondi[a], Guillaume Fertin[b], Stéphane Vialette[c]

[a] *Dipartimento di Scienze dei Linguaggi, della Comunicazione e degli Studi Culturali Università degli Studi di Bergamo, Piazza Vecchia 8, 24129 Bergamo - Italy*
[b] *Laboratoire d'Informatique de Nantes-Atlantique (LINA), UMR CNRS 6241 Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3 - France*
[c] *LIGM, CNRS UMR 8049, Université Paris-Est Marne-la-Vallée, 5 Bd Descartes 77454 Marne-la-Vallée, France*

Abstract

Searching for motifs in graphs has become a crucial problem in the analysis of biological networks. In the context of metabolic network analysis, Lacroix *et al* [*V. Lacroix, C.G. Fernandes and M.-F. Sagot, IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3 (2006), no. 4, 360368*] introduced the **NP**-hard general problem of finding occurrences of motifs in vertex-colored graphs, where a *motif* $\mathcal{M}$ is a multiset of colors and an occurrence of $\mathcal{M}$ in a vertex-colored graph G, called the *target graph*, is a subset of vertices that induces a connected graph and the multiset of colors induces by this subset is exactly the motif.

Pursuing the line of research pioneered by Lacroix *et al.* and aiming at dealing with approximate solutions, we consider in this paper the above-mentioned problem in two of its natural optimization forms, referred hereafter as the Min-CC and the Maximum Motif problems. The Min CC problem seeks for an occurrence of a motif $\mathcal{M}$ in a vertex-colored graph G that induces a minimum number of connected components whereas the Maximum Motif problem is concerned with finding a maximum cardinality submotif $\mathcal{M}' \subseteq \mathcal{M}$ that occurs as a connected motif in G.

We prove the Min-CC problem to be **APX**-hard even in the extremal case where the motif is a set and the target graph is a path. We complement this result by giving a polynomial-time algorithm in case the motif is built upon a fixed number of colors and the target graph is a path. Also, extending [*M. Fellows, G. Fertin, D. Hermelin, and S. Vialette, Proc. 34th International Colloquium on Automata, Languages and Programming (ICALP), Lecture Notes in Computer Science, vol. 4596, Springer, 2007, pp. 340351*], we prove the Min-CC

[☆] Extended abstracts of this paper appeared in [8] and [9].

[☆☆] Supported by the Italian-French PAI Galileo Project 08484VH

Email addresses: riccardo.dondi@unimib.it (Riccardo Dondi), guillaume.fertin@univ-nantes.fr (Guillaume Fertin), vialette@univ-mlv.fr (Stéphane Vialette)

problem to be fixed-parameter tractable when parameterized by the size of the motif, and we give a faster algorithm in case the target graph is a tree. Furthermore, we prove the MIN-CC problem for trees not to be approximable within ratio $c \log n$ for some constant $c > 0$, where n is the order of the target graph, and to be **W[2]**-hard when parameterized by the number of connected components in the occurrence of the motif. Finally, we give an exact exponential-time algorithm for the MIN-CC problem in case the target graph is a tree.

We prove that the MAXIMUM MOTIF problem is **APX**-hard even in the case where the target graph is a tree of maximum degree 3, the motif is actually a set and each color occurs at most twice in the tree. Next, we strengthen this result by proving that the problem is not approximable within factor $2^{\log^{\delta} n}$, for any constant $\delta < 1$, unless $\mathbf{NP} \subseteq \mathbf{DTIME}(2^{\mathrm{poly}\log n})$. We complement these results by presenting two fixed-parameter algorithms for the problem, where the parameter is the size of the solution. Finally, we give exact exponential-time algorithms for this problem.

---

1. Introduction

Searching for motifs in graphs has become a crucial problem in the analysis of biological networks (*e.g.* protein-protein interaction, regulatory and metabolic networks). Roughly speaking, there exist two different views of graph motifs. Topological motifs (patterns occurring in the network) are the classical view [22, 23, 30–32] and computationally reduce to graph isomorphism, in the broad meaning of that term. These motifs have recently been identified as basic modules of molecular information processing. By way of contrast, functional motifs, introduced recently by Lacroix *et al.* [24], do not rely on the key concept of topology conservation but focus on connectedness of the network vertices sought. This latter approach has been considered in subsequent papers [2, 6, 12]. Formally, searching for a functional motif reduces to the following graph problem (referred hereafter as GRAPH MOTIF) [24]: Given a target vertex-colored graph $G = (V, E)$ and a multiset of colors $\mathcal{M}$ of size k, find a subset $V' \subseteq V$, $|V'| = k$ $(= |\mathcal{M}|)$ such that (i) the vertex induced subgraph $G[V']$ is connected and (ii) there exists a color-preserving bijective mapping from $\mathcal{M}$ to V'.

The GRAPH MOTIF problem is **NP**-complete even if G is a tree with maximum degree 3 and $\mathcal{M}$ is a set [12]. **NP**-completeness has also been shown in case G is a bipartite graph with maximum degree 4 and $\mathcal{M}$ is built over two colors only [12]. The seeming intractability of the GRAPH MOTIF problem has naturally led to parameterized complexity considerations [10]. The GRAPH MOTIF problem can be solved in $\mathcal{O}(4.32^k\, k^2\, m)$ randomized time [2], where m is the number of edges in G, and in $\mathcal{O}(n^{2c\omega+2})$ time [12], where ω is the treewidth of G and c is the number of distinct colors in $\mathcal{M}$. When the number of distinct colors in the motif is taken as a parameter, the GRAPH MOTIF problem is, however, **W**[**1**]-hard even in case G is a tree.

Aiming at accurate models, variants of the GRAPH MOTIF problem are greatly needed. To this aim, Betzler *et al.* [2] replaced the connectedness

demand by more robust requirements, and proved the problem of finding a biconnected occurrence of $\mathcal{M}$ in G to be **W[1]**-complete when the parameter is the size of the motif. This result is of particular importance as it sheds light on the fact that a seemingly small step towards motif topology results in parameterized intractability.

Following the investigations of [2, 12, 24], we discuss in this paper issues about formulations of the Graph Motif problem from an optimization point of view. More precisely, we introduce two optimization problems (a minimization one and a maximization one) that deal with approximate occurrences: Min-CC and Maximum Motif.

The Min-CC problem is concerned with minimizing the number of connected components in $G[V']$, *i.e.*, finding an occurrence of $\mathcal{M}$ in G that results in as few connected components as possible. Being a natural generalization of the Graph Motif problem, this problem is clearly **NP**-hard as well. As we will see soon, the Min-CC problem is **APX**-hard even in the extremal case where the motif is a set and the target graph is a path and is not approximable within ratio $c \log n$ for some constant $c > 0$ if the target graph is a tree, where n is the order of the target graph. From a parameterized point of view, the Min-CC problem is, however, fixed-parameter tractable when the parameter is the size of the motif but becomes **W[2]**-hard when the parameter is the number of connected components in the occurrence of the motif (the problem is, however, only known to be **W[1]**-hard for paths [2]).

The Maximum Motif problem, a natural **NP**-hard dual variant of the Graph Motif problem, is concerned with finding a maximum cardinality submotif $\mathcal{M}' \subseteq \mathcal{M}$ that occurs as a connected motif in G. Notice that, although both the Min-CC and Maximum Motif problems deal with approximate solutions, the Min-CC problem focusses on spread-out solutions whereas the Maximum Motif problem considers approximate motifs. We prove here that the Maximum Motif problem is **APX**-hard even in the case where the target graph is a tree of maximum degree 3, the motif is colorful and each color has at most 2 occurrences in the tree. Then, we strengthen this result by proving that the problem is not in **APX** and not approximable within factor $2^{\log^{\delta} n}$ unless **NP** $\subseteq$ **DTIME** $(2^{\text{poly} \log n})$. Finally, we give two fixed-parameter algorithms for the Maximum Motif problem problem.

This paper is organized as follows. We recall basic definitions in Section 2. Section 3 is devoted to the Min CC problem and Section 4 to the Maximum Motif problem.

2. Preliminaries

In this section we introduce preliminary definitions that will be useful in the rest of the paper. Let $G = (V, E)$ be a graph. For any $V' \subseteq V$, we denote by $G[V']$ the *subgraph of G induced by V'*, that is $G[V'] = (V', E')$ and $\{u, v\} \in E'$ if and only if $u, v \in V'$ and $\{u, v\} \in \mathbf{E}(G)$. Let $v \in V$, we denote by $N(v)$ the set of vertices $u \in V$ such that $\{u, v\} \in E$. Let $V' \subseteq V$, we denote by $N(V')$ the set of vertices $u \in (V \setminus V')$ such that $\{u, v\} \in E$ for some $v \in V'$.

A *coloring* of G is a mapping $\lambda : V \to \mathbf{C}$, where $\mathbf{C}$ is a set of colors. For any subset V' of V, we let $\mathbf{C}(V')$ stand for the multiset of colors assigned to the vertices in V'. A motif $\mathcal{M}$ is a multiset of colors built over a set of colors $\mathbf{C}$. In case $\mathcal{M}$ is actually a set, we call it a *colorful motif*. An *occurrence* of $\mathcal{M}$ in G is a subset $V' \subseteq V$ such that (i) $G[V']$ is connected, and (ii) $\mathbf{C}(V') = \mathcal{M}$. A *color-preserving injective mapping* θ of $\mathcal{M}$ to G (equipped with a coloring λ) is an injective mapping $\theta : \mathcal{M} \to \mathbf{V}(G)$, such that $\lambda(\theta(c)) = c$ for every $c \in \mathcal{M}$. The subgraph induced by a color-preserving injective mapping $\theta : \mathcal{M} \to \mathbf{V}(G)$ is the subgraph of G induced by the images of θ in G.

A tree where a root has been specified is called a *rooted tree*. In a rooted tree with root r, for every non-root node x, let e_x be the unique edge incident to x that lies on the path from x to r. Then e_x can be thought of as connecting each node x to its *parent*. Two vertices with the same parent are said to be *siblings*. Rooted trees can also be considered as directed in the sense that all edges connect parents to their *children*. Given this parent-child relationship, a *descendant* of a node x in a directed tree is defined as any other node reachable from x.

We can now define the two problems we are interested in.

MIN-CC

- **Input** : A vertex colored graph G and a colored motif $\mathcal{M}$.
- **Solution** : A color-preserving injective mapping $\theta : \mathcal{M} \to \mathbf{V}(G)$, *i.e.*, $\lambda(\theta(c)) = c$ for every $c \in \mathcal{M}$.
- **Measure** : The number of connected components in the subgraph induced by θ.

In other words, the MIN-CC problem asks to find a subset $V' \subseteq \mathbf{V}(G)$ that matches $\mathcal{M}$, and that minimizes the number of connected components of $G[V']$. It is thus **NP**-complete even if the target graph is a tree and the occurrence is required to be connected (*i.e.*, the occurrence of $\mathcal{M}$ in G results in one connected component) [24].

MAXIMUM MOTIF

- **Input** : A vertex colored graph G and a colored motif $\mathcal{M}$.
- **Solution** : A submotif $\mathcal{M}' \subseteq \mathcal{M}$ and a color-preserving injective mapping θ of $\mathcal{M}'$ to G such that the subgraph induced by θ is connected.
- **Measure** : The size of the submotif $\mathcal{M}'$

Intuitively, the MAXIMUM MOTIF problem thus asks for the largest submotif $\mathcal{M}' \subseteq \mathcal{M}$ that occurs in G as a connected component. Being a mere restriction of the GRAPH MOTIF problem, the MAXIMUM MOTIF problem is **NP**-complete as well [24].

3. Minimizing the number of connected components: the Min-CC problem

We consider here the MIN-CC problem. We show that this problem is **APX**-hard for paths but is fixed-parameter tractable when parameterized by the size of motif (it is worth mentioning that the MIN-CC problem for paths with respect to the parameter "*number of components*" is **W[1]**-hard [2]). In addition, we give a faster fixed-parameter algorithms for trees and a polynomial-time one for paths with a fixed number of colors. Finally, we show that the MIN-CC problem for trees with respect to the parameter "*number of components*" is **W[2]**-hard and present an exact - but exponential - algorithm for trees.

3.1. Hardness result for paths

In this section we show that the MIN-CC problem is **APX**-hard even in the simple case where the motif $\mathcal{M}$ is a set and the target graph is a path in which each color in $\mathbf{C}$ occurs at most twice. Our proof consists in a reduction from a restricted version of the PAINTSHOP-FOR-WORDS problem [4, 5, 11].

First, we need some additional definitions. Define an *isogram* to be a word in which no letter is used more than once (also referred to as p-sequences in the literature, see e.g. [13, 16]). A 2-*isogram* is a word in which each letter occurs exactly twice. A *cover* of size k of a word u is an ordered collection of words $C = (v_1, v_2, \ldots, v_k)$ such that $u = w_1 v_1 w_2 v_2 \ldots w_k v_k w_{k+1}$ and $v = v_1 v_2 \ldots v_k$ is an isogram The cover is called *prefix* (resp. *suffix*) if w_1 (resp. w_{k+1}) is the empty word.

A *proper* 2-*coloring* of a 2-isogram u is an assignment f of colors c_1 and c_2 to the letters of u such that every letter of u is colored with color c_1 once and colored with color c_2 once. If two adjacent letters x and y are colored with different colors we say that there is a *color change* between x and y (and that the color change occurs at the position of y). For the sake of brevity, we denote a 2-isogram u together with a proper 2-coloring f of it as the pair (u, f).

The 1-REGULAR-2-COLORS-PAINT-SHOP problem is defined as follows: Given a 2-isogram u, find a 2-coloring f of u that minimizes the number of color changes in (u, f). Bonsma [4] proved that the 1-REGULAR-2-COLORS-PAINT-SHOP problem is **APX**-hard. We show here how to reduce the 1-REGULAR-2-COLORS-PAINT-SHOP problem to the MIN-CC problem for paths. We need the following easy lemmas.

Lemma 1. *Let u be a 2-isogram and C be a minimum cardinality cover of u. Then C cannot be both prefix and suffix.*

PROOF. Suppose that there exists a minimum cardinality cover C of u that is both prefix and suffix. Write $C = (v_1, v_2, \ldots, v_k)$. Then u can be written as $u = v_1 w_1 v_2 w_2 \ldots w_{k-1} v_k$ for appropriate words $w_1, w_2, \ldots, w_{k-1}$. By definition, $v = v_1 v_2 ... v_k$ is an isogram. But, since u is a 2-isogram, then $w = w_1 w_2 ... w_{k-1}$ is an isogram as well. Hence it follows that $C' = (w_1, w_2, ..., w_{k-1})$ is a cover of u of size $k - 1$, a contradiction. □

It follows easily from the proof of the above lemma that a 2-isogram has a minimum cardinality prefix cover if and only if it has a minimum cardinality suffix cover, and that it has a minimum cardinality prefix or suffix cover if and only if it has not a minimum cardinality cover which is neither prefix nor suffix.

Lemma 2. *A 2-isogram has a proper 2-coloring with at most k color changes if and only if it has a cover of size at most $\lceil \frac{k}{2} \rceil$.*

PROOF. Let u be a 2-isogram and denote the i-th letter of u by $u[i]$. For ease of exposition, write $n = |u|$ and $l = \lceil \frac{k}{2} \rceil$. Let f be a proper 2-coloring of u with k color changes and suppose that the color changes occur at positions $i_1 < i_2 < \ldots < i_k$ in u. Without loss of generality, assume that the first letter of u is colored with color c_1 by f. Define the collection of words $C = (v_1, v_2, \ldots, v_l)$ as follows:

$$v_j = u[i_{2j-1}]u[i_{2j-1}+1]\ldots u[i_{2j}-1] \; j = 1, 2, \ldots, l-1$$

$$v_l = \begin{cases} u[i_{k-1}]u[i_{k-1}+1]\ldots u[i_k-1] & \text{if } k \text{ is even} \\ u[i_k]u[i_k+1]\ldots u[n] & \text{if } k \text{ is odd.} \end{cases}$$

We claim that C is a cover of u. Indeed, by construction $v = v_1 v_2 \ldots v_l$ is a monochromatic word that contains all the letters in u colored with color c_2 by f. Since f is a proper 2-coloring of u, exactly one occurrence of each letter is colored with color c_2, and hence C is a cover of u of size $l = \lceil \frac{k}{2} \rceil$.

Conversely, let $C = (v_1, v_2, \ldots, v_l)$ be a cover of u. By definition, the word $v = v_1 v_2 \ldots v_l$ is an isogram and u can be written as $u = w_1 v_1 w_2 v_2 \ldots w_l v_l w_{l+1}$ for appropriate - possibly empty - words $w_1, w_2, \ldots, w_{l+1}$. Define a 2-coloring f of u as follows: color all the letters of the words in C with color c_2 and all the other letters with color c_1. By construction, f is a proper 2-coloring of u. According to the proof of Lemma 1, there is no loss of generality in assuming that (i) C is suffix or (ii) C is neither prefix nor suffix. If C is neither prefix nor suffix then f is a proper 2-coloring of u with $2l$ color changes. If C is suffix then f is a proper 2-coloring of u with $2l-1$ color changes. □

Combining Lemma 2 with the fact that the 1-REGULAR-2-COLORS-PAINT-SHOP problem is **APX**-hard, we have the following result.

Proposition 1. *The following problem is* **APX***-hard: Given a 2-isogram u, find a minimum cardinality cover of u.*

Corollary 1. *The* MIN-CC *problem is* **APX***-hard even if $\mathcal{M}$ is a set and P is a path in which each color appears at most twice.*

PROOF. Let u be a 2-isogram over the alphabet Σ. Starting from u construct an instance of the MIN-CC problem as follows. Let $\mathcal{M} = \Sigma$ and let $P = (x_1, x_2, x_{|u|})$ be a vertex-colored path, where each vertex x_i is colored $u[i]$, $1 \leq i \leq |u|$. Since u is a 2-isogram, each color of Σ occurs at most twice. Now, it is a simple matter to check that cardinality covers of u are in bijective correspondence with occurrences of $\mathcal{M}$ in P. □

3.2. Fixed-parameter algorithms

Corollary 1 gives us a sharp hardness result for the MIN-CC problem. To complement this negative result, we first prove here that the MIN-CC problem is fixed-parameter tractable [10, 14, 26] when parameterized by the size of the motif $\mathcal{M}$. The algorithm is a straightforward extension of a recent result [12] and is based on the *color-coding* technique [1]. Then, we give a faster fixed-parameter algorithm in case the target graph is a tree.

3.2.1. The MIN-CC *problem is fixed-parameter tractable*

Let G be a graph and k be a positive integer. Recall that a family $\mathcal{F}$ of functions from $\mathbf{V}(G)$ to $\{1, 2, \ldots, k\}$ is *perfect* if for any subset $V \subseteq \mathbf{V}(G)$ of k vertices there is a function $f \in \mathcal{F}$ which is injective on V (see [1]). Let $(G, \mathcal{M})$ be an instance of the MIN-CC problem, where $\mathcal{M}$ is a motif of size k. Then there is an occurrence of $\mathcal{M}$ in G, say $V \subseteq \mathbf{V}(G)$, that results in a minimum number of connected components. Furthermore, suppose we are provided with a perfect family $\mathcal{F}$ of functions from $\mathbf{V}(G)$ to $\{1, 2, \ldots, k\}$. Since $\mathcal{F}$ is perfect, we are guaranteed that at least one function in $\mathcal{F}$ assigns V with k distinct labels. Let $f \in \mathcal{F}$ be such a function. We now turn to defining a dynamic programming table T indexed by vertices of G and subsets of $\{1, 2, \ldots, k\}$. For any $v \in \mathbf{V}(G)$ and any $L \subseteq \{1, 2, \ldots, k\}$, we define $T_L[v]$ to be the family of all submotifs $\mathcal{M}' \subseteq \mathcal{M}$, $|\mathcal{M}'| = |L|$, for which there exists an exact occurrence of $\mathcal{M}'$ in G, say V, such that $v \in V$ and the set of (unique) labels that f assigns to V is exactly L. We need the following lemma.

Lemma 3 ([12]). *For any labeling function $f : \mathbf{V}(G) \to \{1, 2, \ldots, k\}$, there exists a dynamic programming algorithm that computes the table T in $O(2^{5k}kn^2)$ time.*

Now, denote by $\mathcal{P}$ the set of all pairs $(\mathcal{M}', L') \in 2^{\mathcal{M}} \times 2^{L}$ with $|\mathcal{M}'| = |L'|$ such that there exists an exact occurrence of $\mathcal{M}'$ in G, say V', such that the set of (unique) labels that f assigns to V' is exactly L'. Clearly, $|\mathcal{P}| \leq 2^{2k}$. Furthermore, by resorting to any data structure for searching and inserting that guarantees logarithmic time [**?**] (and observing that any two pairs $(\mathcal{M}', L')$ and $(\mathcal{M}'', L'')$ can be compared in $O(k)$ time), one can construct the set $\mathcal{P}$ in $O(nk^2 2^{2k})$ time by running through the table T. Our algorithm now exhaustively considers all subsets of $\mathcal{P}$ of size at most k to find an occurrence of $\mathcal{M}$ in G that results in a minimum number of connected components. The rationale of this approach is that two pairs $(\mathcal{M}', L')$ and $(\mathcal{M}'', L'')$ with $L' \cap L'' = \emptyset$ correspond to non-overlapping occurrences in G. The total time of this latter procedure is certainly upper-bounded by $\sum_{i=1}^{k} k\binom{2^{2k}}{i} \leq k^2 2^{2k^2}$. Summing up and taking into account the time for computing the table T, the running time for a given $f \in \mathcal{F}$ is $O(2^{5k}kn^2 + nk^2 2^{2k} + k^2 2^{2k^2})$.

According to Alon *et al.* [1], we need to use $O(2^{O(k)} \log n)$ functions $f : \mathbf{V}(G) \to \{1, 2, \ldots, k\}$, and such a family $\mathcal{F}$ can be computed in $O(2^{O(k)} n \log n)$ time. For each $f \in \mathcal{F}$ we use the above procedure to determine an occurrence

of $\mathcal{M}$ in G that results in a minimum number of connected components. We have thus proved the following.

Proposition 2. *The* Min-CC *problem is fixed-parameter tractable when parameterized by the size of the motif.*

3.2.2. A faster fixed-parameter algorithm for trees

We proved in Section 3.1 that the Min-CC problem is **APX**-hard even if the target graph is a path. To complement Proposition 2, we give here a dynamic programming algorithm for trees that does not rely on the color-coding technique (notice, however, that the color-coding technique has been recently shown practical, see e.g. [19]).

Let $(G, \mathcal{M})$ be an instance of the Min-CC problem for trees where both G and $\mathcal{M}$ are built upon a set of colors $\mathbf{C}$. Let $k = |\mathcal{M}|$ and $q = |\mathbf{C}|$. Furthermore, for ease of exposition, write $\mathbf{V}(G) = \{1, 2, \ldots, n\}$ and assume G is rooted at some arbitrary vertex $r(G)$.

Our dynamic programming algorithm is basically an exhaustive search procedure. More precisely, it consists in storing - in a bottom-up fashion - for each vertex i of G and each submotif $\mathcal{M}' \subseteq \mathcal{M}$ that occurs in $T(i)$, *i.e.*, the subtree rooted at i, the minimum number of connected components that results in an occurrence of $\mathcal{M}'$ in $T(i)$. More precisely, for each vertex i of G, we compute two dynamic programming tables $X[i]$ and $Y[i]$. Table $X[i]$ stores all pairs $(\mathcal{M}', c)$, where $\mathcal{M}' \subseteq \mathcal{M}$ is a submotif and c is a positive integer, such that (1) there exists an occurrence of $\mathcal{M}'$ in $T(i)$ that matches vertex i, (2) the minimum number of connected components of an occurrence of $\mathcal{M}'$ in $T(i)$ that matches vertex i is c. Table $Y[i]$ stores all pairs $(\mathcal{M}', c)$, where $\mathcal{M}' \subseteq \mathcal{M}$ is a submotif and c is a positive integer, such that (1') there exists an occurrence of $\mathcal{M}'$ in $T(i)$ that *does not match* vertex i, (2') the minimum number of connected components of an occurrence of $\mathcal{M}'$ in $T(i)$ that does not match vertex i is c.

We first claim that both $X[i]$ and $Y[i]$ contain at most k^{q+1} pairs. Indeed, the number of submotifs $\mathcal{M}' \subseteq \mathcal{M}$ is upper-bounded by k^q and any occurrence of any submotif in any subtree of G results in at most k connected components. We now describe how to compute - in a bottom-up fashion - those two dynamic programming tables X and Y. Let i be an internal vertex of G and suppose that vertex i has s_i sons in the subtree $T(i)$ rooted at i, say $\{i_1, i_2, \ldots, i_{s_i}\}$. Notice that $s_i \geq 1$ since i is an internal vertex of G. The entries $X[i]$ and $Y[i]$ are computed thanks to two auxiliary tables W_i and V_i. Table W_i contains s_i entries, one for each son of vertex i in the subtree rooted at i, that are defined as follows:

$$\forall\ 1 \leq j \leq s_i,$$
$$W_i[i_j] = \{(\mathcal{M}', c, 1) : (\mathcal{M}', c) \in X[i_j]\} \cup \{(\mathcal{M}', c, 0) : (\mathcal{M}', c) \in Y[i_j]\}.$$

In other words, we merge $X[i_j]$ and $Y[i_j]$ in $W_i[i_j]$, differentiating the origin of a pair by means of a third element (an integer that is equal to 1 for $X[i_j]$ and 0 for $Y[i_j]$). Clearly, each entry $W_i[i_j]$ contains at most $2k^{q+1}$ triples, and hence

table W_i on the whole contains at most $2\,s_i\,k^{q+1} \leq 2\,n\,k^{q+1}$ triples. Table V_i also contains s_i entries, one for each son of vertex i in the subtree rooted at i, that are computed as follows: $V_i[i_1] = W_i[i_1]$ and

$$\begin{aligned}&\forall\, 2 \leq j \leq s_i,\\ &V_i[i_j] = W_i[i_j] \;\cup\; \{(\mathcal{M}' \cup \mathcal{M}'', c' + c'', r' + r'') \subseteq \mathcal{M} \times k \times k :\\ &\qquad\qquad (\mathcal{M}', c', r') \in W_i[i_j] \text{ and } (\mathcal{M}'', c'', r'') \in V_i[i_{j-1}]\}.\end{aligned}$$

Each entry $V_i[i_j]$ contains at most k^{q+2} triples, and hence table V_i on the whole contains at most $s_i\,k^{q+2} \leq n\,k^{q+2}$ triples. All the needed information is stored in $V_i[i_{s_i}]$, and $X[i]$ and $Y[i]$ can be now computed as follows:

$$\begin{aligned}X[i] &= \{(\mathcal{M}', c - r + 1) : (\mathcal{M}', c, r) \in V_i[i_{s_i}] \text{ and } r > 0\}\\ Y[i] &= \{(\mathcal{M}', c) : (\mathcal{M}', c, 0) \in V_i[i_{s_i}]\}.\end{aligned}$$

The two entries $X[i]$ and $Y[i]$ are next filtered according to the following procedure: for each submotif $\mathcal{M}' \subseteq \mathcal{M}$ that occurs in at least one pair of $X[i]$ (resp. $Y[i]$), we keep in $X[i]$ (resp. $Y[i]$) the pair $(\mathcal{M}', c)$ with the minimum c.

The base cases, *i.e.*, vertex i is a leaf, are defined as follows: $X[i] = \{(\lambda(i), 1)\}$ and $Y[i] = \emptyset$. In other words, $X[i]$ contains exactly one pair $(\mathcal{M}', c)$, where $\mathcal{M}'$ consists of one occurrence of the color associated to vertex i, and $Y[i]$ does not contain any pair. The solution for the Min-CC problem consists in finding a pair $(\mathcal{M}, c)$ in X or Y with minimum c. If such a pair cannot be found in any entry of both X and Y, then the motif $\mathcal{M}$ does not occur in the tree G.

Proposition 3. *The* Min-CC *problem for trees is solvable in* $O(n^2k^{2(q+2)})$ *time, where* n *is the order of the target graph,* k *is the size of the motif* $\mathcal{M}$*, and* q *is the number of distinct colors in* $\mathcal{M}$*.*

Proof. Let us consider any internal vertex i. For ease of exposition, we assume no pointer sharing in the implementation and thus consider that each pair or triple in any dynamic programming table is written in $O(k)$ time. We first observe that the construction of the dynamic programming table W_i can certainly be done in $O(nk^{q+2})$ time. Second, the construction of the dynamic programming table V_i is done in $O(nk^{2(q+2)})$ time. Finally, computing $X[i]$ and $Y[i]$ is done in $O(nk^{q+2})$ time, and so is the time to filter the dynamic programming tables. Therefore, the running time as a whole is in $O(n^2k^{2(q+2)})$. □

The above-mentioned result is of interest in view of the fact that the Min-CC problem for trees parameterized by q is **W[1]**-hard [12].

3.3. A polynomial-time algorithm for paths with a bounded number of colors

We complement here the results of the two preceding subsections by showing that the Min-CC problem for paths is polynomial-time solvable in case the motif is built upon a fixed number of colors. Observe, however, that each color may still have an unbounded number of occurrences in the motif.

In what follows, we describe a dynamic programming algorithm for this case. Suppose the path G under study is such that $\mathbf{V}(G) = \{1, 2, \ldots, n\}$ and $\mathbf{E}(G) = \{\{i, i+1\} | 1 \leq i < n\}$. The basic idea of our approach is as follows. Suppose we are left by the algorithm with the problem of finding an occurrence of a submotif $\mathcal{M}' \subseteq \mathcal{M}$ in the subpath G' of G induced by $\{i, i+1, \ldots, j\}$, $1 \leq i < j \leq n$. Furthermore, suppose that any occurrence of $\mathcal{M}'$ in G' results in at least k' connected components. This minimum number of occurrences k' can be computed as follows. Assume that we have found one leftmost connected component C_{left} of the occurrence of $\mathcal{M}'$ in G' and let i_2, $i \leq i_2 < j$, be the rightmost (according to the natural order of the vertices) vertex of C_{left}. Let $\mathcal{M}''$ be the motif obtained from $\mathcal{M}'$ by subtracting to each color $c_\ell \in \mathbf{C}$ the number of occurrences of color c_ℓ in the leftmost connected component C_{left}. Then the occurrence of $\mathcal{M}'$ in G' is given by C_{left} plus the occurrence of the motif $\mathcal{M}''$ in the subpath G'' of G' induced by $\{i_2 + 1, i_2 + 2, \ldots, j\}$, which results in $k' - 1$ connected components. From an optimization point of view, the problem thus reduces to finding a subpath $\{i_1, i_1+1, \ldots, i_2\}$, $i \leq i_1 \leq i_2 < j$, such that the occurrence of the motif $\mathcal{M}''$ modified according to the colors in $\{i_1, i_1 + 1, \ldots, i_2\}$ in the subpath induced by $\{i_2 + 1, i_2 + 2, \ldots, j\}$ results in a minimum number of connected components.

Let $(G, \mathcal{M})$ be an instance of the Min-CC problem, where G is a (vertex-colored) path built upon the set of colors $\mathbf{C}$, and let $q = |\mathbf{C}|$. We denote by m_i the number of occurrences of color $c_i \in \mathbf{C}$ in $\mathcal{M}$. Clearly, $\sum_{c_i \in \mathbf{C}} m_i = |\mathcal{M}|$. We now introduce our dynamic programming table T. Define $T[i, j; p_1, p_2, \ldots, p_q]$, $1 \leq i \leq j \leq n$ and $0 \leq p_\ell \leq m_\ell$ for $1 \leq \ell \leq q$, to be the minimum number of connected components in the subpath of G that starts at node i, ends at node j and that covers p_ℓ occurrences of color c_ℓ, $1 \leq \ell \leq q$. The base conditions are as follows:

- for all $1 \leq i \leq j \leq n$,
  - $T[i, j; 0, 0, \ldots, 0] = 0$
- for all $1 \leq i \leq n$
  - $T[i, i; p_1, p_2, \ldots, p_q] = \infty$ if $\sum_{1 \leq \ell \leq q} p_\ell > 1$
  - $T[i, i; p_1, p_2, \ldots, p_q] = \infty$ if $\sum_{1 \leq \ell \leq q} p_\ell = 1$ and $\exists 1 \leq k \leq q$ s.t. $p_k = 1$ and $\lambda(i) \neq c_k$
  - $T[i, i; p_1, p_2, \ldots, p_q] = 1$ if $\sum_{1 \leq \ell \leq q} p_\ell = 1$ and $\exists 1 \leq k \leq q$ s.t. $p_k = 1$ and $\lambda(i) = c_k$

The entry $T[i, j; p_1, p_2, \ldots, p_q]$ of the dynamic programming table T can be computed according to the following recurrence

$$T[i, j; p_1, p_2, \ldots, p_q] = \min_{i \leq i_1 \leq i_2 < j} T[i_2 + 1, j; p'_1, p'_2, \ldots, p'_q] + 1 \qquad (1)$$

where each $p'_\ell \geq 0$ is equal to p_ℓ minus the number of occurrences of color c_ℓ in the subpath of G induced by the vertices $\{i_1, i_1 + 1, \ldots, i_2\}$. The optimal solution is clearly stored in $T[1, n; m_1, m_2, \ldots, m_q]$.

We claim that our dynamic programming table T contains $O(n^{q+2})$ entries. Indeed, there are q colors in $\mathcal{M}$, each color $c_i \in \mathbf{C}$ has at most n occurrences in G and we have $O(n^2)$ subpaths in G to consider. We now turn to evaluating the time complexity for computing $T[i, j; p_1, p_2, \ldots, p_q]$. Assuming each entry $T[i', j'; p'_1, p'_2, \ldots, p'_q]$ with $i \leq i' \leq j' \leq j$ and $|j' - i'| < |j - i|$ has already been computed, $T[i, j; p_1, p_2, \ldots, p_q]$ is obtained by taking a minimum number among $O(|j - i + 1|^2) = O(n^2)$ numbers, and hence is $O(n^2)$ time. We have thus proved the following.

Proposition 4. *The* Min-CC *problem for paths is solvable in* $O(n^{q+4})$ *time, where* n *is the number of vertices and* q *is the number of distinct colors in* $\mathcal{M}$*.*

As an immediate consequence of the above proposition, the Min-CC problem is polynomial-time solvable in case the motif $\mathcal{M}$ is built upon a fixed number of colors and the target graph G is a path.

3.4. Hardness of approximation for trees

We investigate in this section approximation issues for restricted instances of the Min-CC problem. Unfortunately, as we shall now prove, it turns out that, even if $\mathcal{M}$ is a set and G is a tree, the Min-CC problem cannot be approximated within ratio $c \log n$ for some constant $c > 0$, where n is the size of the target graph G. As a side result, we prove that the Min-CC problem is **W[2]**-hard when parameterized by the number of connected components of the occurrence of $\mathcal{M}$ in the target graph G.

At the core of our proof is an L-reduction [27] from the Set-Cover problem. Let I be an arbitrary instance of the Set-Cover problem consisting of a universe set $X(I) = \{x_1, x_2, \ldots, x_n\}$ and a collection of sets $\mathcal{S}(I) = S_1, S_2, \ldots, S_m$, each over $X(I)$. For each $1 \leq i \leq m$, write $t_i = |S_i|$ and denote by $e_j(S_i)$, $1 \leq j \leq t_i$, the j-th element of S_i. For ease of exposition, we present the corresponding instance of the Min-CC problem as a rooted tree G. We construct the tree G as follows (see Figure 1). Define a root r and vertices $S'_1, S'_2, \ldots, S'_m$ such that each vertex S'_i is connected to the root r. For each S'_i define the subtree $G(S'_i)$ rooted at S'_i as follows: each vertex S'_i has a unique child S_i and each vertex S_i has children $e_1(S_i), e_2(S_i), \ldots, e_{t_i}(S_i)$. The set of colors $\mathbf{C}$ is defined as follows: $\mathbf{C} = \{c(S_i) : 1 \leq i \leq m\} \cup \{c(x_j) : 1 \leq j \leq n\} \cup \{c(r)\}$. The coloring mapping $\lambda : \mathbf{V}(G) \to \mathbf{C}$ is defined by: $\lambda(S_i) = \lambda(S'_i) = c(S_i)$ for $1 \leq i \leq m$, $\lambda(x_j) = c(x_j)$ for $1 \leq j \leq n$ and $\lambda(r) = c(r)$. The motif $\mathcal{M}$ is the set defined as follows: $\mathcal{M} = \{c(S_i) : 1 \leq i \leq m\} \cup \{c(x_i) : 1 \leq i \leq n\} \cup \{c(r)\}$.

Proposition 5. *For any instance* I *of the* Set-Cover *problem, there exists a solution of size* h *for* I *(*i.e.*, a subset* $\mathcal{S} \subseteq \mathcal{S}(I)$*,* $|\mathcal{S}| = h$*, such that* $\bigcup_{S_i \in \mathcal{S}} S_i = X$*), if and only if there exists an occurrence of* $\mathcal{M}$ *in* G *that results in* $h + 1$ *connected components.*

Proof. The "*only if*" part is obvious, and we only prove the "*if*" part.

Suppose thus that there exists at least one occurrence of $\mathcal{M}$ in G that results in $h + 1$ connected components. We first make the following important

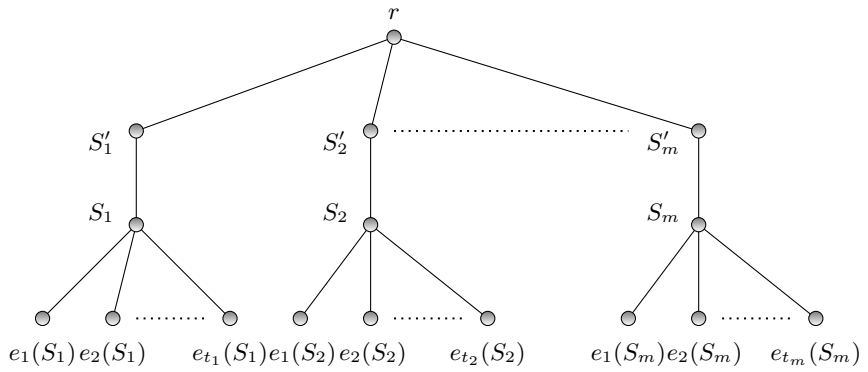

Figure 1: Illustration of the L-reduction from Set-Cover to Min-CC.

observation: for any occurrence V of $\mathcal{M}$ in G, (i) V contains r, (ii) V contains either S_i or S'_i, $1 \leq i \leq m$, and (iii) V contains exactly one vertex colored with color $c(x_j)$, $1 \leq j \leq n$. Now, denote by $\mathcal{V}(\mathcal{M})$ the set of all occurrences of $\mathcal{M}$ in G that result in at most $h+1$ connected components. For each $V \in \mathcal{V}(\mathcal{M})$, define $\ell(V)$ to be the number of connected components induced by V that are solely composed of a leaf of G, *i.e.*, a vertex associated to an element of the universe set $X(I)$. We claim that there exists $V^* \in \mathcal{V}(\mathcal{M})$ such that $\ell(V^*) = 0$. Indeed, suppose for the sake of contradiction that $\ell(V^*) > 0$. Then there exists a connected component induced by V^* that is composed of solely a leaf in G, say x_j. Write $\mathcal{S} \subseteq \mathcal{S}(I)$ the set of all subsets that contain the element x_j in the instance of the Set-Cover problem. If V^* contains a vertex associated to an element of $\mathcal{S}$ we are done. Hence we may now assume that V^* does not contain a vertex associated to an element of $\mathcal{S}$. Then it follows that V^* contains vertex S'_i, for each $S_i \in \mathcal{S}$. Consider the subset $V' \subseteq \mathbf{V}(G)$ which is identical to V except that we add the father of x_i in G, say S_f, and delete the vertex S'_f. It is easily seen that V' is an occurrence of $\mathcal{M}$ in G that results in at most $h+1$ connected components and $\ell(V') < \ell(V^*)$. This is the desired contradiction, and hence no connected component of the occurrence of $\mathcal{M}$ in G is solely composed of a leaf. Then it follows that all the elements of the universe set are covered by h sets in $\mathcal{S}(I)$, which concludes the proof. □

It is easily seen that the above reduction is an L-reduction [27]. It is known that the Set-Cover problem cannot be approximated within ratio $c \log n$ for some constant $c > 0$ [29]. Then it follows that there exists a constant $c' > 0$ such that the Min-CC problem for trees cannot be approximated within performance ratio $c' \log n$, where n is the number of vertices in the target graph.

As a side result, we also observe that the above reduction is a parameterized reduction. Since the Set-Cover problem is **W[2]**-hard when parameterized by the size of the solution [28], the following result holds.

Corollary 2. *The* Min-CC *problem for trees is* $\mathbf{W[2]}$*-hard when parameterized by the number of connected components of the occurrence of the motif in the graph.*

3.5. An exact algorithm for trees

We proved in Section 3.2 that the Min-CC problem for trees is solvable in $O(n^2k^{(q+1)^2+1})$ time, where n is the order of the target tree, k is the size of the motif $\mathcal{M}$ and q is the number of distinct colors in $\mathcal{M}$. We propose here a new algorithm for this special case, which turns out not to be a fixed-parameter algorithm, but has a better running time in case the size k of the motif is not that small compared to the order n of the target graph. More precisely, we give an algorithm for solving the Min-CC problem for trees that runs in $O(n^2 2^{2n/3})$, where n is the order of the target tree.

Let T be the target tree. For any vertex x of T, denote by $T(x)$ the subtree of T rooted at x. The first step of our algorithm is to split the target tree in a *balanced way*, *i.e.*, in such a way that T is rooted at a vertex r having children, $r_1, r_2, \ldots, r_h$ such that none of the trees $T(r_i)$, $1 \leq i \leq h$, has order greater than $\lceil \frac{n}{2} \rceil$. Such a root vertex is called a *centroid*, and it can be shown that every tree has either one centroid or two (in the latter case, the two centroids are connected by an edge) [17]. Goldman [15] and Megiddo *et al.* [25] proposed linear algorithms for finding the centroid of a tree.

From now on, suppose the tree is rooted at a centroid node r. We show how to construct two disjoint subsets R_1 and R_2 of $r_1, r_2, \ldots, r_h$ such that

$$\frac{1}{3}|T| \leq \sum_{r_i \in R_1} |T(r_i)| \leq \left\lceil \frac{1}{2}|T| \right\rceil, \quad \text{and}$$
$$\left\lceil \frac{1}{2}|T| \right\rceil \leq \sum_{r_i \in R_2} |T(r_i)| \leq \frac{2}{3}|T|$$

Let r_i such that $|T(r_i)|$ is maximum. If $|T(r_i)| \geq \frac{1}{3}|T|$, then $T(r_i)$ is in R_1 and all the other subtrees $T(r_j)$ are added to R_2. If $|T(r_i)| \leq \frac{1}{3}|T|$, then perform the following procedure: starting from r_1, add elements to R_1 until $|R_1| \geq \frac{1}{3}|T|$. Thus consider the last element r_ℓ added to R_1. Observe that $|T(r_\ell)| \leq \frac{1}{3}|T|$, since the maximum element has size less than $\frac{1}{3}|T|$. Moreover, $|R_1| - |T(r_\ell)| \leq \frac{1}{3}|T|$, otherwise we would have stopped before adding $T(r_\ell)$ to R_1. It follows that $\frac{1}{3}|T| \leq |R_1| \leq \frac{2}{3}|T|$, and the same holds for R_2. Without loss of generality, we can assume that R_1 is the minimum of sets R_1, R_2.

Given $V' \subseteq V$, we say that V' does not violate $\mathcal{M}$ if the multiset of colors $C(V')$ is a subset of $\mathcal{M}$. Given a subtree T' of T, we define a *partial solution* F for the Min-CC problem over then instance $(T', \mathcal{M})$ as a set of connected components of T' that does not violate the multiset $\mathcal{M}$.

Let S be a partial solution for the Min-CC problem over the instance $(R_i, \mathcal{M})$, $i \in \{1, 2\}$. Thus, S consists of a set of connected components of R_i. This set of connected components can be obtained as follows: determine the subset V' of vertices of R_i that do not belong to S and remove V' and the

edges incident to V' from R_i. It follows that the set of partial solutions for the MIN-CC problem over then instance $(R_i, \mathcal{M})$ can be computed by deleting a (possibly empty) subset of vertices of R_i such that the connected components obtained do not violate $\mathcal{M}$.

We denote by $\mathcal{S}_1$ and $\mathcal{S}_2$ the sets of partial solutions for the MIN-CC problem over the instances $(R_1, \mathcal{M})$ and $(R_2, \mathcal{M})$, respectively. In what follows, we describe how the algorithm computes an optimal solution for the entire tree T starting from the partial solutions for $(R_1,\ \mathcal{M})$ and $(R_2,\ \mathcal{M})$. We need the following definition.

Definition 1. Let S be a feasible solution for the MIN-CC problem over the instance $(T, \mathcal{M})$. The *restriction* of S to a subtree R_1 (resp. R_2) consists of the connected components of S having all the vertices in R_1 (resp. R_2).

It is easy to see that the following property holds for each feasible solution for the MIN-CC problem.

Proposition 6. *Let S be a feasible solution for the* MIN-CC *problem over the instance $(T, \mathcal{M})$. Then, S restricted to R_1 is a partial solution over the instance $(R_1, \mathcal{M})$, and S restricted to R_2 is a partial solution over the instance $(R_2, \mathcal{M})$.*

For each partial solution S_P over the instance $(R_i, \mathcal{M})$, with $i \in \{1, 2\}$, let $C(S_P)$ be the multiset of colors covered by S_P. Recall that the multiset $\mathcal{M}$ is built over the set of colors $\{c_1, c_2, \ldots, c_q\}$. Let f_i be a partial solution having occurrences o^i_j of each color c_j. We represent a multiset as a vector $[o^i_1, o^i_2, \ldots, o^i_q]$. Moreover, since the algorithm needs to distinguish whether a solution contains the root r of the tree T, the algorithm adds a bit at position $q+1$ of the vector o_i for each partial solution f_i. Now $o^i_{q+1} = 1$ if f_i contains r, else $o^i_{q+1} = 0$.

Observe that the we can order the partial solution of a set (for example $\mathcal{S}_1$) on the basis of the covered multisets. The algorithm orders the set of partial solutions $\mathcal{S}_1$ on the basis of their covered multisets. Comparing two multisets requires $O(q)$ time. Hence, the solutions of $\mathcal{S}_1$ can be ordered in time $O(n2^{n/2} \log 2^{n/2}) = O(n^2 2^{n/2})$.

Assume that the partial solutions of $\mathcal{S}_1$ are ordered on the basis of their covered multisets. In what follows we will show how the algorithm rearranges the data in order to compute an optimal solution.

Proposition 7. *Let S be a solution for the* MIN-CC *problem over the instance $(T,\ \mathcal{M})$. Observe that if $r \in S$, then the two restrictions $S_a \in \mathcal{S}_1$ and $S_b \in \mathcal{S}_2$ both contain r; if $r \notin S$, then the two restrictions $S_a \in \mathcal{S}_1$ and $S_b \in \mathcal{S}_2$ both do not contain r.*

From the above property, it follows that we can compute a solution S for the whole tree, by merging two solutions of $\mathcal{S}_1$ and $\mathcal{S}_2$ that both contain r, if $r \in S$, or by merging two solutions of $\mathcal{S}_1$ and $\mathcal{S}_2$ that both do not contain r, if $r \notin S$. It follows that combining two partial solutions $t_1 \in \mathcal{S}_1$ and $t_2 \in \mathcal{S}_2$, we

can restrict ourselves to two kinds of solutions of $\mathcal{S}_2$ for each covered multiset CM:

1. **type 1 solution** covering multiset CM: a solution that covers CM with the minimum number of connected components and that contains r ;
2. **type 2 solution** covering multiset CM: a solution that covers CM with the minimum number of connected components and that does not contain r.

Furthermore, next we have to rearrange the ordered solutions of $\mathcal{S}_1$ such that, for each vector, there exists at most one solution associated to that vector. Note that this implies that there exists at most two partial solutions, a solution of type 1 and a solution of type 2, that cover a certain multiset of colors. Recall that for each multiset CM, the solution of type 1 that covers CM is after the solution of type 2 that covers CM, since the last bit is 1 for type 1 solutions and is 0 for type 2 solutions. The rearrangement of $\mathcal{S}_1$ can be easily computed in time $O(n2^{n/2})$, since each comparison takes $O(n)$ time and since $\mathcal{S}_1$ is ordered.

Observe that the algorithm rearranges the set $\mathcal{S}_1$ such that $\mathcal{S}_1$ contains only solutions of type 1 or type 2. This implies that, for each possible o_i, there exists at most one solution associated to this vector. If such a solution exists, it will be a solution that contains the minimum number of connected components among the solutions associated to vertex o_i.

Now, in order to obtain a solution for T from the set of partial solutions $\mathcal{S}_1$ and $\mathcal{S}_2$, from Proposition 7 it follows that for each solution $t_i^1 \in \mathcal{S}_2$, if t_i^1 contains r, we must look for a solution of type 1 (*i.e.*, that contains r) in the rearrangement of $\mathcal{S}_1$ that covers $\mathcal{M} - c(t_i^1) + c(r)$. Similarly, if t_i^1 does not contain r, we must look for a solution of type 2 that covers $\mathcal{M} - c(t_i^1)$. If such a solution t_j^2 exists, we say that t_j^2 completes t_i^1.

In the first case, the number of connected components of the solution is $|t_i^1| + |t_j^2| - 1$, since the connected component containing root r is counted both in t_i^1 and in t_j^2. In the second case the number of connected components of the solution is $|t_i^1| + |t_j^2|$. Given a partial solution t of $\mathcal{S}_2$, we are able to find a partial solution t' that completes t in time logarithmic in the size of $\mathcal{S}_1$.

4. Maximizing the Number of Colors: the Maximum Motif problem

In this section, we focus on the Maximum Motif problem. We first give in Section 4.1 several results showing that the Maximum Motif problem is hard to approximate. Section 4.2 gives two exact (thus exponential) algorithms for trees. Finally, Section 4.3 gives two FPT algorithms: one for trees, the other for general graphs.

4.1. Hardness of approximation

We first prove here **APX**-hardness of the Maximum Motif problem. Recall that, given a graph $G = (V, E)$, the maximum independent set problem (Independent Set) seeks for a maximum cardinality subset $V' \subseteq V$ such that

no two vertices in V' are joined by an edge. The Independent Set problem is known to be **APX**-hard even when restricted to cubic graphs [27].

Proposition 8. *The* Maximum Motif *problem is* **APX***-hard even if the motif is colorful and the target graph is a tree with maximum degree* 3*.*

Proof. The proof is by L-reduction from the Independent Set problem for cubic graphs. Let $G = (V, E)$ be an instance of the Independent Set problem for cubic graphs. Write $\mathbf{V}(G) = \{v_1, v_2, \ldots, v_n\}$ and $\mathbf{E}(G) = \{e_1, e_2, \ldots, e_m\}$. For each $v_i \in V$, let us denote by $E(v_i)$ the three edges of $\mathbf{E}(G)$ that are incident to v_i. Furthermore, denote by $e(v_i, j)$ the j-th edge of $E(v_i)$, $1 \leq j \leq 3$, where the order is arbitrary. We show how to construct the corresponding instance of the Maximum Motif problem. This instance consists of a vertex-colored tree $T = (V_T, E_T)$ of maximum degree 3 and a colorful motif $\mathcal{M}$. The tree T is defined as follows:

$$\begin{aligned}
V_T =& \{a_i, b_i, x_{i,I}, x_{i,C}, l_i : 1 \leq i \leq n\} \cup \\
& \{d_{i,j}, f_{i,j}, e_{i,j} : 1 \leq i \leq n \wedge 1 \leq j \leq 3\} \\
E_T =& \{\{a_i, b_i\}, \{b_i, x_{i,I}\}, \{b_i, x_{i,C}\}, \{x_{i,C}, d_{i,1}\}, \{x_{i,I}, f_{i,1}\} : 1 \leq i \leq n\} \cup \\
& \{\{a_i, a_{i+1}\} : 1 \leq i < n\} \cup \\
& \{\{d_{i,j}, d_{i,j+1}\}, \{f_{i,j}, f_{i,j+1}\} : 1 \leq i \leq n \wedge 1 \leq j < 3\} \cup \\
& \{\{d_{i,j}, e_{i,j}\} : 1 \leq i \leq n \wedge 1 \leq j \leq 3\}\{\{f_{i,3}, l_i\} : 1 \leq i \leq n\}
\end{aligned}$$

Refer to Figure 2 for a schematic representation of the tree T. Vertex a_i, $1 \leq i \leq n$, is colored $c(a_i)$, vertex b_i, $1 \leq i \leq n$, is colored $c(b_i)$, the two vertices $x_{i,C}$ and $x_{i,I}$, $1 \leq i \leq n$, are colored $c(x_i)$, vertex l_i, $1 \leq i \leq n$, is colored $c(l_i)$, the two vertices $d_{i,j}$ and $f_{i,j}$, $1 \leq i \leq n$ and $1 \leq j \leq 3$, are colored $c(i, j)$, and vertex $e_{i,j}$, $1 \leq i \leq n$ and $1 \leq j \leq 3$, is colored $c(e_k)$, where $e_k = e(v_i, j)$. Write $\mathbf{C}$ for the set of all colors that occur in T (notice that each color in $\mathbf{C}$ occurs at most twice in T). The motif $\mathcal{M}$ is defined by $\mathcal{M} = \mathbf{C}$, and is hence colorful.

We claim that there exists an independent set of size k in G if and only if there exists a submotif of size $6n + m + k$ that occurs in T.

Suppose there exists an independent set V' of size k in G. For each $e = \{v_i, v_j\} \in E$, define $\min(e)$ to be

$$\min(e) = \begin{cases} v_i & \text{if } (v_j \in V') \vee (v_i \notin V' \wedge v_j \notin V' \wedge i < j), \\ v_j & \text{otherwise.} \end{cases}$$

Consider the subset $V'_T \subseteq V_T$ defined by

$$\begin{aligned}
V'_T =& \{a_i, b_i : 1 \leq i \leq n\} \cup \\
& \{x_{i,I}, f_{i,1}, f_{i,2}, f_{i,3}, l_i : v_i \in V'\} \cup \\
& \{x_{i,C}, d_{i,1}, d_{i,2}, d_{i,3} : v_i \notin V'\} \cup \\
& \{e_{i,j} : e \in E \wedge \min(e) = e(v_i, j)\}
\end{aligned}$$

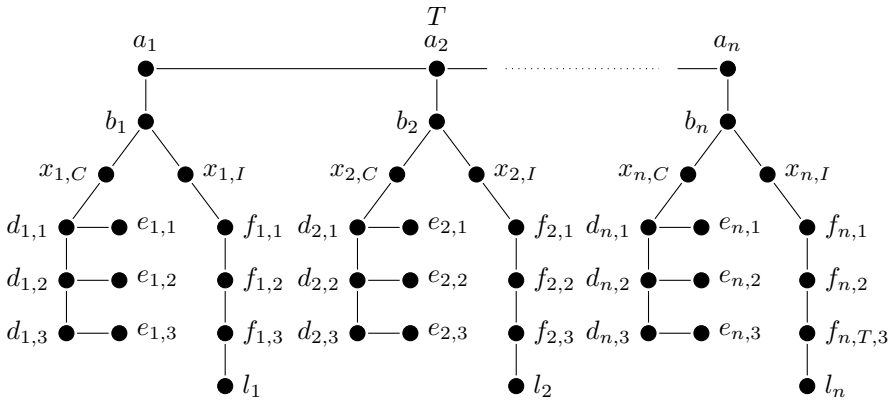

Figure 2: Schematic representation of the tree T described in Proof of Proposition 8.

Observe that V'_T induces a connected component in T. Furthermore, $\mathbf{C}(V'_T) = \mathcal{M}' \subseteq \mathcal{M}$, contains all colors from $\mathcal{M}$ except those $c(l_i)$ with $v_i \notin V'$.

Conversely, suppose that there exists a motif $\mathcal{M}' \subset \mathcal{M}$, $|\mathcal{M}'| \geq 7$, that occurs in T. Fix one occurrence of $\mathcal{M}'$ in T and write $V'_T \subseteq V_T$ for the vertices of T involved in this occurrence. Without loss of generality, suppose that T' is maximal for inclusion (adding any adjacent vertex to T' results in a subtree that is not an occurrence of a submotif of $\mathcal{M}$). Since $|M'| \geq 7$, we can assume that at least one of a_i and b_i belongs to M'. Furthermore, observe that $a_i, b_i \in V'_T$, $1 \leq i \leq n$, since adding any of these missing vertices would result in a larger connected component T'' of T, such that $\mathbf{C}(T'') \subseteq \mathcal{M}$, thereby contradicting the maximality of T'. Then it follows that $c(a_i)$, $c(b_i) \in \mathcal{M}'$, $1 \leq i \leq n$. Moreover, since $\mathcal{M}$ is colorful, V'_T contains at most one of $x_{i,C}$ and $x_{i,I}$, $1 \leq i \leq n$; they indeed both have the same color. Therefore, by the maximality of T', V'_T contains exactly one of $x_{i,C}$ and $x_{i,I}$, $1 \leq i \leq n$, and hence $\mathcal{M}'$ contains the color $c(x_i)$, $1 \leq i \leq n$. Pursuing our maximality argument, if $x_{i,C} \in V'_T$ then V'_T also contains the three vertices $d_{i,j}$, $1 \leq i \leq n$ and $1 \leq j \leq 3$, and if $x_{i,I} \in V'_T$ then V'_T also contains the three vertices $f_{i,j}$, $1 \leq i \leq n$ and $1 \leq j \leq 3$. Therefore, $\mathcal{M}'$ contains the colors $c(i,j)$, $1 \leq i \leq n$ and $1 \leq j \leq 3$. In case $x_{i,I}, f_{i,1}, f_{i,2}, f_{i,3} \in V'_T$, $1 \leq i \leq n$, $l_i \in V'_T$, and hence $\mathcal{M}'$ contains in addition color $c(l_i)$, $1 \leq i \leq n$. We now claim that we may assume that $c(e) \in \mathcal{M}'$ for all $e \in E$, *i.e.*, submotif $\mathcal{M}'$ contains the color associated to each edge of G. Indeed, suppose that for some color $c(e) \in \mathcal{M}$, say $e = \{v_i, v_j\}$, T' has no vertex colored $c(e)$, *i.e.*, $c(e) \notin \mathcal{M}'$. Then, by the maximality of T' (and $\mathcal{M}'$), it follows that $\{x_{i,I}, f_{i,1}, f_{i,2}, f_{i,3}, l_i\} \subseteq V'_T$ and $\{x_{j,I}, f_{j,1}, f_{j,2}, f_{j,3}, l_j\} \subseteq V'_T$, and hence that $\{x_{i,C}, d_{i,1}, d_{i,2}, d_{i,3}\} \cap V'_T = \emptyset$ and $\{x_{j,C}, d_{j,1}, d_{j,2}, d_{j,3}\} \cap V'_T = \emptyset$. Therefore, $V''_T = (V'_T - \{x_{i,I}, f_{i,1}, f_{i,2}, f_{i,3}, l_i\}) \cup \{x_{i,C}, d_{i,1}, d_{i,2}, d_{i,3}\} \cup e_{i,p}$, with $c(e_{i,p}) = c(e)$, induces a subtree in T, and this subtree is an occurrence of $\mathcal{M}'' = (\mathcal{M}' - \{c(l_i)\}) \cup \{c(e)\}$. Applying the above procedure will eventually result in a submotif that contains the color associated to each edge of G. It

follows that $\{v_i : x_{i,C} \in V'_T\}$ is a vertex cover of G, and hence $\{v_i : x_{i,I} \in V'_T\}$ is an independent set in G.

We have thus shown that there is an independent set of size k in G if and only if there exists a submotif of size $6n + m + k$ that occurs in T. But G is a cubic graph, and hence $k \geq \frac{n}{4}$ and $m = \frac{3n}{2}$. Then it follows that the described reduction is indeed an L-reduction [27] from the INDEPENDENT SET problem for cubic graphs to the MAXIMUM MOTIF problem for trees, which proves the proposition. □

We now strengthen the inapproximability of the MAXIMUM MOTIF problem for trees and colorful motifs. More precisely, we show that, for any constant $\delta < 1$, the MAXIMUM MOTIF problem cannot be approximated within factor $2^{\log^{\delta} n}$ in polynomial-time unless **NP** $\subseteq$ **DTIME**$[2^{\text{poly}\log n}]$. The proof is by the *self-improvement* technique (see for example [18, 20, 21]). For the sake of clarity, let us introduce the MAXIMUM LEVEL MOTIF problem which is the restriction of the MAXIMUM MOTIF problem to colorful motifs and rooted trees in which two vertices can have the same color only if they are at the same level (*i.e.*, at the same distance to the root) in the target tree.

First, we show the following easy lemma that will prove useful in the sequel.

Lemma 4. *Let $I = (T, \mathcal{M})$ be an instance of the* MAXIMUM LEVEL MOTIF *problem and T' be a solution for instance I. One can compute in polynomial-time a solution T'' for I, such that (i) $|T''| \geq |T'|$ and (ii) T'' contains the root of T.*

PROOF. Let $T' = (V', E')$ be a solution for the MAXIMUM LEVEL MOTIF problem for instance I, and assume that T' does not contain the root r of T. Notice that T' must be a rooted subtree of T, and let $y \in V'$ be the root of T'. Now consider the unique path $P = (r, x'_1, \ldots, x'_p = y)$, from the root r to y. Two vertices x'_i and x'_j of P, $1 \leq i \neq j \leq p$, have distinct colors, since they belong to different levels of T. Moreover, each vertex x'_i, with $1 \leq i \leq p-1$, has a distinct color from each vertex $v \in V'$, since vertices x'_i and v belong to different levels of T. Define T'' as the subtree of T induced by the set of vertices $V'' = V' \cup \bigcup_{i=1}^{p-1} x_i$. Notice that T'' contains the root r of T, and by construction $|V''| \geq |V'|$. □

It can also be easily seen that Proposition 8 can be modified to prove the following result (this result is indeed is needed for proving, latter, a stronger inapproximabilty result for the MAXIMUM MOTIF problem).

Proposition 9. *The* MAXIMUM LEVEL MOTIF *problem is* **APX***-hard.*

PROOF. We prove that the problem is **APX**-hard by modifying the L-reduction for the MAXIMUM MOTIF problem on bounded tree presented in Proposition 8. Let $G = (V, E)$ be an instance of the INDEPENDENT SET problem on cubic graph. Write $\mathbf{V}(G) = \{v_1, v_2, \ldots, v_n\}$ and $\mathbf{E}(G) = \{e_1, e_2, \ldots, e_m\}$. For each $v_i \in V$, let us denote by $E(v_i)$ the three edges of E that are incident to v_i.

Furthermore, denote by $e(v_i, j)$ the j-th edge of $E(v_i)$, $1 \leq j \leq 3$, where the order is arbitrary. We now show how to construct the corresponding instance of the Maximum Level Motif problem. This instance consists of a rooted vertex-colored tree $T = (V_T, E_T)$ and a colorful motif $\mathcal{M}$.

The tree T is defined as follows:

$$\begin{aligned} V_T =& \{r\} \cup \{b_i, x_{i,I}, x_{i,C}, l_i : 1 \leq i \leq n\} \ \cup \\ & \{e_{i,j} : 1 \leq i \leq n \ \wedge \ 1 \leq j \leq 3\} \\ E_T =& \{\{r, b_i\}, \{b_i, x_{i,I}\}, \{b_i, x_{i,C}\}, : 1 \leq i \leq n\} \ \cup \\ & \{\{x_{i,C}, e_{i,j}\} : 1 \leq i \leq n \ \wedge \ 1 \leq j \leq 3\} \ \cup \\ & \{\{x_{i,I}, l_i\} : 1 \leq i \leq n\} \end{aligned}$$

Moreover, we root T at r.

Vertex r is colored $c(r)$, vertex b_i, $1 \leq i \leq n$, is colored $c(b_i)$, the two vertices $x_{i,C}$ and $x_{i,I}$, $1 \leq i \leq n$, are colored $c(x_i)$, vertex l_i, $1 \leq i \leq n$, is colored $c(l_i)$, vertex $e_{i,j}$, $1 \leq i \leq n$ and $1 \leq j \leq 3$, is colored $c(e_k)$, where $e_k = e(v_i, j)$. Write $\mathbf{C}$ for the set of all colors that occur in T (notice that each color in $\mathbf{C}$ occurs at most two times in T The motif $\mathcal{M}$ is defined by $\mathcal{M} = \mathbf{C}$, and is hence colorful.

First, observe that this is an instance of the Maximum Level Motif problem. Indeed, the tree T is rooted, has 4 levels and all the leaves are at level 4. Two vertices have the same color either if they both are at level 3, *i.e.*, a pair $(x_{i,C}, x_{i,I})$, or if they are both at level 4, *i.e.*, they are leaves associated to the same edge e_k.

In what follows, we show that, given a cubic graph G, there exists a solution of cardinality k for the Independent Set problem on G iff there exists a solution for the Maximum Level Motif problem, of size $1 + 2n + k + m$, on the instance $(T, \mathcal{M})$ built from G.

Let $V' \subseteq V$ be a solution for the Independent Set problem on G, such that $|V'| = k$. We define a solution T' for the Maximum Level Motif problem on $(T, \mathcal{M})$ as follows. Consider the subset $V'_T \subseteq V_T$ defined as by

$$\begin{aligned} V'_T =& \{r\} \cup \{b_i : 1 \leq i \leq n\} \ \cup \\ & \{x_{i,I}, l_i : v_i \in V'\} \ \cup \\ & \{x_{i,C} : v_i \notin V'\} \ \cup \\ & \{e_{i,j} : e \in E \ \wedge \ \min(e) = e(v_i, j)\}. \end{aligned}$$

It is easy to see that V'_T induces a subtree T' in T. Furthermore, $\mathbf{C}(V'_T) = \mathcal{M}' \subseteq \mathcal{M}$ is defined by deleting in $\mathcal{M}$ every color $c(l_i)$ such that $v_i \notin V'$, and it can be seen by definition of V'_T that $|\mathcal{M}'| = 1 + 2n + k + m$.

Conversely, let $T' = (V'_T, E'_T)$ be a solution for the Maximum Level Motif problem, where $|\mathbf{C}(V'_T)| = 1 + 2n + k + m$. Notice that $\mathbf{C}(V') = \mathcal{M}' \subseteq \mathcal{M}$. Without loss of generality, suppose that T' is maximal for inclusion (*i.e.*, adding any adjacent vertex to T' results in a submotif that does not occur in T). By Lemma 4, we can assume that $r \in V'_T$, hence $\mathcal{M}'$ contains color $c(r)$. Furthermore, notice that $b_i \in V'_T$, $1 \leq i \leq n$, since adding any of these missing vertices

would result in a larger connected component T'' of T, such that $\mathbf{C}(T'') \subseteq \mathcal{M}$, thereby contradicting the maximality of T'. Then it follows that $c(b_i) \in \mathcal{M}'$, $1 \leq i \leq n$. Therefore, still by the maximality, V'_T contains exactly one of $x_{i,C}$ and $x_{i,I}$, $1 \leq i \leq n$, and hence $\mathcal{M}'$ contains color $c(x_i)$, $1 \leq i \leq n$. Pursuing our maximality argument, if $x_{i,C} \in V'_T$ then in case $x_{i,I} \in V'_T$, $1 \leq i \leq n$, $l_i \in V'_T$, and hence $\mathcal{M}'$ contains in addition color $c(l_i)$, $1 \leq i \leq n$. Furthermore, we may assume that $c(e) \in \mathcal{M}'$ for all $e \in E$, *i.e.*, submotif $\mathcal{M}'$ contains the color associated to each edge of G. Indeed, suppose that there is no vertex associated to color $c(e)$, say $e = \{v_i, v_j\}$, in T', that is $c(e)$ is not part of $\mathcal{M}'$. Then, by maximality of T' and $\mathcal{M}'$, it follows that $\{x_{i,I}, l_i\} \subseteq V'_T$ and $\{x_{j,I}, l_j\} \subseteq V'_T$, and hence that $x_{i,C}, x_{j,C} \notin V'_T$. Therefore, $V''_T = V'_T - \{x_{i,I}, l_i\} \cup \{x_{i,C}\} \cup e_{i,p}$, with $c(e_{i,p}) = c(e)$, induces a subtree in T and $\mathbf{C}(V''_T) = \mathcal{M}'' = (\mathcal{M}' - \{c(l_i)\}) \cup \{c(e)\}$. Applying the above procedure will eventually result in a submotif that contains the color associated to each edge of G.

It follows that $\{v_i : x_{i,C} \in V'_T\}$ is a vertex cover of G, and hence $\{v_i : x_{i,I} \in V'_T\}$ is an independent set in G. Moreover, since we have characterized precisely which vertices are in V'_T and since $|\mathcal{M}'| = 1 + 2n + k + m$, a simple counting argument allows us to conclude that $|\{v_i : x_{i,I} \in V'_T\}| = k$.

Finally, since G is a cubic graph, we have $k \geq \frac{n}{4}$ and $m = \frac{3n}{2}$. It then follows that the above reduction is indeed an L-reduction [27] from the Independent Set problem for cubic graphs to the Maximum Level Motif problem for trees, which proves the proposition. □

Aiming at applying the self-improvement technique we need to precisely define the product of two instances I_1 and I_2 of the Maximum Level Motif problem. Let $I_1 = (T_1, \mathcal{M}_1)$ and $I_2 = (T_2, \mathcal{M}_2)$ be two instances of the Maximum Level Motif problem, where $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ are vertex-colored trees rooted at r_1 and r_2, respectively. The product $I_1 \times I_2$ is defined to be the instance $(T_{1,2}, \mathcal{M}_{1,2})$ where $T_{1,2} = (V_{1,2}, E_{1,2})$ is a rooted tree defined by $V_{1,2} = \{v_i(v_j) : v_i \in V_1 \ \wedge \ v_j \in V_2\}$ and $E_{1,2} = \{\{v_i(v_{j,1}), v_i(v_{j,2})\} : \{v_{j,1}, v_{j,2}\} \in E_2 \ \wedge \ v_i \in V_1\} \cup \{\{v_i(r_2), v_j(r_2)\} : \{v_i, v_j\} \in E_1\}$, and $\mathcal{M}_{1,2}$ is a motif defined by $\mathcal{M}_{1,2} = \{c_1(c_2) : c_1 \in \mathcal{M}_1 \ \wedge \ c_2 \in \mathcal{M}_2\}$. The tree $T_{1,2}$ is rooted at vertex $r_1(r_2)$. Informally, $T_{1,2}$ is obtained by replacing each vertex $v_i \in V_1$ by a copy of T_2, connecting these copies through their roots. As for the color of each vertex of $T_{1,2}$, if $v_i \in V_i$ is colored c_i and $v_j \in V_j$ is colored c_j then vertex $v_i(v_j) \in T_{1,2}$ is colored $c_i(c_j)$. Denote by $v_i[T_2]$ the subtree of $T_{1,2}$ isomorphic to T_2 rooted at $v_i(r_2)$. Write $V_{1,2,r} = \{v_i(r_2) : v_i \in V_1\}$. Observe that, by construction, the subtree of $T_{1,2}$ induced by $V_{1,2,r}$ is isomorphic to T_1.

Lemma 5. *Let $I_1 = (T_1, \mathcal{M}_1)$ and $I_2 = (T_2, \mathcal{M}_2)$ be two instances of the* Maximum Level Motif *problem. Then $I_1 \times I_2$ is an instance of the* Maximum Level Motif *problem.*

Proof. Write $T_1 = (V_1, E_1)$ and $T_2 = (V_2, E_2)$ and assume that T_1 and T_2 are rooted at r_1 and r_2, respectively. Let $I_1 \times I_2 = (T_{1,2}, \mathcal{M}_{1,2})$ and write $T_{1,2} = (V_{1,2}, E_{1,2})$. First, we note that $T_{1,2}$ is a tree. Indeed, $T_{1,2}[V_{1,2,r}]$ is

isomorphic to T_1 and each vertex in $V_{1,2} - V_{1,2,r}$ belongs to a subtree rooted at some $v_i(r_2) \in V_{1,2,r}$. Furthermore, $T_{1,2}$ is rooted by definition.

Now, we show that two vertices of $T_{1,2}$ have the same color only if they are at the same level in $T_{1,2}$. Let $u_1(u_2)$ and $v_1(v_2)$ be two vertices of $T_{1,2}$ such that $c(u_1(u_2)) = c(v_1(v_2)) = c_a(c_b)$. If $u_1 = v_1$, we are done. Hence, we may now assume $u_1 \neq v_1$. Therefore, we must have $c(u_1) = c(v_1) = c_a$. Furthermore, observe that, by construction, all vertices in $u_1[T_2]$ and $v_1[T_2]$ are colored $c_a(c_x)$ for some color $c_x \in \mathcal{M}$. Consider the subtree $T_{1,2}[V_{1,2,r}]$ induced by $V_{1,2,r}$. Since $T_{1,2}[V_{1,2,r}]$ is isomorphic to T_1, each vertex $x_i(r_2) \in V_{1,2,r}$ has color $c(x_i)(c(r_2))$. Now, since all vertices of $u_1[T_2]$ and $v_1[T_2]$ are colored $c_a(c_x)$, it follows that the root $x_i(r_2)$ of $u_1[T_2]$ and the root $x_j(r_2)$ of $v_1[T_2]$ have the same color $c_a(c(r_2))$. Then it follows that $x_i(r_2)$ and $x_j(r_2)$ must be at the same level l_1 of $T_{1,2}$, since they both belong to $V_{1,2,r}$ and $T_{1,2}[V_{1,2,r}]$ is isomorphic to T_1, where x_i and x_j must be both at level l_1.

Now, consider the subtrees $u_1[T_2]$ and $v_1[T_2]$ isomorphic to T_2. Recall that vertices $u_1(u_2)$ and $v_1(v_2)$ of $T_{1,2}$ are both colored $c_a(c_b)$. As previously observed, all vertices $u_1(u_j)$ in $u_1[T_2]$ and $v_1[T_2]$ are associated to colors $c_a(c(u_j))$ for some $u_j \in V_2$. Since $I_2 = (T_2, \mathcal{M}_2)$ is an instance of the Maximum Level Motif problem, vertices u_2 and v_2 must be at the same level l_2 in T_2 since $c(u_2) = c(v_2) = c_b$. Then, since $u_1[T_2]$ and $v_1[T_2]$ are both isomorphic to T_2, $u_1(u_2)$ and $v_1(v_2)$ are both at level l_2 in $u_1[T_2]$ and $v_1[T_2]$, respectively. It follows that both $u_1(u_2)$ and $v_1(v_2)$ are at level $l_1 + l_2$ in $T_{1,2}$.

Finally, consider the motif $\mathcal{M}_{1,2}$. By construction, $\mathcal{M}_{1,2}$ is a set, hence it is colorful. □

For any instance I of the Maximum Level Motif problem, write $I^1 = I$ and $I^k = I \times I^{k-1}$ for all $k \geq 2$. According to Lemma 5, it follows by induction that I^k, $k \geq 1$, is an instance of the Maximum Level Motif problem.

Lemma 6. *Let $I = (T, \mathcal{M})$ be an instance of the* Maximum Level Motif *problem and let T_S be a solution for I. Then there exists a solution T_{S^k} for instance I^k such that $|T_{S^k}| \geq |T_S|^k$, for all $k \geq 1$.*

Proof. We prove the lemma by induction on k. The result is certainly valid for $k = 1$. Let $k \geq 2$ and assume that the lemma holds for all $1 \leq k' \leq k-1$. Let $T_S = (V_{T_S}, E_{T_S})$ be a solution for the Maximum Level Motif problem for instance I, with $V_{T_S} = \{v_1, v_2, \ldots, v_z\}$. Observe that T_S is a subtree of T and that all vertices in V_{T_S} have distinct colors since $\mathcal{M}$ is colorful. By Lemma 4, we can assume that the root r of T is part of V_{T_S}. We now construct a solution T_{S^k} for instance I^k as follows.

First, consider the subtree of T^k which consists of the set $V_{T_S,r'}$ of vertices $v_1(r'), v_2(r'), \ldots, v_z(r')$, where each $v_i(r')$ is the root of a subtree of T^k isomorphic to T^{k-1}. Observe that, by construction, the set of vertices $V_{T_S,r'}$ induces a subtree $T^k[V_{T_S,r'}]$ of T^k. Since vertices $v_1, v_2, \ldots, v_z$ all have distinct colors in T, then it follows that $v_1(r'), v_2(r'), \ldots, v_z(r')$ have distinct colors as well. Let $v_i[T^{k-1}]$ and $v_j[T^{k-1}]$, $1 \leq i < j \leq z$, be two subtrees of T isomorphic to

T^{k-1} rooted at $v_i(r')$ and $v_j(r')$, respectively. Observe that any two vertices $x \in v_i[T^{k-1}]$ and $y \in v_j[T^{k-1}]$ cannot have the same color, since $c(v_i) \neq c(v_j)$. Now, consider a subtree rooted at $v_i(r')$, with $1 \leq i \leq z$. By the induction hypothesis, there is a solution $T_{S^{k-1}}$ of the Maximum Level Motif problem over instance $I^{k-1} = (T^{k-1}, \mathcal{M}^{k-1})$, such that $|T_{S^{k-1}}| \geq |T_S|^{k-1}$. Notice that, by Lemma 4, we can assume that $T_{S^{k-1}}$ contains the root of T^{k-1}. Now we build solution T_{S^k}, by adding, for each $v_i(r')$, $1 \leq i \leq z$, a subtree of $v_i[T^{k-1}]$ isomorphic to $T_{S^{k-1}}$. Since T_S^k consists of $|T_S|$ such subtrees, it follows immediately that the inequality holds.

Finally, notice that the solution we have built is a feasible solution for the Maximum Level Motif problem for instance I^k. First, T_S^k is connected by construction. Furthermore, each vertex of T_S^k has a distinct color. Indeed, we have shown that this holds for any two vertices that are not in the same subtree $v_i[T^{k-1}]$. By the induction hypothesis, since $T_{S^{k-1}}$ is a feasible solution for the Maximum Level Motif problem over instance I^{k-1}, it follows that two vertices that belong to the same subtree $v_i[T^{k-1}]$ must have distinct colors. □

Lemma 7. *Let T_{S^k} be a solution for the* Maximum Level Motif *problem for instance $I^k = (T^k, \mathcal{M}^k)$. Then, one can compute in polynomial-time a solution T_S for instance I such that $|T_S|^k \geq |T_{S^k}|$.*

Proof. We prove the lemma by induction on k. The result is certainly valid for $k = 1$. Let $k \geq 2$ and assume that the lemma holds for each $1 \leq k' \leq k-1$. Let $T_{S^k} = (V_{S^k}, E_{S^k})$ be a solution for the Maximum Level Motif problem over instance I^k. According to Lemma 4, there is no loss of generality in assuming that the root of T^k is part of V_{S^k}. It follows that V_{S^k} contains vertices $x_1, \ldots x_p$ of T^k, with $p \leq |T|$, such that at least one vertex in subtree $x_i[T^{k-1}]$ isomorphic to T^{k-1} belongs to T_{S^k}. For each x_i, $1 \leq i \leq p$, denote by $x_i[T_S^{k-1}]$ the subtree of $x_i[T^{k-1}]$ which is part of T_{S^k}. Let $x_{\max}[T_S^{k-1}]$ be a subtree of maximum size among the subtrees $x_i[T_S^{k-1}]$, $1 \leq i \leq p$. Let T_S^{k-1} be a subtree of T^{k-1} isomorphic to $x_{\max}[T_S^{k-1}]$. Notice that T_S^{k-1} is a solution of the Maximum Level Motif problem over instance I^{k-1}. By the induction hypothesis, we can compute in polynomial time a solution $T_{S'}$ over the instance I such that $|T_{S'}|^{k-1} \geq |x_{\max}[T_S^{k-1}]|$. Denote now by T_p the subtree of T_{S^k} induced by $\{x_1 \ldots x_p\}$. Now $|T_{S^k}| \leq |T_p||x_{\max}[T_S^{k-1}]| \leq |T_p||T_{S'}|^{k-1}$. If $|T_{S'}| \geq |T_p|$, then $T_S = T'_S$ and the lemma holds, since $|T_{S'}||T_{S'}|^{k-1} \geq |T_p||T_{S'}|^{k-1}| \geq |T_{S^k}|$. Otherwise, if $|T_{S'}| < |T_p|$, let T_S be the subtree of T isomorphic to T_p. It follows that $|T_p||T_p|^{k-1} > |T_p||T_{S'}|^{k-1} \geq |T_{S^k}|$.

Observe that T_S is a feasible solution of the Maximum Level Motif problem over instance I. In the former case, when T_S is equal to $T_{S'}$, T_S is feasible by the induction hypothesis. Consider the latter case, when T_S is equal to T_p. Let $x_1, \ldots x_p$ be the vertices of T_p. Vertex x_i of T_p, $1 \leq i \leq p$, is associated to color $c_i(c(r), c(r), \ldots, c(r))$, where $c(r)$ is the color associated to the root of T and $c_i \in \mathcal{M}$. Observe that, since $\mathcal{M}^k$ is colorful, $c_i \neq c_j$, when $i \neq j$, hence the vertices of T_S all have distinct colors. □

We are now in position to state the main results of this section.

Theorem 1. *For any constant $\delta < 1$, the* Maximum Level Motif *problem cannot be approximated within ratio $2^{\log^{\delta} n}$ in polynomial-time unless* **NP** $\subseteq$ **DTIME**$[2^{\text{poly}\log n}]$.

Proof. Assume that there exists a constant $\delta < 1$ such that the Maximum Level Motif problem can be approximated within ratio $2^{\log^{\delta} n}$ in $O(n^c)$ time, for some constant c. For any fixed $\varepsilon > 0$, let $k = \lceil (\frac{\log^{\delta} n}{\log(1+\varepsilon)})^{\frac{1}{(1-\delta)}} \rceil$. Given an instance I of the Maximum Level Motif problem of size n, let I^k be the instance obtained by applying the product k times. Now, since the problem can be approximated within ratio $2^{\log^{\delta} n}$ in $O(n^c)$ time, it follows that there is an algorithm for the Maximum Level Motif problem for instance I^k with performance ratio $2^{\log^{\delta} n^k}$ that runs in $O(n^{ck}) = O(2^{\text{poly}\log n})$ time. But, according to Lemmas 6 and 7, there is an algorithm for instance I with performance ratio $(2^{\log^{\delta} n^k})^{1/k} \leq (1+\varepsilon)$, and hence we have designed a PTAS algorithm for the Maximum Level Motif problem. The result now follows from Proposition 9. □

Notice that the Maximum Level Motif problem is a special case of the Maximum Motif problem, and hence Theorem 1 holds for the Maximum Motif problem.

Substituting the complexity hypothesis **NP** $\subseteq$ **DTIME**$[2^{\text{poly}\log n}]$ by the classical **P** = **NP** yields the following result.

Proposition 10. *No polynomial-time algorithm achieves a constant factor approximation algorithm for the* Maximum Level Motif *problem unless* **P** = **NP**.

Proof. Let A be an algorithm for the Maximum Level Motif problem that achieves approximation factor α for some constant $\alpha > 1$. For some constant ε, let $k = \lceil \log_{1+\varepsilon} \alpha \rceil$. Let I be an instance of the Maximum Level Motif problem of size n. Then, I^k has size n^k. Let $\mathbf{opt}_k$ and $\mathbf{opt}$ denote the optimum solution for instances I^k and I, respectively. According to Lemma 6, $|\mathbf{opt}_k| \geq |\mathbf{opt}|^k$. Now, using Algorithm A, one can compute an α-approximate solution S_k for instance I^k, *i.e.*, $|S_k| \geq \mathbf{opt}_k/\alpha$. Now, thanks to Lemma 7, starting from S_k we can compute a solution for instance I of size at least $\left(\frac{OPT_k}{\alpha}\right)^{1/k}$. Applying Lemma 6 thus yields

$$\left(\frac{OPT_k}{\alpha}\right)^{1/k} \geq \frac{OPT}{\alpha^{1/k}}.$$

But $k = \lceil \log_{1+\varepsilon} \alpha \rceil$, and hence $(1+\varepsilon)^k \geq \alpha$. Therefore, $\alpha^{1/k} \leq 1+\varepsilon$. We have thus described a polynomial-time approximation scheme (PTAS) for the Maximum Level Motif problem since the running time of the algorithm is $O(n^k)$ for some constant k (this follows from Lemma 7 and the fact that computing the product instance is $O(n^k)$ time). The result now follows from Proposition 9. □

The same result holds also for the Maximum Motif problem.

4.2. Exponential-time Algorithms for Trees

We give here two exact branch-and-bound algorithms for the MAXIMUM MOTIF problem in case the target graph is a tree. The two results are summarized in the following proposition.

Proposition 11. *The* MAXIMUM MOTIF *problem for trees of size n can be solved in $O^*(1.62^n)$ time. In case the motif is colorful, the time complexity reduces to $O^*(1.33^n)$.*

The remaining part of Section 4.2, is devoted to proving Proposition 11. Let $I = (T, \mathcal{M})$ be an instance of the MAXIMUM MOTIF problem, where $T = (V, E)$ is a tree. Given a tree T', denote by $L(T')$ the set of leaves of T'. First, the algorithm chooses a vertex $r \in V$, and the tree T is rooted at r. Notice that the algorithm is iterated for each possible choice of r.

Lemma 8. *Let $T' = (V', E')$ be a subtree of T, let $V_S \subseteq (V' - L(T'))$, with $\mathbf{C}(V_S) \subseteq \mathcal{M}$, and let T_S be the subtree of T' induced by V_S. Then, we can compute in polynomial-time the maximum cardinality submotif $\mathcal{M}' \subseteq \mathcal{M}$, sych that that there is a set of leaves $L' \subseteq L(T')$, with $\mathbf{C}(L') = \mathcal{M}'$, and $\mathbf{C}(V_S \cup L') \subseteq \mathcal{M}$.*

PROOF. Denote by $m_{T_S}(c_i)$ (resp. $m_{L(T')}(c_i)$) the number of occurrences in $\mathbf{C}(T_S)$ (resp. $\mathbf{C}(L(T'))$) of color c_i, with $c_i \in \mathbf{C}$. For each $c_i \in \mathbf{C}$, denote by $m_{\mathcal{M}}(c_i)$ the occurrences of color c_i in $\mathcal{M}$. Observe that $m_{T_S}(c_i) \leq m_{\mathcal{M}}(c_i)$. Let $l_{T_S}(c_i) = m_{\mathcal{M}}(c_i) - m_{T_S}(c_i)$. Then $\mathcal{M}'$ can be computed by taking independently for each color $c_i \in \mathbf{C}$, $\min(l_{T_S}(c_i), m_{L(T')}(c_i))$ occurrences of color c_i. □

Consider $T' = (V', E')$ and $v_x \in V' - L(T')$. Let $T_S = (V_S, E_S)$ with $V_S \subseteq V'$ and $\mathbf{C}(V_S) \subseteq \mathcal{M}$. Let $T'(v_x)$ be the subtree of T' rooted at v_x. Then, if $v_x \in (V' - V_S)$, each vertex of the subtree $T'(v_x)$ is not in V_S.

Let $T_S = (V_S, E_S)$ be a feasible solution of the MAXIMUM MOTIF problem over the instance $I' = (T', \mathcal{M})$, where $T' = (V', E')$ is a subtree of T (T_S is a subtree of T'). An internal vertex $v_x \in V'$ not included in V_S, and adjacent to a vertex in V_S, is called a *candidate* vertex for T_S. A feasible solution T'_S for the MAXIMUM MOTIF problem is said to *extend* T_S, if it can be computed starting from T_S. The algorithm considers a candidate vertex v_x of V'. The algorithm branches in two cases associated to vertex v_x:

1. v_x is added to the solution T_S;
2. v_x is not added to the solution S, and the subtree $T'(x)$ is removed from T'.

Notice that, since v_x is an internal vertex of T', the subtree $T'(x)$ has size at least 2. Hence the number of vertices of V' that the algorithm has to consider is decreased by 1 in Case 1. above and by at least 2 in Case 2. above. Let I be an instance of the MAXIMUM MOTIF problem, consisting of a motif $\mathcal{M}$ of size m and a tree T with n vertices. Observe that $m \leq n$. Denote by $Z(n)$ the worst

case time complexity of the algorithm. Then $Z(n) = Z(n-1)+Z(n-2)+O(n)$. It follows that $Z(n) = O^*(1.62^n)$.

Consider now the case when the motif $\mathcal{M}$ is a set of colors. Consider a candidate vertex $v_x \in V'$ for T_S, colored $c(v_x)$. Assume that v_x is the only vertex of $V' - L(T')$ colored $c(v_x)$. Then, v_x is added to V_S. Indeed, since v_x is candidate and $\mathcal{M}$ is a set, there is no vertex in T_S that has color $c(x)$ and a vertex y of T' colored $c(x)$ must be a leaf.

The algorithm considers the following cases associated to a candidate vertex v_x for T_S:

1. v_x is added to the solution T_S; then, for each $v_y \in V'$ colored $c(v_x)$, the subtree rooted at vertex v_y is removed.
2. v_x is not added to the solution T_S; then the subtree of T' rooted at v_x is removed from T'.

Observe that only Case 1. above is modified, as Case 2. above is identical to the case when $\mathcal{M}$ is a multiset. In Case 1., v_x is added to T_S and, since there exists at least one internal vertex of V' colored $c(v_x)$, the number of vertices that the algorithm has still to consider is decreased by 3. Then, $Z(n) = Z(n-2) + Z(n-3) + O(n)$. It follows that $Z(n) = O^*(1.33^n)$.

Let $T_S = (V_S, E_S)$ be a solution constructed by the algorithm. Then, Lemma 8 is applied in order to add the maximum number of vertices x of $V - V_S$, such that x is adjacent to some vertex of V_S.

4.3. Parameterized Complexity

In this section, we present two fixed-parameter algorithms for the MAXIMUM MOTIF problem. The algorithms are based on the color-coding technique. Combining two families of perfect hash functions, we assign label to both the vertices of the graph and the colors that will be part of a solution of the MAXIMUM MOTIF, and compute a solution by dynamic programming. We first describe the perfect family of hash functions used in both algorithms. Next, we give an FPT algorithm in case the target graph is a tree and finally present a (slower) algorithm for the general case.

Consider an instance $I = (G, \mathcal{M})$ of the MAXIMUM MOTIF problem, where $G = (V, E)$ is a graph and $\mathcal{M}$ is a multiset of colors. For a color c_i of $\mathcal{M}$ and a subset $V' \subseteq V$, we denote by $m_{\mathcal{M}}(c_i)$ the number of occurrences of c_i in $\mathcal{M}$ and by $m_{V'}(c_i)$ the number of vertices in V' colored c_i. In the sequel, we assume that $m_{\mathcal{M}}(c_i) \leq m_V(c_i)$ since an occurrence of $\mathcal{M}$ in G has at most $\min\{m_{\mathcal{M}}(c_i), m_V(c_i)\}$ occurrences of color c_i. For a subset of vertices $V' \subseteq V$ and a submotif $\mathcal{M}' \subseteq \mathcal{M}$, we say that V' *violates* $\mathcal{M}'$ if $m_{\mathcal{M}'}(c_i) < m_{V'}(c_i)$ for some $c_i \in \mathcal{M}$.

Both algorithms are based on the color-coding technique [1]. We recall here the basic definition of perfect hash functions: for a set S, a family $\mathcal{F}$ of functions from S to $\{1, 2, \ldots, k\}$ is *perfect* if for any $S' \subseteq S$ of size k, there exists an injective function $f \in \mathcal{F}$ from S' to $\{1, 2, \ldots, k\}$. In the sequel, k denotes the size of a solution for the MAXIMUM MOTIF problem. Consider a family H of perfect hash functions from $\mathcal{M}$ to the set $\{1_H, 2_H, \ldots, k_H\}$ (we use the subscript

H to emphasis that this set is related to the family H). Let $\mathcal{M}'$ be a submotif of size k and let $G' = (V', E')$ be the occurrence of $\mathcal{M}'$ in G. Since H is perfect, there exists an injective function $h \in H$ that assigns to each occurrence of a color in $\mathcal{M}'$ a distinct label in $\{1_H, 2_H, \ldots, k_H\}$.

Fix some function $h \in H$. For any $c_i \in \mathcal{M}$, denote by $S_H(c_i) \subseteq \{1_H, 2_H, \ldots, k_H\}$ the set of labels associated to occurrences of color c_i by function h. Furthermore, we associate to each vertex v colored c_i the set of labels $S_H(v) = S_H(c_i)$. Let $V' \subseteq V$, $L_H \subseteq \{1_H, \ldots, k_H\}$, then $\mathbf{C}(S_H, V', L_H)$ is defined as the family of sets $S_H(v) \cap L_H$, with $v \in V'$. Notice that $\mathbf{C}(S_H, V', L_H)$ may contain more occurrences of the same set of labels. For example, if $v_1, v_2 \in V'$ and $c(v_1) = c(v_2)$, then $(S_H(v_1) \cap L_H) = (S_H(v_2) \cap L_H)$. In case $L_H = \{1_H, \ldots, k_H\}$, we abbreviate $\mathbf{C}(S_H, V', L_H)$ by $\mathbf{C}(S_H, V')$.

Definition 2. Let $\mathbf{C}(S_H, V', L_H)$ be a family of sets $S_H(v)$ with $v \in V'$ and $L_H \subseteq \{1_H, \ldots, k_H\}$, then $\mathbf{C}(S_H, V', L_H)$ is *feasible* if and only if there exists an injective function p from the sets of $\mathbf{C}(S_H, V', L_H)$ to L_H, such that, for each $S_H(v) \in \mathbf{C}(S_H, V', L_H)$, $p(S_H(v))$ is a label of $S_H(v) \cap L_H$.

Consider now a family $\mathbf{C}(S_H, V')$ of sets associated to V'. Let c_i be a color of $\mathcal{M}$, then by construction $|S_H(c_i)| \leq m_{\mathcal{M}}(c_i)$. Hence, if $\mathbf{C}(S_H, V')$ is feasible, then V' does not violate $\mathcal{M}$.

We now present an FPT algorithm for the case the target graph is a tree $T = (V, E)$. Let $r \in V$, and we want to compute a solution $T' = (V', E')$ of the Maximum Motif problem, such that $|V'| = k$ and $r \in V'$ (we run the algorithm for each $r \in V$.) Define r as the root of T and, for each internal vertex v of V, define a left-to-right ordering on the children of v. Assume that r is colored $c(r)$. Observe that, since r must belong to T', we can safely remove an occurrence of color $c(r)$ from $\mathcal{M}$. Furthermore, we assume that function h assigns to this occurrence of $c(r)$ label 1_H and that $S_H(r) = \{1_H\}$. Observe that there is no other vertex $u \in V - \{r\}$, such that $S_H(u)$ contains 1_H. We can now give the definition of the rightmost vertex of a subtree T' of T.

Definition 3. Let $T' = (V', E')$ be a subtree of T. A vertex $v \in V'$ is defined to be the rightmost vertex of T' if and only if (i) v has no children in V' and (ii) for each vertex $u \in V'$ on the path from r to v, V' does not contain the right sibling of u.

Now, consider a vertex $v \in V$ and a subset L_H of labels in $\{1_H, \ldots, k_H\}$. Define $P_r[v, L_H]$ as follows:

$$P[v, L_H] = \begin{cases} 1 & \text{if there exists a subtree } T' = (V', E') \text{ of } T \text{ with } r \in V' \text{ and with} \\ & \text{rightmost vertex } v \text{ and such that } \mathbf{C}(S_H, V', L_H) \text{ is feasible,} \\ 0 & \text{otherwise.} \end{cases}$$

The recurrence to compute $P[v, L_H]$ is as follows.

$$P[v, L_H] = \bigvee_{u, L'_H} P[u, L'_H], \tag{2}$$

where u is either a descendant of a left sibling of v or the parent of v, and $L'_H = L_H - \{i_H\}$, for some $i_H \in S_h(v) \cap L_H$. Notice that $P[v, \{1_H\}] = 0$, for each $v \in V - \{r\}$, $P[r, \{1_H\}] = 1$, and that $P[r, \{i_H\}] = 0$ for each $i_H \in \{2_H, \ldots, k_H\}$.

Lemma 9. *Given a labeling h of the motif $\mathcal{M}$, we can compute in $O(n^2 2^k)$ time if there is a subtree T' of T of size k that matches a submotif $\mathcal{M}'$ of $\mathcal{M}$.*

Proof. We have to show that $P[v, L_H] = 1$ if and only if there exists a subtree $T' = (V', E')$ of T having root r, which is an occurrence of a submotif $\mathcal{M}'$ of $\mathcal{M}$ of size $|L_H|$. Since T' must contain r, we assume that L_H contains 1_H.

First, consider a subtree $T' = (V', E')$ with root r. Let v be the rightmost vertex of T'. From the definition of the rightmost vertex, it follows that there is no child of v in V' and that there is no vertex in T' which is a right sibling of a vertex on the path from r to v. Denote by $T'' = (V'', E'')$ the tree obtained from T' by removing v. Let u be the rightmost vertex of T''. By definition of the rightmost vertex, u is either the parent of vertex v, denoted by $p(v)$, or a descendant of a child v' of $p(v)$ in T'' (with v' a left sibling of v in T' by definition of rightmost vertex).

Let $\mathcal{M}' = \mathbf{C}(V')$ be the multiset of colors associated to the vertices of T'. Consider $\mathbf{C}(S_H, V', L_H)$, the collection of sets of labels in L_H assigned to V'. Notice that $\mathbf{C}(S_H, V', L_H)$ is feasible, as T' is a solution of the Maximum Motif problem. It follows that there is an injective function p that assigns to each set $S_H(u)$, with $u \in V'$, a label i_H in $S_H(u)$. But then, function p assigns label $i_H \in S_H(v)$ to the set $S_H(v)$. It follows that the family of sets $S_H(u)$ with $u \in (V' - \{v\})$ must be feasible when p assigns a label in the set $\{1_H, \ldots, k_H\} - \{i_H\}$ to each set $S_H(u)$, with $u \in (V(T') - \{v\})$. Hence $P[v, L_H] = 1$.

Assume now that $P[v, L_H] = 1$. We will prove the results by induction. Since $P[v, L_H] = 1$, by Recurrence (2) it follows that there must exist a vertex $u \in V'$ and a label $i_H \in S_H(v)$, such that $P[u, L_H - \{i_H\}] = 1$. By the induction hypothesis, it follows that there is a subtree of $T'' = (V'', E'')$ of T having root r, so that T'' has size $|L_H| - 1$, u is the rightmost vertex of T'' and $\mathbf{C}(S_H, V'', L_H - \{i_H\})$ is feasible. Hence, by construction, also $\mathbf{C}(S_H, V', L_H)$ is feasible. We will show that v is adjacent to a vertex of T''. By definition of rightmost vertex, u is either the parent of vertex v, denoted by $p(v)$, or a descendant of a child v' of $p(v)$ in T'' (with v' a left sibling of v in T'). In the former case clearly u and v are adjacent. In the latter case, that is u is not $p(v)$, since T'' must be rooted at r, $p(v)$ belongs to T'', hence v is adjacent to a vertex of T''.

Observe that, if $P[v, \{1_H, \ldots, k_H\}] = 1$, it follows that there is a subtree $T' = (V', E')$ containing the root of T, so that each $\mathbf{C}(V')$ is assigned a distinct label in $\{1_H, \ldots, k_H\}$. By construction, V' does not violate $\mathcal{M}$, hence $\mathbf{C}(V')$ is a submotif of $\mathcal{M}$ of size k.

Now, we consider the time complexity of the algorithm. Observe that there exist $O(n2^k)$ values of the form $P[v, K']$, with $v \in V(T)$ and $L'_H \subseteq \{1_H, \ldots, k_H\}$. Now, in order to compute value $P[v, K']$, we have to check at most $O(nk)$ other values $P[u, K'']$. Hence the time complexity is $O(n^2 k 2^k)$. □

Observe that we have to choose $O(n)$ possible roots. Furthermore, since the family of perfect hash functions has size $O(\log n)\, 2^{O(k)}$, it follows that the algorithm is $O(k2^k n^3 \log n)\, 2^{O(k)}$ time.

Next, we describe a parameterized algorithm when the instance of the Maximum Motif problem consists of a graph $G = (V, E)$ and a motif $\mathcal{M}$. The algorithm for this case consists in combining two perfect families of hash functions, and then applying a strategy similar to that presented in [8, 12].

Consider two different perfect families of hash functions: a family H from $\mathcal{M}$ to $\{1_H, \ldots, k_H\}$, as we have previously introduced in this section, and a family F from the set V to $\{1_F, \ldots, k_F\}$. By the property of the family of perfect hash functions, we know that there is a function $f \in F$ such that the vertices of G that belong to a solution of size k are associated to distinct labels of $\{1_F, \ldots, k_F\}$. Similarly, we know that there is a function $h \in H$ such that the occurrences of colors of $\mathcal{M}$ that belong to an optimal solution, are associated to different labels of $\{1_H, \ldots, k_H\}$. Observe that each family of perfect hash functions consists of $O(\log n)\, 2^{O(k)}$ functions. Hence, we can combine all the possible pairs (f, h) of functions, with $f \in F$ and $h \in H$, in $O(\log^2 n)\, 4^{O(k)}$ time.

Recall that, for each color $c_i \in \mathcal{M}$, $S_H(c_i)$ denotes the set of labels associated to occurrences of color c_i by function h, and that, given that v is colored c_i, $S_H(v) = S(c_i)$. Now, for each $v \in V$ and for each subset $L \subseteq \{1_F, \ldots, k_F\}$, define $M_L(v)$ as the family of all sets of labels $H' \subseteq \{1_H, \ldots, k_H\}$ such that there exists an occurrence V', with $v \in V'$, where the set of labels in $\{1_F, \ldots, k_F\}$ that f assigns to V' is exactly L and such that $\mathbf{C}(S_H, V', H')$ is feasible. Now, we present a method called the *Batch procedure* for computing $M_L(v)$, similar to that introduced in [8, 12]. Assume that we have computed the family of sets $M_{L'}(v)$, with $L' \subseteq L \setminus f(v)$, we apply the following procedure.

Batch Procedure(L, v):

- Define C_H to be the family of all pairs (H', L') such that $H' \subseteq \{1_H, \ldots, k_H\} - \{i_H\}$ for some $i_H \in S_H(v_i)$, $L' \subseteq L \setminus \{f(v)\}$, and $H' \in M_{L'}(u)$ for some $u \in N(v)$.

- Run through all pairs of (H', L'), (H'', L'') in C_H and determine whether $H' \cap H'' = \emptyset$ and $H' \cup H'' \subseteq \{1_H, \ldots, k_H\} - \{i_H\}$, for some $i_H \in S_H(v_i)$, and whether $L' \cap L'' = \emptyset$. If there is such a pair, add $(H' \cup H'', L' \cup L'')$ to C_H and repeat this step. Otherwise, continue to the next step.

- Set $M_L(v)$ to be all the sets of labels $H' \cup \{i_H\}$, where $i_H \in S_H(v_i) - H'$, $(H', L') \in C_H$ and $L' = L \subseteq -\{f(v)\}$.

Lemma 10. *Given a vertex $v \in V$ and $L \subseteq \{1_F, \ldots, k_F\}$, the batch procedure computes correctly $M_L(v)$, assuming $M_{L'}(u)$ is given for each u adjacent to v and for each $L' \subseteq L \setminus \{f(v)\}$.*

PROOF. Consider the family C_H computed by the batch procedure. Let $(H', L') \in C_H$, where $H' \subseteq \{1_H, \ldots, k_H\} - \{i_H\}$ for some $i_H \in S_H(v)$ and $L' \subseteq L \setminus f(v)$. By construction, $H' = H'_1 \cup \cdots \cup H'_t$, where each $H'_i \subseteq \{1_H, \ldots, k_H\} - \{i_H\}$, $1 \leq i \leq t$, is associated by the function h to a submotif $\mathcal{M}'_i$ that has an occurrence in a set V'_i, such that V'_i includes a vertex adjacent to v. Notice that each V'_i is associated to a set of labels $L_i \subseteq \{1_F, \ldots, k_F\}$, such that $L_i \cap L_j = \emptyset$ for each V_j with $i \neq j$. Hence, all the connected components $G[V'_1]$, ..., $G[V'_t]$ are pairwise disjoint, and $\{v\} \cup V_1 \cdots \cup V_n$ is connected. It follows that $H' \cup \{i_h\}$ is then feasible and it is associated to vertices having labels L', so L' belongs to $M_L(v)$.

Consider now L', a set of labels in $M_L(v)$, such that $L' \cup f(v)$ is part of $M_L(v)$. Observe that, by definition, there exists a set of vertices V', associated to the set of labels L', such that the function f assigns to V' the set of labels in L'. Consider now the connected components induced by sets V_1, ..., V_t where $V_1 \cup V_2 \cdots \cup V_t = V'$. Since $V' \cup \{v\}$ must be a connected component, each V_i must have a vertex adjacent to v. Each connected component V_i is associated to a set of labels L_i, such that $L_i \cap L_j \neq \emptyset$, for each $j \neq i$. Now, the batch procedure will compute the pair (H', L') in its second step, and $H' \cup \{i_H\}$ will be added to C_H. □

Notice that the function h assigns a distinct label in $\{1_H, \ldots, k_H\}$ to each occurrence of a color in a submotif $\mathcal{M}'$, with $|\mathcal{M}'| = k$. Consider $M_L(v) = \{1_H, 2_H, \ldots, k_H\}$ with $L = \{1_F, 2_F, \ldots, k_F\}$. The set of vertices in V' associated to labels $\{1_F, 2_F \ldots, k_F\}$ is then associated to colors having labels in $\{1_H, 2_H \ldots, k_H\}$. Hence, $C(V')$ does not violate $\mathcal{M}$.

Lemma 11. *Given labeling functions $h : \mathcal{M} \to \{1, \ldots, k\}$ and $f : V \to \{1, \ldots, k\}$, the batch procedure determines in $O(2^{5k}kn^2)$ time whether there exists a solution of the* MAXIMUM MOTIF *problem of size k.*

PROOF. First, we will show that a set $M_L(v)$ is computed by the batch procedure in $O(2^{4k}kn)$ time. The first step of the batch procedure searches at most $2^k n$ families of subsets H' of labels in $\{1_H, \ldots, k_H\}$, for each $i_H \in S_H(v)$. Notice that $|S_H(v)| \leq k$. Each family consists of at most 2^k sets. Hence, the first step requires $O(2^{2k}kn)$.

For the second step of the batch procedure, observe that there are at most 2^{2k} sets of label-subset pairs H' and L', and hence the second step is repeated 2^{2k} times. Each iteration of this step can be computed in $O(2^k n)$ time, hence the second step requires $O(2^{4k}kn)$ time. Accounting also for the third step, the overall time complexity for one invocation of the batch procedure is $O(2^{4k}k + 2^{2k}kn) = O(2^{4k}kn)$.

According to Lemma 10, the batch procedure must be invoked at most $2^k n$ times in order to obtain $\mathcal{M}_L(v)$ for every $v \in V$ and every label subset $L' \subseteq \{1_F, \ldots, k_F\}$, hence the overall time complexity is $O(2^{5k}kn^2)$. □

Since each perfect family of hash functions has size $O(\log n)\, 2^{O(k)}$, the overall time complexity of the algorithm is $O(2^{5k} k n^2 \log^2 n)\, 4^{O(k)}$.

5. Conclusion

This paper was concerned with two optimization variants of the Graph Motif problem [24**?**]: Min-CC and Maximum Motif. We proved the problem of finding an occurrence of a colored motif that induces a minimum number of connected components (Min -CC) to be **APX**-hard for colorful motifs and paths, and we proved this problem to be fixed-parameter tractable when parameterized by the size of the motif but to be **W[2]**-hard when parameterized by the number of connected components. To complement the above results, we gave an exact exponential-time algorithm for the Min-CC problem in case the target graph is a tree. As for the Maximum Motif problem, *i.e.*, find a maximum cardinality submotif that occurs in the target graph, we proved this problem to be **APX**-hard in case the motif is colorful and the target graph is a bounded degree tree and to be not approximable within ratio $2^{\log^{\delta} n}$, for any constant $\delta < 1$, unless $\mathbf{NP} \subseteq \mathbf{DTIME}(2^{\mathrm{poly}\log n})$. We complemented these results by presenting two fixed-parameter algorithms and an exact exponential-time algorithm for the Maximum Motif problem.

We mention here some directions of interest for future works. First, approximation issues of the Min-CC problem are widely unexplored. In particular, is the Min-CC problem for paths approximable within a constant? How approximable is the Maximum Motif problem for colorful motifs in case the target graph is a tree in which each color occurs at most twice? Are Fourier methods [3] relevant for the parameterized Min-CC and Maximum Motif problems? Kernalization issues of both problems are also completely unexplored.

References

[1] N. Alon, R. Yuster, and U. Zwick, *Color coding*, Journal of the ACM **42** (1995), no. 4, 844–856.

[2] N. Betzler, M.R. Fellows, C. Komusiewicz, and R. Niedermeier, *Parameterized algorithms and hardness results for some graph motif problems*, Proc. 19th Annual Symposium on Combinatorial Pattern Matching (CPM), Pisa, Italy, Lecture Notes in Computer Science, vol. 5029, Springer, 2008, pp. 31–43.

[3] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, *Fourier meets möbius: fast subset convolution*, Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, ACM New York, NY, USA, 2007, pp. 67–74.

[4] P. Bonsma, *Complexity results for restricted instances of a paint shop problem*, Tech. Report 1681, Dep. of Applied Mathematics, Univ. of Twente, 2003.

[5] P. Bonsma, Th. Epping, and W. Hochstättler, *Complexity results on restricted instances of a paint shop problem for words*, Discrete Applied Mathematics **154** (2006), no. 9, 1335–1343.

[6] S. Bruckner, F. Hüffner, R.M. Karp, R. Shamir, and R. Sharan, *Topology-free querying of protein interaction networks*, Proc. 13th Annual International Conference on Computational Molecular Biology (RECOMB), Tucson, USA, Springer, 2009, p. 74.

[7] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein, *Introduction to algorithms*, third ed., MIT Press, Cambridge, 1992.

[8] R. Dondi, G. Fertin, and S. Vialette, *Weak pattern matching in colored graphs: Minimizing the number of connected components*, Proc.10th Italian Conference on Theoretical Computer Science (ICTCS), Roma, Italy, World-Scientific, 2007, pp. 27–38.

[9] ———, *Maximum motif problem in vertex-colored graphs*, Proc. 20th Annual Symposium on Combinatorial Pattern Matching (CPM'09), Lille, France (G. Kucherov and E. Ukkonen, eds.), Lecture Notes in Computer Science, vol. 5577, 2009, pp. 221–235.

[10] R. Downey and M. Fellows, *Parameterized complexity*, Springer-Verlag, 1999.

[11] W. Hochstättler T. Epping and P. Oertel, *Complexity results on a paint shop problem*, Discrete Applied Mathematics **136** (2004), no. 2-3, 217–226.

[12] M. Fellows, G. Fertin, D. Hermelin, and S. Vialette, *Sharp tractability borderlines for finding connected motifs in vertex-colored graphs*, Proc. 34th International Colloquium on Automata, Languages and Programming (ICALP), Wroclaw, Poland, Lecture Notes in Computer Science, vol. 4596, Springer, 2007, pp. 340–351.

[13] M.R. Fellows, M.T. Hallett, and U. Stege, *Analogs and duals of the MAST problem for sequences and trees*, Journal of Algorithms **49** (2003), no. 1, 192–216.

[14] J. Flum and M. Grohe, *Parameterized complexity theory*, Springer Verlag, 2006.

[15] A.J. Goldman, *Optimal center location in simple networks*, Transportation Science **5** (1971), no. 2, 212–221.

[16] S. Guillemot, *Parameterized complexity and approximability of the SLCS problem*, Proc. 3rd International Workshop on Parameterized and Exact Computation (IWPEC), Victoria, Canada, Lecture Notes in Computer Science, 2008, pp. 115–128.

[17] F. Harary, *Graph theory*, Addison-Wesley, 1969.

[18] J. Hein, T. Jiang, L. Wang, and K. Zhang, *On the complexity of comparing evolutionary trees*, Discrete Applied Mathematics **71** (1996), 153–169.

[19] F. Hüffner, S. Wernicke, and T. Zichner, *Algorithm engineering for color-coding with applications to signaling pathway detection*, Algorithmica **52** (2008), no. 2, 114–132.

[20] T. Jiang and M. Li, *On the approximation of shortest common supersequences and longest common subsequences*, SIAM Journal on Computing **24** (1995), 1122–1139.

[21] D. Karger, R. Motwani, and G.D.S. Ramkumar, *On approximating the longest path in a graph*, SIAM Journal on Computing **24** (1995), 1122–1139.

[22] B.P. Kelley, R. Sharan, R.M. Karp, T. Sittler, D. E. Root, B.R. Stockwell, and T. Ideker, *Conserved pathways within bacteria and yeast as revealed by global protein network alignment*, Proceedings of the National Academy of Sciences **100** (2003), no. 20, 11394–11399.

[23] M. Koyutürk, A. Grama, and W. Szpankowski, *Pairwise local alignment of protein interaction networks guided by models of evolution*, Proc. 9th Annual International Conference on Research in Computational Molecular Biology (RECOMB), Cambridge, MA, USA, Lecture Notes in Bioinformatics, vol. 3500, Springer, 2005, pp. 48–65.

[24] V. Lacroix, C.G. Fernandes, and M.-F. Sagot, *Motif search in graphs: application to metabolic networks*, IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB) **3** (2006), no. 4, 360–368.

[25] N. Megiddo, A. Tamir, E. Zemel, and R. Chandrasekaran, *An $o(n \log^2 n)$ algorithm for the k-th longest path in a tree with applications to location problems*, SIAM J. Comp. **10** (1981), no. 2, 328–337.

[26] R. Niedermeier, *Invitation to fixed parameter algorithms*, Lecture Series in Mathematics and Its Applications, Oxford University Press, 2006.

[27] C.H. Papadimitriou and M. Yannakakis, *Optimization, approximation and complexity classes*, Journal of Computer and System Sciences **43** (1991), 425–440.

[28] A. Paz and S. Moran, *Non deterministic polynomial optimization problems and their approximations*, Theoretical Computer Science **15** (1981), 251–277.

[29] R. Raz and S. Safra, *A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP*, Proc. 29th Annual ACM Symposium on Theory of Computing (STOC), El Paso, Texas, United States, 1997, pp. 475–484.

[30] J. Scott, T. Ideker, R.M. Karp, and R. Sharan, *Efficient algorithms for detecting signaling pathways in protein interaction networks*, Journal of Computational Biology **13** (2006), 133–144.

[31] R. Sharan, T. Ideker, B. Kelley, R. Shamir, and R.M. Karp, *Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data*, Proc. 8th annual international conference on Computational molecular biology (RECOMB), San Diego, California, USA, ACM Press, 2004, pp. 282–289.

[32] R. Sharan, S. Suthram, R.M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R.M. Karp, and T. Ideker, *Conserved patterns of protein interaction in multiple species*, Proc. Natl Acad. Sci. USA **102** (2005), no. 6, 1974–1979.