Graph Motif Problems Parameterized by Dual

Guillaume Fertin[1] and Christian Komusiewicz*[2]

1 **Laboratoire d'Informatique de Nantes-Atlantique, UMR CNRS 6241, Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3, France**
guillaume.fertin@univ-nantes.fr

2 **Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany**
christian.komusiewicz@uni-jena.de

Abstract

Let $G = (V, E)$ be a vertex-colored graph, where C is the set of colors used to color V. The Graph Motif (or GM) problem takes as input G, a multiset M of colors built from C, and asks whether there is a subset $S \subseteq V$ such that (i) $G[S]$ is connected and (ii) the multiset of colors obtained from S equals M. The Colorful Graph Motif (or CGM) problem is the special case of GM in which M is a set, and the List-Colored Graph Motif (or LGM) problem is the extension of GM in which each vertex v of V may choose its color from a list $\mathcal{L}(v)$ of colors.

We study the three problems GM, CGM, and LGM, parameterized by $\ell := |V| - |M|$. In particular, for general graphs, we show that, assuming the strong exponential time hypothesis, CGM has no $(2-\epsilon)^{\ell} \cdot |V|^{O(1)}$-time algorithm, which implies that a previous algorithm, running in $O(2^{\ell} \cdot |E|)$ time is optimal [2]. We also prove that LGM is W[1]-hard even if we restrict ourselves to lists of at most two colors. If we constrain the input graph to be a tree, then we show that GM can be solved in $O(4^{\ell} \cdot |V|)$ time but admits no polynomial-size problem kernel, while CGM can be solved in $O(\sqrt{2}^{\ell} + |V|)$ time and admits a polynomial-size problem kernel.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.1 Combinatorics, G.2.2 Graph Theory

Keywords and phrases NP-hard problem, subgraph problem, fixed-parameter algorithm, lower bounds, kernelization

Digital Object Identifier 10.4230/LIPIcs.CPM.2016.7

1 Introduction

The Subgraph Isomorphism problem is the following pattern matching problem in graphs: given a (typically large) host graph G and a (small) query graph H, return one (or all) occurrence(s) of H in G, where the term occurrence denotes here a subset S of $V(G)$ such that $G[S]$, the subgraph of G induced by S, is isomorphic to H. This type of graph mining problem has numerous applications, notably in biology [20]. Subgraph Isomorphism is a *structural* graph pattern matching problem, where one looks for similar graph structures between H and G. In some biological contexts, however, additional information is provided to the vertices of the graphs, for example their biological function. This can be modeled by labeling each vertex of the graph, for example by giving it one or several colors, each corresponding to an identified function. In the presence of such functional annotation, the structure of a given induced subgraph may be of less importance than the functions it

* Christian Komusiewicz was supported by the DFG, project "Multivariate algorithmics for graph and string problems in bioinformatics" (KO 3669/4-1).

corresponds to. Thus, a new set of *functional* graph pattern matching problems has emerged, starting with the Graph Motif problem [15], which was introduced in the context of the analysis of metabolic networks. Here, what is primarily sought in the host graph is a multiset M of colors that represents the functions of interest.

Graph Motif (GM)
Input: A multiset M built on a set C of colors, an undirected graph $G = (V, E)$, and a coloring $\chi : V \to C$.
Question: Is there a set $S \subseteq V$ such that $G[S]$ is connected and there is a one-to-one mapping f from S to M?

Many variants of the GM problem have been introduced and studied. In particular, List-Colored Graph Motif (or LGM) is a generalization of GM that is used to identify protein complexes in protein interaction networks that are similar to a given protein complex from a different species [7]. In LGM, a list-coloring $\mathcal{L} : V \to 2^C$ is assigned to each vertex of G, and the question asked is the existence of $S \subseteq V$ such that (i) $G[S]$ is connected and (ii) the one-to-one mapping f from S to M we look for satisfies $\forall v \in S : f(v) \in \mathcal{L}(v)$. The special case of GM in which M is a set is called Colorful Graph Motif (or CGM). Many optimization problems related to GM have received interest, including some that are related to tandem mass spectrometry and where the input graph is directed [19]. All these problem variants have given rise to a very abundant literature. CGM, GM, and LGM are NP-hard even in very restricted cases [10]. Consequently, many of the above-mentioned studies have focused on (dis)proving fixed-parameter tractability of the problems (see e.g. [21] for the most recent survey on the topic). In such cases, very often the parameter $k := |M| = |S|$ is considered.

In this paper, we study the parameterized complexity of GM, CGM, and LGM, but we differ from the usual viewpoint by focusing on the *dual* parameter $\ell := |V| - |S|$, that is, ℓ is the number of vertices to be deleted from G to obtain a solution. Although the choice of ℓ may be disputable because it may *a priori* be too large to expect a good behavior in practice, there are several arguments for choosing such a parameter: First, after some initial data reduction, the input may be divided into smaller connected components, where ℓ is not much larger than k. Second, the algorithms for parameter k rely on algebraic techniques or dynamic programming, and in both cases, the worst-case running time is equivalent to the actual running time. In contrast, for example for CGM, the algorithm for parameter ℓ is a search tree algorithm [2], and search tree algorithms can be accelerated substantially via pruning rules. Finally, there are subgraph mining problems where the dual parameter ℓ is usually bigger than the parameter k but leads to the current-best algorithm (in terms of performance on real-world instances) [13]. Hence, parameterization by ℓ may be useful even if ℓ is bigger than k, and thus deserves to be studied.

Related work and our contribution. GM is NP-hard, even when M is composed of two colors [10]. Concerning the parameterized complexity for parameter $k := |M|$, the current-best randomized algorithm has a running time of $2^k \cdot n^{O(1)}$ [3, 18] where $n := |V|$, and there is some evidence that this cannot be improved to a running time of $(2 - \epsilon)^k \cdot n^{O(1)}$ [3]. The current-best running time for a deterministic algorithm is $5.22^k \cdot n^{O(1)}$ [17]. GM on trees can be solved in $n^{O(c)}$ time where c is the number of colors in M [10], but it is W[1]-hard with respect to c [10]. Other parameters, essentially related to the structure of the input graph G, have been studied by Bonnet and Sikora [6]. Finally, concerning parameter ℓ, GM has been shown to be W[1]-hard, even when M is composed of two colors [2].

Table 1 Overview of new and previous results with respect to the dual parameter $\ell := n - k$, where $n := |V|$, $m := |E|$, $k := |M|$, and $\Delta := \max_{v \in V} |\mathcal{L}(v)|$ denotes the maximum list size in G. The lower bound result for CGM assumes the strong exponential time hypothesis (SETH) [14].

	General graphs	Trees
LGM	W[1]-hard [2]	?
LGM, $\Delta = 2$	W[1]-hard (Cor. 4)	?
GM	W[1]-hard [2]	$O(4^\ell \cdot n)$ (Thm. 5) no poly. kernel (Thm. 8)
CGM	$O(2^\ell \cdot m)$ [2], no $(2-\epsilon)^\ell \cdot n^{O(1)}$ (Thm. 1) no poly. kernel (Thm. 2)	$O(\sqrt{2}^\ell + n)$ (Thm. 13), $(2\ell+1)$-vertex kernel (Thm. 10)

Since CGM is a special case of GM, any above-mentioned positive result for GM also holds for CGM. Besides, CGM is NP-hard, even for trees of maximum degree 3 [10], and does not admit a polynomial-size problem kernel with respect to k even if G has diameter two or if G is a comb graph (a special type of tree with maximum degree 3) [1]. Finally, CGM can be solved in $O(2^\ell \cdot m)$ time [2]. The LGM problem is an extension of GM and thus any negative result for GM propagates to LGM. Moreover, LGM is known to be fixed-parameter tractable with respect to k, the current-best algorithm runs in $2^k \cdot n^{O(1)}$ time [18]. Concerning parameter ℓ, LGM has been shown to be W[1]-hard even when M is a set [2].

As mentioned above, we study GM, LGM and CGM with respect to the dual parameter $\ell := n - k$. Since many results in general graphs turn out to be negative, we also chose to focus on the special case where the input graph G is a tree. Our results are summarized in Table 1. In a nutshell, we strengthen previous hardness results for the general case and show that the $O(2^\ell \cdot m)$-time algorithm for CGM is essentially optimal. Then, we show that for GM on trees a fixed-parameter algorithm can be achieved, and that, for CGM on trees, a polynomial problem kernel and better running times than for general graphs can be achieved.

Preliminaries. Throughout the paper, the input graph for our three problems is $G = (V, E)$, and we let $n := |V|$ (resp. $m := |E|$) denote its number of vertices (resp. edges). We use $[n] := \{1, \ldots, n\}$ to denote the set of the integers from 1 through n. The set S of vertices sought for in the three problems is called an *occurrence* of M. If G is vertex-colored, we call a vertex set S *colorful* if $|S| = |M|$ and all vertices in S have pairwise different colors. A vertex v is called *unique* if it is assigned a color c that is assigned to no other vertex in V.

We briefly recall the relevant notions of parameterized algorithmics [8]. A *reduction to a problem kernel*, or *kernelization*, is an algorithm that takes as input an instance (I, k) of a parameterized problem and produces in polynomial time an equivalent instance (I', k') (that is, having the same solution) such that (i) $|I'| \leq g(k)$, and (ii) $k' \leq k$. The instance (I', k') is called *problem kernel* and g is called the *size of the problem kernel*. If g is a polynomial function, then the problem admits a *polynomial-size problem kernelization*. The class W[1] is a basic class of presumed fixed-parameter intractability [8], that is, if a problem is W[1]-hard for parameter k, then we assume that it cannot be solved in $f(k) \cdot n^{O(1)}$ time [8]. The strong exponential time hypothesis (SETH) assumes that CNF-SAT with n variables cannot be solved in time $(2-\epsilon)^n$ for any $\epsilon > 0$ [14].

This work is structured as follows. In Section 2, we present lower bounds for LGM and CGM on general graphs. These negative results motivate our study of the case when G

is a tree; our results for GM on trees and CGM on trees will be presented in Section 3 and Section 4, respectively. Due to lack of space, some proofs are deferred to a full version of the article.

2 Parameterization by Dual in General Graphs: Tight Lower Bounds

CGM can be solved in $O(2^\ell \cdot m)$ time [2]. We show here that this running time bound is essentially optimal.

▶ **Theorem 1.** Colorful Graph Motif *cannot be solved in* $(2-\epsilon)^\ell \cdot n^{O(1)}$ *time unless the strong exponential time hypothesis fails.*

Proof. We present a polynomial-time reduction from CNF-SAT:

> **Input:** A boolean formula Φ in conjunctive normal form with clauses $\mathcal{C}_1, \ldots, \mathcal{C}_q$ over variable set $X = \{x_1, \ldots, x_r\}$.
> **Question:** Is there an assignment β to X that satisfies Φ?

The reduction works as follows. First, for each variable $x_i \in X$ introduce two *variable vertices* v_i^t and v_i^f and color each of the two vertices with color χ_i^x. The idea is that (with the final occurrence) we must select exactly one vertex for this color. This selection will correspond to a truth assignment to X. Now, introduce for each clause $\mathcal{C}_i$ a *clause vertex* u_i, color u_i with a unique color $\chi_i^{\mathcal{C}}$ and make u_i adjacent to vertex v_j^t if x_j occurs nonnegated in $\mathcal{C}_i$ and to vertex v_j^f if x_j occurs negated in $\mathcal{C}_i$. Finally, introduce one further vertex v^* with a unique color χ^*, make v^* adjacent to all variable vertices and let M be the set containing each of the introduced colors exactly once. Note that there are exactly $|X|$ colors that appear twice in G and that all other colors appear exactly once. Hence, $\ell = |X|$. We next show the correctness of the reduction. Let I denote the constructed instance of CGM.

First, assume that Φ is satisfiable and let β be a satisfying assignment of X. For the CGM instance consider the vertex set $S \subseteq V$ that contains all clause vertices, vertex v^*, and for each variable x_i the vertex v_i^t if β sets x_i to 'true' and v_i^f otherwise. Clearly, $|S| = |M|$ and no two vertices of S have the same color. To show that I is a yes-instance of CGM it remains to show that $G[S]$ is connected. First, the subgraph induced by the variable vertices in S plus v^* is a star and thus it is connected. Second, since β is a satisfying assignment each clause vertex in S has at least one neighbor in S (which is by construction a variable vertex). Hence, $G[S]$ is connected.

Conversely, assume that I is a yes-instance of CGM, and let S be a colorful vertex set with $|S| = |M|$ such that $G[S]$ is connected. Since S is colorful, the variable vertices in S correspond to a truth assignment of X. This assignment satisfies X: Indeed, since $G[S]$ is connected, there is a path in $G[S]$ between each clause vertex u_i and v^*, and thus there is a neighbor of u_i that is in S. If this neighbor is v_j^t (resp. v_j^f), then by construction, β assigns 'true' (resp. 'false') to x_j and thus $\mathcal{C}_i$ is satisfied.

Thus, the two instances are equivalent. Now observe that since $\ell = |X| = r$ and $n = 2r+q+1$, any $(2-\epsilon)^\ell \cdot n^{O(1)}$-time algorithm implies a $(2-\epsilon)^r \cdot (r+q)^{O(1)}$-time algorithm for CNF-SAT. This directly contradicts the SETH. ◀

The above reduction also makes the existence of a polynomial-size problem kernel for parameter ℓ unlikely. This is implied by the following two facts. First, CNF-SAT parameterized by the number of variables does not admit a polynomial-size problem kernel unless NP $\subseteq$ coNP/poly [9]. Second, the reduction presented in proof of Theorem 1 is a *polynomial parameter transformation* [5] from CNF-SAT parameterized by the number of

variables to CGM parameterized by ℓ. More precisely, given an input CNF-SAT formula Φ on variable set X, the reduction produces an instance $I = (M, G, \chi)$ of CGM with $\ell = |X|$. Now, any polynomial-size problem kernelization applied to I produces in polynomial time an equivalent CGM instance I' of size $\ell^{O(1)} = |X|^{O(1)}$. Since CNF-SAT is NP-hard, we can now transform this CGM instance in polynomial time into an equivalent CNF-SAT instance that has size $\ell^{O(1)} = |X|^{O(1)}$. Hence, a polynomial-size problem kernel for CGM parameterized by ℓ implies a polynomial-size problem kernel for CNF-SAT parameterized by $|X|$. This implies NP $\subseteq$ coNP/poly [9] (which in turn implies a collapse of the polynomial hierarchy).

▶ **Theorem 2.** Colorful Graph Motif *parameterized by ℓ does not admit a polynomial-size problem kernel unless NP $\subseteq$ coNP/poly.*

We have thus resolved the parameterized complexity of CGM parameterized by ℓ on general graphs and now turn to the more general LGM which is W[1]-hard with respect to ℓ [2]. Here, it would be desirable to obtain fixed-parameter algorithms for the parameter ℓ at least for some restricted inputs. In other words, we would like to further exploit the structure of real-world instances to obtain tractability results. A very natural approach here is to consider the size and structure of the list-colorings $\mathcal{L}(v)$ as additional parameter. Unfortunately, the problem remains W[1]-hard even for the following very restricted case of list-colorings. Herein, the vertex-color graph is the graph with vertex set $V \cup C$ and edge set $\{\{v, c\} \mid c \in \mathcal{L}(v)\}$.

▶ **Theorem 3.** List-Colored Graph Motif *is W[1]-hard with respect to ℓ even if the vertex-color graph is a disjoint union of paths.*

We immediately obtain the following.

▶ **Corollary 4.** List-Colored Graph Motif *is W[1]-hard with respect to ℓ even if $|\mathcal{L}(v)| \leq 2$ for every vertex in G.*

3 Graph Motif on Trees

Motivated by these negative results on general graphs, we now study the special case where the input graph is a tree. For LGM, we were not able to resolve the parameterized complexity with respect to ℓ for this case. Hence, we focus on the more restricted GM problem. We show that GM is fixed-parameter tractable with respect to ℓ if the input graph is a tree. Recall that for general graphs, GM is W[1]-hard for ℓ even if the motif M contains only two colors [2]. Hence, the tree structure helps significantly when parameterizing by ℓ.

3.1 A Dynamic Programming Algorithm

Call a color of M *abundant* if it occurs more often in G than in M. The abundant colors are exactly the ones that have to be "deleted" to obtain a solution S. Let $c_1, \ldots, c_j$ denote the abundant colors of M, and let ℓ_i denote the difference between the number of vertices in V that have color c_i and the multiplicity of c_i in M. This implies in particular that $\sum_{1 \leq i \leq j} \ell_i = \ell$.

The algorithm is a dynamic programming algorithm that works on a rooted representation of G. Thus, obtain a rooted tree T by rooting G at an arbitrary vertex $r \in V$. As usual for dynamic programming on trees, the idea is to combine partial solutions of subtrees. Our algorithm is somewhat similar to a previous dynamic programming algorithm for GM on graphs of bounded treewidth [10] but the analysis and concrete table setup is different.

▶ **Theorem 5.** Graph Motif *can be solved in $O(4^\ell \cdot n)$ time if G is a tree.*

The fixed-parameter tractability of GM on trees also implies the following result for LGM.

▶ **Corollary 6.** LGM *can be solved in* $O(4^\ell \cdot n)$ *time if* G *is a tree and the vertex-color graph* $H = (V \cup C, \{\{v, c\} \mid c \in \mathcal{L}(v)\})$ *is a disjoint union of paths.*

Proof. We describe a reduction of this special case of LGM on trees to GM on trees. Here, we call the vertices of H that are from V the V-vertices of H and those that are from C the C-vertices. Observe that without loss of generality, we can assume that all colors in the lists are contained in M. First, if H has a connected component that contains more C-vertices than V-vertices, then the instance $(M, G, \mathcal{L})$ is a no-instance and can be immediately rejected. Second, for any connected component H' of H that contains at least two C-vertices c_1 and c_2 that have multiplicity two in M, then the instance is also a no-instance: In H', the number of V-vertices exceeds the number of C-vertices by at most one. Hence, if four or more V-vertices are assigned only to c_1 or c_2, then there is some other C-vertex in H' that is assigned to none of the V-vertices. A similar argument applies if H' contains a C-vertex that has multiplicity at least three in M.

If the instances are not rejected because any of the cases described above applies, then each connected component H' of H has at most one C-vertex that has multiplicity two in M and all other C-vertices have multiplicity at most one. We show that in both cases, the constraints of $\mathcal{L}$ for H' can be replaced by simple coloring constraints.

Case 1: Every C-vertex of H' has multiplicity one in M. If H' has the same number of V-vertices as C-vertices (equivalently, H' has an even number of vertices), then every occurrence S of M contains all V-vertices from H'. Otherwise, if H' has more V-vertices than C-vertices (equivalently, H' has an odd number of vertices), then every occurrence S of M contains all except one V-vertex from H'. In both cases, we can replace the constraints as follows. Introduce a color $c_{H'}$, color all V-vertices in H' with color $c_{H'}$ and replace in M every C-vertex of H' by $c_{H'}$. In the first case, the number of vertices with color $c_{H'}$ is exactly the multiplicity of $c_{H'}$ in M, in the second case it is the multiplicity of $c_{H'}$ in M plus one.

Case 2: One C-vertex c of H' has multiplicity two in M. If H' has the same number of V-vertices as C-vertices (equivalently, H' has an even number of vertices), then the instance is a no-instance and can be rejected immediately: any assignment of colors to the V-vertices either fails to assign one of the C-vertices or assigns at most one V-vertex to c. Otherwise, if H' has an odd number of vertices, every occurrence S of M contains all V-vertices of H'. The constraints posed by H' may thus be replaced as follows: Introduce a color $c_{H'}$, color all V-vertices in H' with color $c_{H'}$ and replace in M every C-vertex of H' by $c_{H'}$ (replace c twice). Then the multiplicity of $c_{H'}$ in M is exactly the number of V-vertices in H'.

Applying these replacements exhaustively then results in an equivalent instance of GM on trees which can be solved in the claimed running time due to Theorem 5. ◀

3.2 A Kernelization Lower Bound

We now show that GM does not admit a polynomial-size problem kernel with respect to ℓ even if G is a tree. The proof is based on a cross-composition [4] from the W[1]-hard Multicolored Clique problem [11].

Multicolored Clique
Input: A graph $H = (W, F)$ and a vertex-coloring $\chi : W \to \{1, \ldots, k\}$.
Question: Is there a vertex set $S \subseteq W$ such that S is colorful, that is, $|S| = k$ and the vertices in S have pairwise different colors, and $H[S]$ is a clique?

To avoid confusion between the colors of the Multicolored Clique instance and the GM instance, we refer to the colors of the Multicolored Clique instance as *labels* in the following. Informally, cross-compositions are reductions that combine many instances of one problem into one instance of another problem. The existence of a cross-composition from an NP-hard problem to a parameterized problem Q implies that Q does not admit a polynomial-size problem kernel (unless NP $\subseteq$ coNP/poly) [4].

▶ **Definition 7** ([4]). Let $L \subseteq \Sigma^*$ be a language, let R be a polynomial equivalence relation on Σ^*, and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. An *or-cross-composition of L into Q* (with respect to R) is an algorithm that, given t instances $x_1, x_2, \ldots, x_t \in \Sigma^*$ of L belonging to the same equivalence class of R, takes time polynomial in $\sum_{i=1}^{t} |x_i| + k$ and outputs an instance $(y, k) \in \Sigma^* \times \mathbb{N}$ of Q such that

- the parameter value k is polynomially bounded in $\max_{i=1}^{t} |x_i| + \log t$, and
- the instance (y, k) is a yes-instance for Q if and only if at least one instance x_i is a yes-instance for L .

We present an or-cross composition of Multicolored Clique into GM on trees parameterized by ℓ. The polynomial equivalence relation R will be simply to assume that all the Multicolored Clique instances have the same number of vertices n. The main trick is to encode vertex identities in the graph of the Multicolored Clique instance by numbers of colored vertices in the GM instance; note that this approach was also followed in previous works on GM [10, 6]. Given t instances $(H_1 = (W_1, F_1), \chi_1), H_2 = (W_2, F_2), \chi_2), \ldots, H_t = (W_t, F_t), \chi_t)$ of Multicolored Clique such that $|W_i| = n$ for all $i \in [t]$, we reduce to an instance of GM where the input graph is a tree as follows. Herein, we assume without loss of generality that $t = 2^s$ for some integer s.

The first construction step is to add one vertex r that connects the different parts of the instance and which will be contained in every occurrence of the motif. The vertex r thus receives a unique color that may not be deleted. To this vertex r we attach subtrees corresponding to edges of the input instances. Deleting vertices of such a subtree then corresponds to selecting the endpoints of the corresponding edge.

Instance selection gadget. The technical difficulty in the construction is to ensure that the solution deletes only vertices in subtrees corresponding to edges of the same graph. To achieve this, we introduce $k \cdot (k-1) \cdot \log t$ instance selection colors $\iota[p, q, \tau]$ where $p \in [k]$, $q \in [k] \setminus \{p\}$, and $\tau \in [\log t]$, and demand that the solution deletes exactly one vertex of each instance selection color. To ensure that exactly one instance is selected, we use two further colors ι^+ and ι^-. For each Multicolored Clique instance (H_i, χ_i), attach an *instance selection path* P_i to r that is constructed based on the number i. Let $b(i)$ denote the binary expansion of i and let $b_\tau(i)$, $\tau \in [\log t]$, denote the τth digit of $b(i)$. Construct a path P_i containing first a vertex with color ι^+, then in arbitrary order exactly one vertex of each color in the color set $I_i := \{\iota[p, q, \tau] : b_\tau(i) = 1\}$, and then a vertex with color ι^-. Attach the path P_i to r by making the vertex with color ι^+ a neighbor of r.

The idea of the construction is that exactly one instance selection path P_i is deleted completely and that this will force any solution to delete paths that "complement" P_i (that is, paths which contain all $\iota[p, q, \tau]$ such that $b_\tau(i) = 0$) in the rest of the graph.

Edge selection gadget. To force deletion of subtrees corresponding to exactly $\binom{k}{2}$ edges with different labels, we introduce $2k(k-1)$ label selection colors $\lambda[p, q]^+$ and $\lambda[p, q]^-$ where $p \in [k]$ and $q \in [k] \setminus \{p\}$. These colors will ensure that for each pair of labels p and q the solution deletes exactly one path corresponding to the ordered pair (p, q) and one path corresponding to the pair (q, p).

There are two further sets of colors. One set is used for ensuring vertex consistency of the chosen edges, that is, to make sure that all the selected edges with label pair $(p, \cdot)$ correspond to the same vertex with label p. More precisely, we introduce a color $\omega[p, q]$ for each $p \in [k]$ and each $q \in [k] \setminus \{p\}$, except for the biggest $q \in [k] \setminus \{p\}$.

The final color set is used to check that the edges selected for label pair (p, q) and for label pair (q, p) are the same. To this end, we introduce a set of colors $\varepsilon[p, q]$ for each $p \in [k]$ and each $q \in [k] \setminus \{p\}$ such that $q > p$. To perform the checks of vertex and edge consistency, we encode the identities of vertices and edges into path lengths. More precisely, we assign each vertex $v \in W_i$ a unique (with respect to the vertices of W_i) number $\#(v) \in [n]$.

Now, for each label pair (p, q) and each instance i, attach one path $P_i(u, v)$ to r for each edge $\{u, v\}$ where u has color p and v has color $q \neq p$. The path $P_i(u, v)$

- starts with a vertex with color $\lambda[p, q]^+$ that is made adjacent to r,
- then contains exactly one vertex of each color in $\{\iota[p, q, \tau] : \iota[p, q, \tau] \notin I_i\}$,
- then contains $\#(u)$ vertices of color $\varepsilon[p, q]$ if $p < q$ and $n - \#(v)$ vertices of color $\varepsilon[q, p]$ if $p > q$,
- then, if q is not the biggest label in $[k] \setminus p$, contains $\#(u)$ vertices with color $\omega[p, q]$,
- then, if q is not the smallest label in $[k] \setminus p$, contains $n - \#(u)$ vertices with color $\omega[p, q']$, where q' is the next-smaller label in $[k] \setminus p$ (if $p = q-1$, then $q' = q-2$; otherwise $q' = q-1$), and
- ends with a vertex with color $\lambda[p, q]^-$.

Let $\mathcal{C}$ denote the multiset containing all the vertex colors of all vertices added during the construction with their respective multiplicities. In the correctness proof it will be easier to argue about the colors that are not contained in M. Hence, the construction is completed by setting the multiset D of colors to "delete" to contain each color with multiplicity one except

- the color of r which is not contained in D,
- the vertex consistency colors $\omega[p, q]$ each of which is contained with multiplicity n, and
- the edge selection colors $\varepsilon[p, q]$ each of which is contained with multiplicity n.

The motif M is defined as $M := \mathcal{C} \setminus D$. It remains to show the correctness.

▶ **Theorem 8.** Graph Motif *does not admit a polynomial-size problem kernel with respect to ℓ even if G is a tree.*

4 Colorful Graph Motif on Trees

For the combination of vertex-colored trees as input graphs and motifs that are sets, the problem becomes considerably easier. First, we show that CGM admits a linear-vertex problem kernel in this case. Moreover, we show that this problem kernel can be computed in linear time. The idea for the problem kernelization is based on two simple observations. First, in all graphs, not only in trees, the number of vertices that are not unique is bounded.

▶ **Observation 9.** *Let (M, G, χ) be an instance of* Colorful Graph Motif. *Then at most 2ℓ vertices in G are not unique.*

Proof. Let C^+ denote the set of colors that occur more than once in G and let $\mathrm{occ}(c)$ denote the number of occurrences of a color c in G. We denote $c^+ := |C^+|$, $n^+ := \sum_{c \in C^+} \mathrm{occ}(c)$, and n^- the number of unique vertices in G. By definition, no color is repeated in M, thus $|M| = c^+ + n^-$; moreover, $|V| = n^+ + n^-$. Hence, the number $\ell = |V| - |M|$ of vertices to delete satisfies $\ell = n^+ - c^+$. By definition $n^+ \geq 2c^+$, and thus we conclude that $\ell \geq n^+/2$. ◀

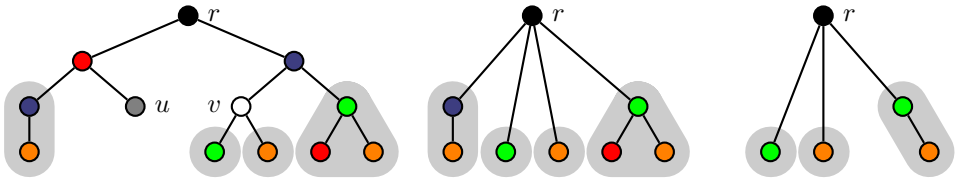

Figure 1 The two phases of the kernelization. Left: The input instance, where r, u, and v have unique colors. The pendant non-unique subtrees are highlighted by the grey background. Middle: after Phase I, all vertices on paths between unique vertices are contracted into r. Right: In Phase II, all vertices with a color that was removed in Phase I are removed together with their descendants.

Second, if there are two vertices that are unique, then the uniquely determined path between these vertices is contained in every occurrence of the motif. The kernelization accordingly removes all the vertices that lie on these paths. More precisely, these vertices are "contracted" into the root r. Afterwards, in a second phase some further vertices are removed because their colors have been used during the contraction. Eventually, this results in an instance which has at most one unique vertex and thus, by Observation 9, bounded size. For an example of the kernelization, see Figure 1. Below, we give a more detailed description.

▶ **Theorem 10.** Colorful Graph Motif *on trees admits a problem kernel with at most* $2\ell + 1$ *vertices that can be computed in* $O(n)$ *time.*

Proof. We first describe the kernelization algorithm, then we show its correctness and finally bound its running time. By Observation 9, the size bound holds if the instance has no unique vertex. Thus, we assume that there is a unique vertex in the following.

Given an instance (G, M, χ) of CGM, first root the input tree G at an arbitrary unique vertex r. Now call a subtree with root v *pendant* if it contains all descendants of v in G. Then, compute in a bottom-up fashion maximal pendant subtrees such that no vertex in this subtree is unique. Call these subtrees the *pendant non-unique subtrees*. By Observation 9, the total number of vertices in pendant non-unique subtrees is at most 2ℓ. Now the algorithm removes vertices in two phases.

Phase I. Remove from G all vertices except r that are not contained in a pendant non-unique subtree. Remove all colors of removed vertices from M. If there is a color c such that two vertices with color c are removed in this step, then return "no". Make r adjacent to the root of each pendant non-unique subtree.

Phase II. In the first step of this phase, for each color c where at least one vertex has been removed in Phase I, remove all vertices from G that have color c. In the second step of this phase, remove all descendants of these vertices. Finally, let M' denote the set of colors that are contained in the remaining instance. This completes the kernelization algorithm; the resulting instance has at most $2\ell + 1$ vertices since all vertices except r are unique. To show correctness, we first observe the following.

Claim: Every occurrence of M in G contains no vertex v that is removed during Phase II of the kernelization. This can be seen as follows. First, every occurrence of M in G contains all vertices removed during Phase I: these vertices are either unique or lie on the uniquely determined path between two unique vertices. Now consider a vertex v removed during Phase II. If v is removed in the first step of Phase II, then v has the same color c as a vertex u

removed during Phase I. Consequently, v is not contained in an occurrence of M: By the above, the occurrence contains u and it contains no other vertex with color c. Otherwise, v is removed in the second step of Phase II, because v is not connected to r. Since every occurrence of M contains r, it thus cannot contain v.

We now show the correctness of the kernelization, that is, the equivalence of the original instance (M, G, χ) and the resulting instance (M', G', χ'). First, assume that (M, G, χ) is a yes-instance. Let S_T be an occurrence of M in G, and let T denote $G[S_T]$; by the above claim, T contains only vertices that are removed during Phase I or that are contained in G'. Consider the subtree T' of G that contains all vertices of T that are not removed during the kernelization. We show that T' is connected in G' and contains all colors of M'. Connectivity can be seen as follows. First, observe that T and T' contain r. Second, any vertex $v \neq r$ of T' is contained in some pendant non-unique subtree of G. Thus, v is in T connected to r via a path that first visits only vertices of T', including the root of the pendant non-unique subtree. The root of the pendant non-unique subtree is in G' adjacent to r. Thus, each vertex $v \neq r$ has in T' a path to r which implies that T' is connected. It remains to prove that T' contains all colors of M'. Consider a color $c \in M'$. Since $c \in M'$, none of the vertices with color c are removed in Phase I of the kernelization. Moreover, since no vertex of T is removed in Phase II of the kernelization, we have that the vertex of T with color c is contained in T'. Thus, T' contains each color of M'. Finally, T' contains each color at most once since T does.

Now assume that (M', G', χ') is a yes-instance and let $S_{T'}$ be an occurrence of M' in G'. Let T denote $G[S_{T'} \cup V_I]$, where V_I is the set of vertices removed during Phase I of the kernelization. We show that T is connected and contains every color of G exactly once. To see that T is connected observe the following: Clearly, $G[\{r\} \cup V_I]$ is connected. Moreover, each vertex $v \neq r$ of T' has in T' a path to r. This path contains a subpath from v to the root r' of the pendant non-unique subtree containing v. In G, r' is adjacent to some vertex of $\{r\} \cup V_I$. Therefore, r' is connected to r in T and thus T is connected. It remains to show that T contains every color of G exactly once. Clearly, T' contains at least one vertex of each color $c \in M'$. Moreover, it also contains at least one vertex of each color $c \in M \setminus M'$ since it contains all vertices of V_I. Besides, it contains each color only once: The vertices of T' have pairwise different colors and different colors than those of the vertices of V_I. Finally, the vertices of V_I have different pairwise colors since the kernelization did not return "no".

The running time can be seen as follows. Determining the pendant non-unique subtrees can be done by a standard bottom-up procedure in linear time. Removing all vertices during Phase I can also be achieved in linear time. After removing a vertex with color c in Phase I, we label c as *occupied*. When we remove a vertex with an occupied color during Phase I, we immediately return "no". After the removal of vertices during Phase I, we can construct M' from M in linear time by removing each occupied color. Finally, we can in linear time add an edge between r and each root of the pendant non-unique subtrees and then remove all remaining vertices that have an occupied color. The final graph G' is obtained by performing a depth-first search from r, in order to include only those vertices still reachable from r. ◀

Now, let us turn to developing fast(er) FPT algorithms for CGM. It can be seen that it is possible to solve CGM in trees in time $1.62^{\ell} \cdot n^{O(1)}$, by 'branching on colors with the most occurrences' until every color appears at most twice. More precisely, for a color c that appears at least three times and some vertex v with color c, we can branch into the two cases to either delete v or to delete the at least two other vertices that have color c. The branching vector[1] for this branching rule is $(1, 2)$ or better. Now, if every color appears at

[1] For an introduction to the analysis of branching vectors, refer to [8, 12].

most twice, then CGM on trees can be solved in polynomial time [10, Lemma 2]. However, by a different branching approach, the above running time can be further reduced.

▶ **Branching Rule 11.** *If there is a color c such that there are two vertices u and v with color c that are both not leaves of the tree G, then branch into the case to delete from G either*

- *the maximal subtree containing u and all vertices w such that the path from v to w contains u, or*
- *the maximal subtree containing v and all vertices w such that the path from u to w contains v.*

Proof of correctness. No occurrence may contain vertices of both subtrees, since in this case it contains u and v which have the same color. ◀

If the rule does not apply, then one can solve the problem in linear time; here, let $\mathrm{occ}(c)$ denote the number of occurrences of a color c in G.

▶ **Lemma 12.** *Let (M, G, χ) be an instance of* Colorful Graph Motif *such that G is a tree and for each color c with $\mathrm{occ}(c) > 1$ at least $\mathrm{occ}(c) - 1$ occurrences of c are leaves of G, then (M, G, χ) can be solved in $O(n)$ time.*

Proof. For each color c with $\mathrm{occ}(c) > 1$, the algorithm simply deletes $\mathrm{occ}(c) - 1$ leaves with color c. This can be done in linear time by visiting all leaves via depth-first search, checking for each leaf in $O(1)$ time whether $\mathrm{occ}(c) > 1$ and deleting the leaf in $O(1)$ time if this is the case. The resulting graph contains each color exactly once, and it is connected since a tree cannot be made disconnected by deleting leaves. ◀

Altogether, we arrive at the following running time.

▶ **Theorem 13.** Colorful Graph Motif *can be solved in $O(\sqrt{2}^{\ell} + n)$ time if G is a tree.*

Proof. The algorithm is as follows. First, reduce the input instance in $O(n)$ time to an equivalent one with $O(\ell)$ vertices using the kernelization of Theorem 10. Now, apply Branching Rule 11. If this rule is no longer applicable, then solve the instance in $O(\ell)$ time (by applying the algorithm behind Lemma 12). Since the graph has $O(\ell)$ vertices, applicability of Branching Rule 11 can be tested in $O(\ell)$ time. Thus, the overall running time is $O(\ell)$ times the number of search tree nodes. Since each application of Branching Rule 11 creates two branches and reduces ℓ by at least two in each branch, the search tree has size $O(2^{\ell/2}) = O(\sqrt{2}^{\ell})$. The resulting running time is $O(\sqrt{2}^{\ell} \cdot \ell + n)$. Furthermore, the factor of ℓ in the running time can be removed by interleaving search tree and kernelization [16], that is, by applying the kernelization algorithm of Theorem 10 in each search tree node. ◀

References

1. Abhimanyu M. Ambalath, Radheshyam Balasundaram, Chintan Rao H., Venkata Koppula, Neeldhara Misra, Geevarghese Philip, and M. S. Ramanujan. On the kernelization complexity of colorful motifs. In *Proc. of the 5th Int'l Symp. on Parameterized and Exact Computation (IPEC'10)*, volume 6478 of *LNCS*, pages 14–25. Springer, 2010.
2. Nadja Betzler, René van Bevern, Christian Komusiewicz, Michael R. Fellows, and Rolf Niedermeier. Parameterized algorithmics for finding connected motifs in biological networks. *IEEE/ACM Trans. on Computational Biology and Bioinformatics*, 8(5):1296–1308, 2011.
3. Andreas Björklund, Petteri Kaski, and Lukasz Kowalik. Constrained multilinear detection and generalized graph motifs. *Algorithmica*, 74(2):947–967, 2016.

4 Hans L. Bodlaender, Bart M. P. Jansen, and Stefan Kratsch. Kernelization lower bounds by cross-composition. *SIAM Journal on Discrete Mathematics*, 28(1):277–305, 2014.

5 Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel bounds for disjoint cycles and disjoint paths. *Theoretical Computer Science*, 412(35):4570–4578, 2011.

6 Edouard Bonnet and Florian Sikora. The graph motif problem parameterized by the structure of the input graph. In *Proceedings of the 10th International Symposium on Parameterized and Exact Computation (IPEC'15)*, volume 43 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 319–330, 2015.

7 Sharon Bruckner, Falk Hüffner, Richard M. Karp, Ron Shamir, and Roded Sharan. Topology-free querying of protein interaction networks. *Journal of Computational Biology*, 17(3):237–252, 2010. doi:10.1089/cmb.2009.0170.

8 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

9 Holger Dell and Dieter van Melkebeek. Satisfiability allows no nontrivial sparsification unless the polynomial-time hierarchy collapses. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC'10)*, pages 251–260. ACM, 2010.

10 Michael R. Fellows, Guillaume Fertin, Danny Hermelin, and Stéphane Vialette. Upper and lower bounds for finding connected motifs in vertex-colored graphs. *Journal of Computer and System Sciences*, 77(4):799–811, 2011.

11 Michael R. Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009.

12 Fedor V. Fomin and Dieter Kratsch. *Exact Exponential Algorithms*. Springer-Verlag, 1st edition, 2010.

13 Sepp Hartung, Christian Komusiewicz, and André Nichterlein. Parameterized algorithmics and computational experiments for finding 2-clubs. *Journal of Graph Algorithms and Applications*, 19(1):155–190, 2015.

14 Russell Impagliazzo, Ramamohan Paturi, and Francis Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

15 Vincent Lacroix, Cristina G. Fernandes, and Marie-France Sagot. Motif search in graphs: Application to metabolic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):360–368, 2006.

16 Rolf Niedermeier and Peter Rossmanith. A general method to speed up fixed-parameter-tractable algorithms. *Information Processing Letters*, 73(3-4):125–129, 2000.

17 Ron Y. Pinter, Hadas Shachnai, and Meirav Zehavi. Deterministic parameterized algorithms for the graph motif problem. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS'14)*, volume 8635 of *Lecture Notes in Computer Science*, pages 589–600. Springer, 2014. doi:10.1007/978-3-662-44465-8.

18 Ron Y. Pinter and Meirav Zehavi. Algorithms for topology-free and alignment network queries. *J. of Discrete Algorithms*, 27:29–53, 2014. doi:10.1016/j.jda.2014.03.002.

19 Imran Rauf, Florian Rasche, François Nicolas, and Sebastian Böcker. Finding maximum colorful subtrees in practice. *Journal of Computational Biology*, 20(4):311–321, 2013.

20 Roded Sharan and Trey Ideker. Modeling cellular machinery through biological network comparison. *Nature Biotechnology*, 24(4):427–433, 2006.

21 Florian Sikora. An (almost complete) state of the art around the graph motif problem. Technical report, LIGM Université Paris-Est, March 2012. URL: http://www.lamsade.dauphine.fr/~sikora/pub/GraphMotif-Resume.pdf.