Sharp Tractability Borderlines for Finding Connected Motifs in Vertex-Colored Graphs

Michael R. Fellows[1,2,*], Guillaume Fertin[3],
Danny Hermelin[4,**], and Stéphane Vialette[5]

[1] The University of Newcastle, Callaghan NSW 2308 - Australia
mike.fellows@cs.newcastle.edu.au
[2] Institute of Advanced Study, Durham University,
Durham DH1 3RL - United Kingdom
[3] Laboratoire d'Informatique de Nantes-Atlantique (LINA), FRE CNRS 2729
Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3 - France
fertin@lina.univ-nantes.fr
[4] Department of Computer Science, University of Haifa,
Mount Carmel, Haifa 31905 - Israel
danny@cri.haifa.ac.il
[5] Laboratoire de Recherche en Informatique (LRI), UMR CNRS 8623
Faculté des Sciences d'Orsay - Université Paris-Sud, 91405 Orsay - France
vialette@lri.fr

Abstract. We study the problem of finding occurrences of motifs in vertex-colored graphs, where a motif is a multiset of colors, and an occurrence of a motif is a subset of connected vertices whose multiset of colors equals the motif. This problem has applications in metabolic network analysis, an important area in bioinformatics. We give two positive results and three negative results that together draw sharp borderlines between tractable and intractable instances of the problem.

1 Introduction

Vertex-colored graph problems have numerous applications in bioinformatics. Sandwich problems have applications in DNA physical mapping [6,13,15] and in perfect phylogeny [8,19], while vertex-recoloring problems arise in protein-protein interaction networks and phylogenetic analysis [7,9,20]. In this paper, we consider another natural vertex-colored graph problem with an interesting application in bioinformatics:

Graph Motif:

Input: A vertex-colored graph G and a multiset of colors M.

Question: Does G have a connected subset of vertices whose multiset of
colors equals M?

The Graph Motif problem was introduced in a slightly more general form
by Lacroix, Fernandes, and Sagot (who allowed multiple colors per vertex) in the
context of metabolic network analysis, an important area in bioinformatics [18].
There, vertices correspond to chemical compounds or reactions, and edges cor-
respond to interactions between these compounds and reactions. The vertex col-
oring is used to distinguish between different types of chemicals and reactions.
The transmission of information in these networks can usually be described as a
chain of interacting chemicals, in which chemical interactions enable each chem-
ical in the path to modify its successor so as to transmit biological data. In this
scenario, connected motifs can correspond to relatively functional independent
modules of the network which consist of a specific set of chemical compounds
and reactions. It is argued in [18] that a method for a rational decomposition of
a metabolic network into relatively independent functional subsets is essential
for a better understanding of the modularity and organization principles in the
network. We refer the reader to [11,18] for more biological background of the
problem. We also refer to [16,17] for related work and relevant background.

In [18], Graph Motif is proved to be NP-complete even if the given vertex-
colored graph is a tree, but fixed-parameter tractable in this case when param-
eterized by the size of the given motif (*i.e.* $|M|$). However, as observed by [18],
their fixed-parameter does not apply when the vertex-colored graph is a general
graph. For this case they only provided a heuristic algorithm which works well
in practice. This motivates us to further investigate the tractability landscape of
Graph Motif, and in particular, to investigate it under different parameters
which govern the structure of its input. We give an extensive analysis for Graph
Motif, applying techniques from both classical and parameterized complexity,
that unravels sharp borderlines between tractable and intractable instances of
the problem. More specifically, we give two algorithms and three hardness results
that together imply:

1. For motifs of unbounded size, Graph Motif is NP-complete already for
   trees of maximum degree 3, even if the motif is a set of colors rather than a
   multiset. For motifs of logarithmic size (in the number of vertices of G), the
   problem is polynomial-time solvable in any general graph.
2. Graph Motif is NP-complete for motifs with 2 colors, even if G is bi-
   partite with maximum degree 4. However, it is polynomial-time solvable in
   constant treewidth graphs for motifs consisting of any constant number of
   colors (and arbitrary size). When the number of colors in the motif is taken
   as a parameter, Graph Motif is W[1]-hard even in case G is a tree.

The rest of the paper is organized as follows. In the reminder of this section we
discuss notations that will be used throughout the paper. In Section 2, we give

two NP-hardness results that will motivate the rest of our discussion. Following this, in Section 3 we present a fixed-parameter algorithm (parameterized by $|M|$) that that applies for any general graph. In Section 4 we discuss the case when G has bounded treewidth. Finally, in Section 5, we show that GRAPH MOTIF is W[1]-hard on trees when parameterized by the number of colors in M.

Basic notation and terminology: Throughout the paper, we use $G = (V(G),$ $E(G))$ to denote our given vertex-colored graph, and $n = |V(G)|$ to denote its order. For a vertex $v \in V(G)$, we use $\chi(v)$ to denote the color of v, and for a vertex subset $V \subseteq V(G)$, we let $\chi(V)$ denote the multiset of colors $\bigcup_{v \in V} \chi(v)$. For any vertex subset $V \subseteq V(G)$, we let $G[V]$ denote the subgraph of G induced by V, *i.e.* the subgraph on V along with all edges of G that connect vertices in V. We assume w.l.o.g. that G is connected.

A motif M is a multiset of colors. If M is in fact a set rather than a multiset, we say that M is *colorful*. Given a subset of vertices $V \subseteq V(G)$, $|V| = |M|$, we say that V is *colored by the colors of* M, if $\chi(V) = M$. For V to be an *occurrence* of M, we require not only for V to be colored by the colors of M, but also for $G[V]$ to be connected. If this is in fact the case, we say that M *occurs at* v for any vertex $v \in V$. In these terms, the GRAPH MOTIF problem is the problem of determining whether a given motif M occurs at any vertex of a given vertex-colored graph G. We assume w.l.o.g. that $\chi(v) \in M$ for any $v \in V(G)$.

Our analysis is based both on the classical and parameterized complexity frameworks. Readers unfamiliar with these subjects are referred to [12,14].

2   Tight NP-Hardness Results

As mentioned previously in Section 1, GRAPH MOTIF is already known to be NP-complete for trees in [18]. Our aim in this section is to tighten this result by showing that GRAPH MOTIF remains hard for highly restrictive graph classes, even if we restrict ourselves to motifs which are sets rather than multisets, or to motifs which consist of a small number of colors.

We first consider colorful motifs. Recall that a motif M is colorful if it consists of $|M|$ distinct colors. At first sight, it might seem that occurrences of colorful motifs should be easier to find, at least for certain types of graphs. Unfortunately, the following theorem proves that this is apparently not the case.

Theorem 1. GRAPH MOTIF *is NP-complete, even if M is colorful and G is a tree of maximum degree three.*

Proof. GRAPH MOTIF is clearly in NP. To prove NP-hardness, we present a reduction from the well known NP-complete problem 3-SAT [14]. Recall that 3-SAT asks to determine whether a given 3-CNF formula is satisfiable, that is, whether there is a truth assignment to the boolean variables of the formula, such that the value of the formula under this assignment is 1. The problem remains hard even if each variable appears in at most three clauses and each literal (*i.e.* variable with or without negation) appears in at most two clauses [14]. Hence, we restrict ourselves in our proof to formulas of this type.

Let an instance of 3-SAT be given in the form of a 3-CNF formula $\Phi = c_1 \wedge \cdots \wedge c_m$ over variables $x_1, \ldots, x_n$ such that $|\{c_j \mid x_i \in c_j\}| \leq 2$ and $|\{c_j \mid \bar{x}_i \in c_j\}| \leq 2$ for all $1 \leq i \leq n$. We construct an instance for GRAPH MOTIF as follows. The colored graph G initially consists of a path of n vertices, each colored by a distinct color in $1, \ldots, n$. To a vertex colored i in this path, $1 \leq i \leq n$, we connect a new vertex colored i'. To a vertex colored i', $1 \leq i \leq n$, we connect a pair of new non-adjacent vertices, both colored x_i. Conceptually, each vertex in this pair corresponds to a different truth assignment for x_i. If a truth assignment to variable x_i satisfies clause c_j, we connect a new vertex colored c_j to the vertex colored x_i which corresponds to this assignment. This is done for every $x_i \in \{x_1, \ldots, x_n\}$ and every $c_j \in \{c_1, \ldots, c_m\}$. We conclude our construction by specifying M to be the set of colors $\{1, \ldots, n, 1', \ldots, n', x_1, \ldots, x_n, c_1, \ldots, c_m\}$. Note that G and M are as required by the theorem.

The construction above is clearly polynomial. Hence, to complete the proof, we are left to show that M occurs in G if and only if Φ is satisfiable. For the first direction, assume that there exists a truth assignment ϕ which satisfies Φ. Let $N \subseteq V(G)$ be the subset of vertices in G which are colored by the colors in $\{1, \ldots, n, 1', \ldots, n'\}$, and let $X \subseteq V(G)$ be the subset of vertices which correspond to assignment ϕ. Hence, X consists of n vertices which are colored by the colors in $\{x_1, \ldots, x_n\}$, and $N \cup X$ induces a connected subgraph. Since ϕ satisfies every clause in Φ, by construction of G there is a vertex colored c_j in the neighborhood of X for every $1 \leq j \leq m$. In other words, there exists a subset C of neighbors of vertices in X which is colored by the colors in $\{c_1, \ldots, c_m\}$. It follows that $V = N \cup X \cup C$ is connected and is colored by the colors of M, and therefore is an occurrence of M in G.

For the converse direction, assume there exists an occurrence V of M in G. Let $X \subseteq V$ be the vertices colored by the colors in $\{x_1, \ldots, x_n\}$, $C \subseteq V$ be the vertices colored by the colors in $\{c_1, \ldots, c_m\}$, and ϕ the truth assignment corresponding to X. By construction, a vertex colored c_j is connected in G to a vertex colored x_i if and only if the truth assignment corresponding to this vertex satisfies clause c_j. Since C contains all colors in $\{c_1, \ldots, c_m\}$, and since vertices in C are connected only to vertices in X, it follows that ϕ satisfies every clause in Φ, and so it satisfies Φ itself. $\square$

Theorem 1 implies that for motifs of unbounded cardinality, there are not many interesting graph classes left for which GRAPH MOTIF becomes polynomial-time solvable. Note that if G is a tree of maximum degree two, then G is actually a path, and GRAPH MOTIF becomes trivial (simply search through all subpaths of length $|M|$). However, the motif in the construction above is not only of unbounded size, it also consists of an unbounded number of colors. One might hope that for motifs which consist of only a small number of colors, GRAPH MOTIF would become polynomial-time solvable. The following theorem shows that this is not the case in a very sharp sense.

Theorem 2. GRAPH MOTIF *is* NP-*complete, even if M consists of two colors, and G is bipartite with maximum degree four.*

Proof. We reduce from the EXACT COVER BY 3-SETS (X3C) problem, which is known to be NP-complete [14]. Recall that, given a set $X = \{x_1, x_2, \ldots, x_{3q}\}$ and a collection $S = \{s_1, s_2, \ldots, s_n\}$ of 3-element subsets of X, the X3C problem asks to determine whether there exists an exact cover of X in S, *i.e.* a sub-collection $C \subseteq S$ such that every element of X is included in exactly one subset $s_i \in C$. The problem is hard even if each element of X appears in at most three sets of S [14], so we restrict ourselves in the proof to instances of this type.

Let $\langle X, S \rangle$ be an arbitrary instance of the X3C problem with $|\{s_j \in S \,|\, x_i \in s_j\}| \le 3$ for all $x_i \in X$. We show how to construct a motif M and a colored graph G in such a way that there exists an exact cover of X in S if and only if M occurs in G. First, we define M so as it contains $2n + 3q$ *white* elements and q *black* elements. Then, we define G by $V(G) = X \cup S \cup S' \cup S''$ and $E(G) = E_1 \cup E_2 \cup E_3 \cup E_4$, where $S' = \{s_1', s_2', \ldots, s_n'\}$ and $S'' = \{s_1'', s_2'', \ldots, s_n''\}$ are dummy copies of S, and E_1, E_2, E_3, E_4 are defined by: $E_1 = \{\{x_i, s_j\} \,|\, x_i \in s_j\}$, $E_2 = \{\{s_i, s_i'\} \,|\, 1 \le i \le n\}$, $E_3 = \{\{s_i', s_i''\} \,|\, 1 \le i \le n\}$, and $E_4 = \{\{s_i'', s_{i+1}'\} \,|\, 1 \le i \le n-1\}$. The vertices of $X \cup S' \cup S''$ are colored white and the vertices of S are colored black. It is easily seen that G and M are as required by the theorem, and that our construction can be carried out in polynomial time.

Let us now argue that there exists an exact cover $C \subseteq S$ of X if and only if M occurs in G. For the first direction, suppose that there exists an exact cover $C \subseteq S$ of X. Consider the subset of vertices $V = X \cup C \cup S' \cup S''$. First note that V consists of $q = |C|$ black vertices and $2n + 3q = |X \cup S' \cup S''|$ white vertices. Second, since C is a cover of X, every vertex of X is connected to some vertex in C, and C is connected to $S' \cup S''$, so V itself is connected. It follows that V is an occurrence of M, and M occurs in G.

Conversely, suppose that there exists an occurrence $V \subseteq V(G)$ of M in G. Observe that M contains $2n + 3q$ white elements, and since exactly $2n + 3q$ vertices of G are colored white, we must have $X \cup S' \cup S'' \subset V$. The remaining q vertices in V are q black vertices from S. By construction, we do not have an edge between two vertices of X, nor between a vertex of X and a vertex of $S' \cup S''$. Therefore, since V is connected, each vertex of X has to be adjacent to at least one vertex in $V \cap S$. But $|X| = 3q$ and each vertex in S is connected to exactly 3 vertices in X. Then it follows that no two vertices of $V \cap S$ share a common neighbor in X, and $C = V \cap S$ is an exact cover of X in S.   □

3   A General Fixed-Parameter Algorithm

We now turn to show that GRAPH MOTIF is fixed-parameter tractable for parameter $k = |M|$ on any general graph. More specifically, we present an $\mathcal{O}(2^{\mathcal{O}(k)} n^2 \lg n)$ algorithm for the problem, which implies that GRAPH MOTIF for motifs of $\mathcal{O}(\lg n)$ size is polynomial-time solvable. This is in striking contrast to the sharp hardness results given in the previous section. Our algorithm is based on the *color-coding technique* introduced by Alon *et al.* [2], whose derandomized version crucially relies on the notion of perfect hash families:

Definition 1 (Perfect Hash Family). *A family $\mathcal{F}$ of functions from $V(G)$ to $\{1, \ldots, k\}$ is* perfect *if for any subset $V \subseteq V(G)$ of k vertices there is a function $f \in \mathcal{F}$ which is one-to-one on V.*

Suppose M has an occurrence V in G, and suppose we are provided with a perfect family $\mathcal{F}$ of functions from $V(G)$ to $\{1, \ldots, k\}$. 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. For a given $L \subseteq \{1, \ldots, k\}$, we define $\mathcal{M}_L(v)$ to be the family of all motifs $M' \subseteq M$, $|M'| = |L|$, for which there exists an occurrence V' with $v \in V'$, such that the set of (unique) labels that f assigns to V' is exactly L. Since M occurs in G, we know that $M \in \mathcal{M}_{\{1, \ldots, k\}}(v)$ for some $v \in V(G)$. Hence, to determine whether M occurs in G, we apply dynamic programming to compute $\mathcal{M}_L(v)$ for all $v \in V(G)$ and $L \subseteq \{1 \ldots, k\}$.

Fix L to be some subset of $\{1, \ldots, k\}$, and let v be any vertex of G. Our goal is to compute $\mathcal{M}_L(v)$ assuming $\mathcal{M}_{L'}(u)$ has been previously computed for every vertex $u \in V(G)$ and any $L' \subseteq L \setminus \{f(v)\}$. The straightforward approach is to consider small motifs occurring at neighbors of v. However, a motif occurring at v might be the union of motifs occurring at any number of neighbors of v, and so this approach might require exponential running time in n. We therefore present an alternative method for computing $\mathcal{M}_L(v)$, which we call the *batch procedure*, that uses an even more naive approach, but one that requires exponential-time only with respect to k. Notice that while the motifs computed by the batch procedure are in general multisets of colors, the batch procedure always considers sets of distinct labels.

`Batch Procedure`(L, v)

- Define $\mathcal{M}$ to be the family of all pairs (M', L') such that $M' \subseteq M \setminus \{\chi(v)\}$, $L' \subseteq L \setminus \{f(v)\}$, and $M' \in \mathcal{M}_{L'}(u)$ for some $u \in N(v)$.
- Run through all pairs of $(M', L'), (M'', L'') \in \mathcal{M}$ and determine whether $M' \cup M'' \subseteq M \setminus \{\chi(v)\}$, and whether $L' \cap L'' = \emptyset$. If there is such a pair, add $(M' \cup M'', L' \cup L'')$ to $\mathcal{M}$ and repeat this step. Otherwise, continue to the next step.
- Set $\mathcal{M}_L(v)$ to be all motifs $M' \cup \{\chi(v)\}$ where $(M', L') \in \mathcal{M}$ and $L' = L \setminus \{f(v)\}$.

Lemma 1. *For any $v \in V(G)$ and $L \subseteq \{1, \ldots, k\}$, the batch procedure correctly computes $\mathcal{M}_L(v)$ assuming $\mathcal{M}_{L'}(u)$ is given for every neighbor u of v and every subset of labels $L' \subseteq L \setminus \{f(v)\}$.*

Proof. Let $\mathcal{M}$ be the family of pairs computed by the batch procedure. Consider any pair $(M', L') \in \mathcal{M}$ with $L' = L \setminus \{f(v)\}$. By construction, $M' \subseteq M \setminus \{\chi(v)\}$ and can be written as $M' = M'_1 \cup \cdots \cup M'_\ell$, where each M'_i, $1 \leq i \leq \ell$, is a motif that has an occurrence V'_i which includes a neighbor of v. Furthermore, each V'_i is labeled by a unique set of labels L'_i such that $L'_i \cap L'_j = \emptyset$ for all $1 \leq j \leq \ell$, $j \neq i$. It follows that all the V'_i's are pairwise disjoint, and that $\{v\} \cup V'_1 \cup \cdots \cup V'_\ell$ is connected. Hence, $M' \cup \{\chi(v)\}$ has an occurrence in G which is labeled by $L' \cup \{f(v)\} = L$, and so $M' \cup \{\chi(v)\} \in \mathcal{M}_L(v)$.

On the other hand, consider a motif $M' \cup \{\chi(v)\} \in \mathcal{M}_L(v)$. Then by definition, $M' \cup \{\chi(v)\}$ has an occurrence $V' \cup \{v\}$ such that the set of labels that f assigns $V' \cup \{v\}$ is L. Let $V'_1, \ldots, V'_\ell$ be the connected components of the induced subgraph $G[V']$. Since $V' \cup \{v\}$ is connected, every V'_i, $1 \leq i \leq \ell$, includes a neighbor of v. Furthermore, letting L'_i denote the set of labels that f assigns V'_i for every $1 \leq i \leq \ell$, we have $L' \subseteq L \setminus \{f(v)\}$ and $L'_i \cap L'_j = \emptyset$ for all $1 \leq i, j \leq \ell$. It is now easy to see that the batch procedure will eventually compute the pair $(M', L \setminus \{f(v)\})$ in its second step, and hence $M' \cup \{\chi(v)\}$ will be added to $\mathcal{M}_L(v)$ in its final step. $\square$

Lemma 2. *Given a labeling function $f : V(G) \rightarrow \{1, \ldots, k\}$, one can use the batch procedure iteratively in order to determine in $\mathcal{O}(2^{5k}kn^2)$ time whether there is an occurrence of M which is distinctly labeled by f.*

Proof. To prove the lemma, let us first analyze the complexity of a single invocation of the batch procedure. In its first step, the batch procedure searches through at most $2^k n$ motifs families, each consisting of at most 2^k motifs. Hence, this step requires $\mathcal{O}(2^{2k}kn)$ time. For the second step, notice that number of distinct motif and label-subset pairs is bounded by 2^{2k}, and so the number of times the second step is repeated is also bounded by this term. Since each iteration of this step can be computed in $\mathcal{O}(2^{2k}k)$ time, it follows that the second step requires $\mathcal{O}(2^{4k}k)$ time. Accounting also for the third step, the total time of one invocation of the batch procedure is therefore $\mathcal{O}(2^{4k}k + 2^{2k}kn) = \mathcal{O}(2^{4k}kn)$.

It now can easily be seen that due to Lemma 1, one needs to invoke the batch procedure at most $2^k n$ times in order to obtain $\mathcal{M}_L(v)$ for every vertex $v \in V(G)$ and every label subset $L \subseteq \{1, \ldots, k\}$. It follows that in $\mathcal{O}(2^{5k}kn^2)$ time one can obtain all necessary information to determine whether M has an occurrence which is distinctly labeled by f, and so the lemma follows. $\square$

Note that in case M is colorful, the vertex-coloring of G distinctly colors any occurrence of M, and therefore, in this case we can determine whether M occurs in G within the time complexity given in Lemma 2. For general multiset motifs, we use the result of Alon *et al.* [2] who show how to efficiently construct a family $\mathcal{F}$ of $\mathcal{O}(2^{\mathcal{O}(k)} \lg n)$ functions from $V(G)$ to $\{1, \ldots, k\}$ which is perfect. This construction builds on an earlier slightly less efficient construction of [21] and requires $\mathcal{O}(2^{\mathcal{O}(k)}n \lg n)$ time. Using this and Lemma 2, we obtain a $\mathcal{O}(2^{\mathcal{O}(k)}n^2 \lg n)$ algorithm for GRAPH MOTIF.

Theorem 3. GRAPH MOTIF *can be solved in $\mathcal{O}(2^{\mathcal{O}(k)}n^2 \lg n)$ time.*

4   Bounded Treewidth Graphs

The treewidth parameter of graphs [22] plays a central role in designing exact algorithms for many NP-hard graph problems [3,4,5,10]. Among numerous frameworks developed over the years, we adopt the parsing mechanism developed for bounded treewidth graphs in [1]. For motifs which consist of a constant

number of colors c and graphs with treewidth smaller than some constant ω, we present a polynomial-time algorithm with running time $\mathcal{O}(n^{2c\omega+2c+2})$. This should be compared with the sharp hardness result of Theorem 2. Moreover, our algorithm can also be analyzed as a fixed-parameter algorithm for parameters ω and k which outperforms the algorithm of the previous section when the treewidth of G is sufficiently small. Due to space limitations, we only present a sketch of our result.

Theorem 4. *Let ω be any positive constant. Then* GRAPH MOTIF *can be solved in $\mathcal{O}(n^{2c\omega+2c+2})$ time, when G has treewidth less than ω and M consists of c colors.*

Proof (sketch). The proof is sketched as follows. We employ the parsing operator point of view on bounded treewidth. In particular, we use the notion of ω-boundaried graphs, where an ω-boundaried graph is no more than a graph with ω distinguished vertices, each distinctly labeled by a label in $\{1, \ldots, \omega\}$, which are referred to as boundary vertices. The boundary vertices, together with ω-operators, allow the construction of ω-boundaried graphs from smaller ω-boundaried graphs. The ω-operators are:

1. The null operator $\emptyset$ which creates the trivial boundaried graph with isolated vertices.
2. The binary operator $\oplus$ that takes the disjoint union of two ω-boundaried graphs and then identifies the ith boundary vertex of the first graph with the ith boundary vertex of the second graph.
3. The unary operator that introduces a new isolated vertex and makes this the new vertex 1 of the boundary. The old vertex 1 is removed from the boundary but not from the graph.
4. The unary operator that adds an edge between vertex 1 and vertex 2 of the boundary.
5. Unary operators that permute the labels of the boundary vertices.

A parse tree is an at-most binary rooted tree with labels corresponding to ω-operators. The leaves are labeled with $\emptyset$, the internal unary nodes are labeled with unary operators, and the internal binary nodes are labeled with the binary operator $\oplus$. Each rooted subtree of a parse tree corresponds to an ω-boundaried graph, where the graph at each parent is obtained by applying the corresponding operator of the parent on the ω-boundaried graph(s) of its child(ren). We say that a parse tree *parses* an ω-boundaried graph H, if H corresponds to the ω-boundaried graph of the root. We extend this definition to any graph, by simply assuming that the final parsing operator removes all vertices from the boundary. Any graph of treewidth less than ω can be parsed by a parse tree with $\mathcal{O}(\omega n)$ nodes [1].

Define a *checklist item* for a w-boundaried graph to consist of the following information: (1) A partition π of the set of boundary vertices. Let X denote the set of boundary vertices, and write $\pi = (X_1, \ldots, X_r)$ where $r \leq \omega$, and the X_i denote the sets of the partition π. (2) A *motif family* $\mathcal{M}_\pi = (M_0, M_1, \ldots, M_r)$ of

length $r+1$, where each M_i is non-empty except maybe M_0 and $M_i \subseteq M$. (Note that the number of distinct checklist items is at most $\omega^\omega (n^c)^{\omega+1} = w^w n^{c\omega+c}$, where a better bound is given by replacing w^w with $Bell(\omega)$, the number of distinct partitions of an ω-element set.)

We say that a *checklist item* α as above is *positive* for a ω-boundaried graph A if there is a set of $r+1$ vertex-disjoint subsets $V_0, \ldots, V_r \subseteq V(A)$ satisfying the following conditions:

1. $V_0 \cap X = \emptyset$.
2. For $i = 1, \ldots, r$, $V_i \cap X = X_i$.
3. For $i = 0, \ldots, r$, V_i is an occurrence of M_i in A

Define the *inventory $inv(A)$* of the w-boundaried graph A to be the set of all checklist items that are positive for A.

Claim. Whether a motif occurs in a ω-boundaried graph A can be determined from $inv(A)$ in time linear in the size of the inventory.

Our algorithm proceeds as follows. It first computes a parse tree of G, and then computes, from the leaves up to the root, the inventories of the ω-boundaried graphs corresponding to the nodes of the ω parse tree. Let A be the trivial boundary graph obtained by the null operator $\emptyset$. In this base case, $inv(A)$ consists of single checklist item with a partition $\pi = (X_1, \ldots, X_r)$ that partitions the boundary vertices into singletons, and motif family $\mathcal{M}_\pi = (\emptyset, M_1, \ldots, M_r)$ where M_i consists of the color of the single boundary vertex in $X_i \in \pi$, for all $i = 1, \ldots, r$. For boundary graphs obtained by unary operations, computing the inventory is almost equally easy.

Claim. One can compute $inv(op(A))$ from $inv(A)$.

We proceed to describe the computation for the $\oplus$ operator. Let A and B be two boundaried graphs over the same boundary vertex set X. If $\alpha \in inv(A)$ and $\beta \in inv(B)$ then we define the checklist item $\alpha \oplus \beta$ as follows. As per the definition of a checklist item, we must give two pieces of information to describe $\alpha \oplus \beta$: (1) a partition $\pi_{\alpha \oplus \beta}$ of X, and (2) a motif family $\mathcal{M}_{\pi_{\alpha \oplus \beta}}$ for this partition. Let π_α (π_β) denote the partition of X for the checklist item α (β). Let $\equiv_\alpha$ ($\equiv_\beta$) be the equivalence relation on X defined by π_α (π_β). The partition $\pi_{\alpha \oplus \beta}$ corresponds to the reflexive and transitive closure of the relation $\equiv_\alpha \cup \equiv_\beta$. The motif family $\mathcal{M}_{\pi_{\alpha \oplus \beta}}$ is obtained by adding and subtracting colors of the motifs of $\mathcal{M}_{\pi_\alpha}$ and $\mathcal{M}_{\pi_\beta}$ in the natural way.

Claim. We can compute $inv(A \oplus B)$ as $\{\alpha, \beta, \alpha \oplus \beta \,|\, \alpha \in inv(A), \beta \in inv(B)\}$.

Hence given a parse tree of G, we have to perform at most ωn such "multiplications of inventories". Since each inventory has size bounded by $\omega^\omega n^{c\omega+c}$, and since a single multiplication between two inventories requires $\mathcal{O}(n)$ time, this gives a running time of $\mathcal{O}(n^{2c\omega+2c+2})$ for all inventory multiplications. Since the computation on the $\oplus$ operator requires more time than the computation on any other operator, the entire algorithm runs within this time bound. $\qquad\square$

Showing that our algorithm is a fixed-parameter algorithm for parameters ω and k involves pretty much the same analysis. The only difference is that here we bound the total number of distinct checklist items by $\omega^\omega (2^k)^{\omega+1} = w^w 2^{k\omega+k}$.

5   On Trees and Motifs with Bounded Number of Colors

Although Theorem 4 gives a nice complementary result to the sharp hardness result of Theorem 2, it still leaves a certain gap. In the following section we aim to close this gap, by proving that GRAPH MOTIF, parameterized by the number of colors c in M, is W[1]-hard for trees. Essentially, this means that unless unlikely collapses occur in parameterized complexity theory, there is no fixed-parameter algorithm for parameter c, even in the restricted case of trees. We refer readers unfamiliar with the concept of parameterized reductions to [12].

Theorem 5. *The* GRAPH MOTIF *problem, parameterized by the number of colors c in the motif, is* W[1]-*hard for trees.*

Proof. We present a reduction from CLIQUE which is known to be W[1]-hard [12]. Recall that in CLIQUE we are given a simple graph H and an integer κ, the parameter, and we are asked to determine whether H has a subset of κ vertices which are pairwise adjacent. Given an instance $\langle H, \kappa \rangle$ of CLIQUE, we describe a rooted tree $G = T$ colored with $1 + \kappa + 2\kappa(\kappa - 1) + \binom{\kappa}{2}$ colors. We let m denote the number of edges of H, *i.e.* $m = |E(H)|$.

- The root of T is colored a.
- The root has $\kappa \cdot |V(H)|$ children organized into κ groups $S(1) \ldots S(\kappa)$. The group of $|V(H)|$ children $S(i)$ consists of the nodes $s(i, u)$, where $u \in V(H)$. The color of each node in $S(i)$ is $b(i)$.
- From each node $s(i, u)$ hang $\kappa - 1$ groups of paths. The groups are $P(i, u, j)$ for every $j \in \{1 \ldots, \kappa\} \setminus \{i\}$. There is one path $p(i, u, j, v) \in P(i, u, j)$ for each edge $\{u, v\} \in E(H)$ that is incident to u in H.

The path $p(i, u, j, v)$ begins with a vertex colored $c(i, j)$ and ends with a vertex colored $d(i, j)$, and otherwise consists of some number $m(i, u, j, v)$ of internal vertices colored by $e(i, j) = e(j, i)$. There is an important detail to note here. If $i < j$, then $c(i, j)$ and $c(j, i)$ are different colors, whereas $e(i, j)$ and $e(j, i)$ are the same color. We call the colors $e(i, j)$ the *edge colors*. The number $m(i, u, j, v)$ is calculated as follows. Number the edges in $E(H)$ from 1 to m, letting $l(uv)$ denote the number assigned to the edge $\{u, v\} \in E(H)$. We define:

$$m(i, u, j, v) = \begin{cases} l(uv) & i < j \\ 3m - l(uv) & i > j. \end{cases}$$

The motif M consists of one of each of every color other than the edge colors, and $3m$ elements colored by each edge color. Thus, M consists of $c = 1 + \kappa + 2\kappa(\kappa - 1) + \binom{\kappa}{2}$ different colors, and $|M| = 1 + \kappa + 2\kappa(\kappa - 1) + 3m\binom{\kappa}{2}$. This completes the construction of our instance $\langle (G, M), c \rangle$ for GRAPH MOTIF.

We claim that H has a subset of κ pairwise adjacent vertices if and only if T has a subtree T' which is an occurrence of M. Suppose that the vertices $v_1, ..., v_\kappa$ are pairwise adjacent in H. The subtree T' consists of:

- The root which is colored a.
- The κ children of the root $s(i, v_i)$ for all $1 \leq i \leq \kappa$, where $s(i, v_i)$ is colored $b(i)$.
- The $\kappa(\kappa - 1)$ paths $p(i, v_i, j, v_j)$. The path $p(i, v_i, j, v_j)$ begins with a node colored $c(i, j)$ and ends with a node colored $d(i, j)$ for all $1 \leq i, j \leq \kappa$, $i \neq j$. Note that the path $p(i, v_i, j, v_j)$ is pendant from $s(i, v_i)$ since v_i and v_j are adjacent in H. Together, the two complementary paths $p(i, v_i, j, v_j)$ and $p(j, v_j, i, v_i)$ contain $3m$ nodes colored $e(i, j)$.

In the other direction, suppose that the subtree T' of T is an occurrence of M. Then T' must include the root of T, since it is the only node colored a. Furthermore, T' must contain exactly one node in each of the groups $S(i)$, $1 \leq i \leq \kappa$, since nodes in each $S(i)$ are all colored $b(i)$. Suppose these nodes are $s(1, v_1), \ldots, s(\kappa, v_\kappa)$. We argue that the vertices $v_1, ..., v_\kappa$ are pairwise adjacent in H.

In order for T' to be an occurrence of M in T, T' must contain exactly one pendant path in each of the groups of paths $P(i, v_i, j)$ for any $1 \leq i, j \leq \kappa$, $i \neq j$, and nothing further. To see this, note that T' must contain at least one path in each of the groups of paths $P(i, v_i, j)$ in order for T' to contain a node colored $d(i, j)$. But containing one such path prevents T' from including any nodes of other paths in $P(i, v_i, j)$, else T' would contain too many nodes of color $c(i, j)$.

It follows that for any pair of indices i, j with $1 \leq i < j \leq \kappa$, T' includes exactly two paths $p(i, v_i, j, x)$ and $p(j, v_j, i, y)$ that contain nodes of color $e(i, j) = e(j, i)$. Since M contains exactly $3m$ elements colored by $e(i, j)$, it follows that $x = v_j$ and $y = v_i$, since $p(i, v_i, j, v_j)$ and $p(j, v_j, i, v_i)$ are the only two paths in T with nodes colored $e(i, j)$ that together have exactly $3m$ nodes of this color. But then, by construction of T, v_i and v_j must be adjacent in H. $\qquad\square$

References

1. Abrahamson, K.R., Fellows, M.R.: Finite automata, bounded treewidth, and well-quasiordering. In: N. Robertson and P. Seymoued (eds.) Graph Structure Theory, pp. 539–564 (1993)
2. Alon, N., Yuster, R., Zwick, U.: Color coding. Journal of the ACM 42(4), 844–856 (1995)
3. Arnborg, S.: Efficient algorithms for combinatorial problems on graphs with bounded decomposability. A survey. BIT Numerical Mathematics 25(1), 2–23 (1985)
4. Arnborg, S., Proskurowski, A.: Linear time algorithms for NP-hard problems restricted to partial k-trees. Discrete Applied Mathematics 23, 11–24 (1989)
5. Bodlaender, H.L.: A tourist guide through treewidth. Acta Cybernetica 11, 1–23 (1993)

6. Bodlaender, H.L., Fluiter, L.E.: Intervalizing k-colored graphs. In: Fülöp, Z., Gecseg, F. (eds.) Automata, Languages and Programming. LNCS, vol. 944, pp. 87–98. Springer, Heidelberg (1995)
7. Bodlaender, H.L., Fellows, M.R., Langston, M.A., Ragan, M.A., Rosamond, F.A., Weyer, M.: Kernelization for convex recoloring. In: Proceedings of the 2nd workshop on Algorithms and Complexity in Durham (ACiD), pp. 23–35 (2006)
8. Bodlaender, H.L., Fellows, M.R., Warnow, T.: Two strikes against perfect phylogeny. In: Kuich, W. (ed.) Automata, Languages and Programming. LNCS, vol. 623, pp. 273–283. Springer, Heidelberg (1992)
9. Chor, B., Fellows, M.R., Ragan, M.A., Rosamond, F.A., Snir, S.: Connected coloring completion for general graphs: Algorithms and complexity – Manuscript (2007)
10. Corneil, D.G., Keil, J.M.: A dynamic programming approach to the dominating set problem on k-trees. SIAM Journal on Algebraic and Discrete Methods 8(4), 535–543 (1987)
11. Deville, Y., Gilbert, D., Helden, J.V., Wodak, S.J.: An overview of data models for the analysis of biochemical pathways. Briefings in Bioinformatics 4(3), 246–259 (2003)
12. Downey, R., Fellows, M.: Parameterized Complexity. Springer, Heidelberg (1999)
13. Fellows, M.R., Hallett, M.T., Wareham, H.T.: DNA physical mapping: Three ways difficult. In: Lengauer, T. (ed.) ESA 1993. LNCS, vol. 726, pp. 157–168. Springer, Heidelberg (1993)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
15. Golumbic, M., Kaplan, H., Shamir, R.: On the complexity of DNA physical mapping. Advances in Applied Mathematics 15, 251–261 (1994)
16. Ideker, T., Karp, R.M., Scott, J., Sharan, R.: Efficient algorithms for detecting signaling pathways in protein interaction networks. Journal of Computational Biology 13(2), 133–144 (2006)
17. Kelley, B.P., Sharan, R., Karp, R.M., Sittler, T., Root, D.E., Stockwell, B.R., Ideker, T.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proceedings of the National Academy of Sciences of the United States of America 100(20), 11394–11399 (2003)
18. Lacroix, V., Fernandes, C.G., Sagot, M.-F.: Motif search in graphs: Application to metabolic networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics 3(4), 360–368 (2006)
19. McMorris, F.R., Warnow, T.J., Wimer, T.: Triangulating vertex-colored graphs. SIAM Journal on Discrete Mathematics 7(2), 296–306 (1994)
20. Moran, S., Snir, S.: Convex recolorings of strings and trees: Definitions, hardness results and algorithms. In: Dehne, F., López-Ortiz, A., Sack, J.-R. (eds.) WADS 2005. LNCS, vol. 3608, pp. 218–232. Springer, Heidelberg (2005)
21. Naor, J., Naor, M.: Small-bias probability spaces: Efficient constructions and applications. In: Proceedings of the 25th annual ACM Symposium on Theory Of Computing (STOC), pp. 213–223. ACM Press, New York (1990)
22. Robertson, N., Seymour, P.D.: Graph minors. II. Algorithmic aspects of tree-width. SIAM Journal of Algorithms 7, 309–322 (1986)