Parameterized Algorithms and Hardness Results for Some Graph Motif Problems

Nadja Betzler[1,⋆], Michael R. Fellows[2,⋆⋆], Christian Komusiewicz[1,⋆⋆⋆], and Rolf Niedermeier[1]

[1] Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{betzler,ckomus,niedermr}@minet.uni-jena.de
[2] PC Research Unit, Office of DVC (Research), University of Newcastle, Callaghan, NSW 2308, Australia
Michael.Fellows@newcastle.edu.au

Abstract. We study the NP-complete Graph Motif problem: given a vertex-colored graph $G = (V, E)$ and a multiset M of colors, does there exist an $S \subseteq V$ such that $G[S]$ is connected and carries exactly (also with respect to multiplicity) the colors in M? We present an improved randomized algorithm for Graph Motif with running time $O(4.32^{|M|} \cdot |M|^2 \cdot |E|)$. We extend our algorithm to list-colored graph vertices and the case where the motif $G[S]$ needs not be connected. By way of contrast, we show that extending the request for motif connectedness to the somewhat "more robust" motif demands of biconnectedness or bridge-connectedness leads to W[1]-complete problems. Actually, we show that the presumably simpler problems of finding (uncolored) biconnected or bridge-connected subgraphs are W[1]-complete with respect to the subgraph size. Answering an open question from the literature, we further show that the parameter "number of connected motif components" leads to W[1]-hardness even when restricted to graphs that are paths.

1 Introduction

With the advent of network biology [1, 15] and complex network analysis in general, the study of pattern matching problems in graphs has become more and more important. In this context, the term "graph motif" plays a central role. Roughly speaking, there are two views of graph (or network) motifs. The older is the topological view where one basically ends up with certain subgraph isomorphism problems. For instance, the term "network motif" has been used to represent patterns of interconnections that occur in a network at frequencies much higher than those found in random networks [16, 18]. By way of contrast,

⋆ Supported by the DFG, project DARE, GU 1023/1.
⋆⋆ Supported by the Australian Research Council. Work done while staying in Jena as a recipient of the Humboldt Research Award of the Alexander von Humboldt Foundation, Bonn, Germany.
⋆⋆⋆ Supported by a PhD fellowship of the Carl-Zeiss-Stiftung.

the second and more recent view on graph motifs takes a more "functional approach". Here, topology is of lesser importance but the functionalities of network nodes (expressed by colors) form the governing principle. This approach has been propagated by Lacroix et al. [12] and has been followed up by Fellows et al. [9], defining the following problem.

Graph Motif: Input: A vertex-colored undirected graph $G = (V, E)$ and a multiset of colors M, with $|M| = k$. Question: Does there exist an $S \subseteq V$ such that the induced subgraph $G[S]$ is connected and there is a bijection between the colors of the vertices in S and M?

The different vertex colors are used to model different functionalities. Although originally introduced in a biological context [9, 12], it is conceivable that Graph Motif is an interesting problem not only for biological networks, but also may prove useful when studying complex social or technical networks.

Known Results. Not surprisingly, Graph Motif is a computationally hard problem. It is NP-complete even if the input multiset M actually is a set and the input graph is a tree with maximum vertex degree three [9]. Moreover, NP-completeness has also been shown for the case that M consists of only two colors and the input graph is restricted to be bipartite with maximum degree four [9]. Given the apparent hardness of Graph Motif, Fellows et al. [9] initiated a parameterized complexity analysis. Unfortunately, it turned out that Graph Motif is W[1]-hard for trees when parameterized by the number of different colors in the motif multiset M. That is, there is no hope to confine the seemingly inevitable combinatorial explosion to the number of colors. By way of contrast, there are good news for other parameterizations. First, when parameterized by the motif size $k := |M|$, Graph Motif can be solved by a color-coding algorithm running in $O(87^k \cdot k \cdot n^2)$ time on an n-vertex graph, proving its fixed-parameter tractability with respect to the motif size [9][3]. Finally, Dondi et al. [6] extended these investigations for Graph Motif by studying the case where the subgraph induced by the chosen motif vertices needs not be connected.

New Results. Our work makes two sorts of contributions. First, we present significantly faster algorithms for Graph Motif and two natural variants, now giving hope for practically useful implementations. In all these cases, the motif size is the governing parameter. Second, we further chart the range of tractability of Graph Motif by exploring natural variants that become W[1]-hard (again with respect to the parameter motif size). More specifically, we achieve the following results. On the positive side, we improve the randomized algorithm of Fellows et al. [9] running in $O(87^k \cdot k \cdot n^2)$ time and consuming $O(4^k \cdot n)$ space to a new randomized algorithm running in $O(4.32^k \cdot k^2 \cdot m)$ time and consuming $O(2.47^k \cdot n)$ space on an m-edge graph. Note that both algorithms are based on the color-coding technique due to Alon et al. [2], which has recently proven practical usefulness [5, 7, 10, 14]. Both algorithms can be derandomized, but the

[3] Fellows et al. [9] do not explicitly state the running time of the randomized version of their algorithm. Instead, they demonstrate a running time of $O(2^{5k} \cdot k \cdot n^2)$ per trial. Using k colors for color-coding, $O(e^k)$ trials are needed to achieve a sufficiently low error probability, which results in a total running time of $O(87^k \cdot k \cdot n^2)$.

current state of the art of derandomization techniques seems prohibitive from a practical point of view (also see [10]). We extend our fixed-parameter tractability results for Graph Motif to two variants: List-colored Graph Motif, where each chosen vertex may allow for a list of colors that it can match, and Min-CC Graph Motif, where we specify the number of connected components the graph motif may have. On the negative side, we also provide several parameterized hardness results. First, we investigate the search for somewhat "more robust" motifs. In other words, we show that if one requires that the found motif shall not only be connected but biconnected or bridge-connected, then in both cases the corresponding Graph Motif problem becomes W[1]-complete with respect to the parameter motif size (actually, even special cases thereof do so). Since these are the two most simple demands one may pose for more robust motifs, this shows that the request for connected motifs is already a topology demand close to the border of tractability and intractability.[4] Finally, somewhat aside, we answer an open question of Dondi et al. [6] by proving that the aforementioned Min-CC Graph Motif problem is W[1]-hard with respect to the number of components even if the input graph is restricted to be only a path. Due to the lack of space, some details are deferred to the full version.

Preliminaries. We consider only simple undirected graphs $G = (V, E)$, where $n := |V|$ and $m := |E|$ throughout the whole work. For a vertex $v \in V$, let $N(v) := \{u \mid \{u, v\} \in E\}$ denote the *open neighborhood* of v, and let $N[v] := N(v) \cup \{v\}$ denote the *closed neighborhood* of v. A *coloring* of an undirected graph $G = (V, E)$ is a function $c : V \to C$, where C is a set of colors. Unless stated otherwise, a *motif* is a multi-set of colors. In case that the motif is a set, we call the motif *colorful*. An *occurrence* of a motif M in G is a set of vertices $S \subseteq V$ such that $|S| = |M|$, $G[S]$ is connected, and there are x vertices of color c in S iff M contains c exactly x times. Let $\mathrm{col}(v)$ denote the color of a vertex v and $\mathrm{col}(S)$ the multiset of colors of the vertices of S. A vertex u in an undirected graph is called a *cut vertex* if there are two vertices v, w with $v \neq u$ and $w \neq u$ such that every path from v to w contains u. If an undirected graph G is connected and has no cut-vertex, then G is *biconnected*. In general, if a graph $G = (V, E)$ cannot be disconnected by deletion of any set of $p - 1$ vertices, it is called *p-connected*. A graph is called *p-edge-connected* if it cannot be disconnected by deletion of any set of $p - 1$ edges. A 2-edge-connected graph is called *bridge-connected*.

The *color-coding* technique yields randomized fixed-parameter algorithms [2]. The main idea is to randomly color the vertices of the graph, and then to solve the corresponding problem under the assumption that the subgraph that is searched for obtains a *colorful* coloring, that is, all of the vertices of the subgraph have pairwise different colors. This assumption often leads to a problem solvable more efficiently. The procedure of coloring and then solving the subsequent problem on the colored graph is repeated as often as necessary to obtain a sufficiently

[4] Our results also generalize to higher connectivity demands. Even further, they hold for uncolored graphs, where one searches for a subgraph with the specific connectivity demand, and the parameter is the number of subgraph vertices.

low error probability. We say that a randomized algorithm solves a problem with *error probability* ϵ if the probability that it fails to return the correct answer is at most ϵ.

Parameterized algorithmics aims at a multivariate complexity analysis of problems [8, 13]. The hope lies in accepting the seemingly inevitable combinatorial explosion for NP-hard problems, but to confine it to a parameter k. A given parameterized problem (I, k) is *fixed-parameter tractable (FPT)* with respect to the parameter k if it can be solved within running time $f(k) \cdot \text{poly}(|I|)$ for some computable function f. Downey and Fellows [8] developed a theory of parameterized intractability by means of devising a completeness program with complexity classes. The first level of (presumable) parameterized intractability is captured by the complexity class W[1]. A *parameterized reduction* reduces a problem instance (I, k) in $f(k) \cdot \text{poly}(|I|)$ time to an instance (I', k') such that (I, k) is a yes-instance if and only if (I', k') is a yes-instance and k' only depends on k but not on $|I|$. If for a given parameterized problem L there is a parameterized problem L' such that L' is W[1]-hard and there is a parameterized reduction from L' to L, then L is also W[1]-hard.

2 Fixed-Parameter Algorithms

Our accelerated algorithm for Graph Motif, as the previous one [9], is based on the color-coding technique [2]. However, we make use of the following new observation on colorful motifs.

Lemma 1. *Let (G, M) be a* Graph Motif *instance such that M is colorful. Then,* Graph Motif *can be solved in $O(3^k \cdot m)$ time.*

Proof. We describe a dynamic programming algorithm that finds an occurrence of M. In the dynamic programming table, entry $D_{v,C}$ stores the "minimum score" of a color set C for a vertex v, where a score of 0 means that an occurrence of C that includes v exists. We initialize the entries of the dynamic programming table with

$$D_{v,C} = \begin{cases} 0, & C = \{\text{col}(v)\}, \\ 1, & \text{otherwise.} \end{cases}$$

In the recurrence, we look for the combination of subsets of a color set such that the sum of the entries is minimum:

$$D_{v,C} = \min_{u \in N(v),\, C' \subset C} \left\{ \begin{array}{l} D_{u,C \setminus \{\text{col}(v)\}}, \\ D_{v,C' \cup \{\text{col}(v)\}} + D_{v,(C \setminus C') \cup \{\text{col}(v)\}} \end{array} \right\}.$$

Since the motif M is colorful, we can restrict attention to joining sets of vertices that have disjoint color sets. Therefore, we never join vertex sets that have vertices in common. If a colorful motif M occurs in G, then for some $v \in V$, $D_{v,M} = 0$. Furthermore, during the dynamic programming procedure we only need to consider color sets C that are subsets of M. Therefore, we have $O(2^k)$ entries

per vertex, which results in a table size of $O(2^k \cdot n)$. Overall, the first part of the recursion can be executed in $O(2^k \cdot m)$ time, since for each color set $C \subseteq M$ and for every vertex v we have to scan once through the adjacency list of v and for each neighbor the corresponding table entry can be found in constant time. The second part of the recursion can be executed in $O(3^k \cdot n)$ time overall: for each vertex v the number of combinations that have to be considered is bounded by $O(3^k)$, since we have to consider all possible subsets of M and for each subset we have to consider all possibilities to split this subset. Overall this amounts to $O(3^k)$ combinations, since there are 3^k possibilities to split a subset of size k into three disjoint subsets (in our case these subsets are $M \setminus C$, C' , and C''). For each combination the computation of the recursion can be performed in constant time. Overall, the running time amounts to $O(3^k \cdot m)$. An occurrence of the motif can be computed by traceback within the same asymptotic running time bound. □

The above dynamic programming procedure is basically a simplified version of the procedure for the related problem of finding a minimum-weight tree of size k [14]. The main difference is that for Graph Motif, we do not have additional weights that are associated with the graph vertices.

We now show how to use Lemma 1 in order to obtain an algorithm in case that the motif is a multiset of colors. The main idea is to use the technique of color-coding [2] in order to transform any instance that has a multiset of colors as motif into an instance that has a colorful motif. To this latter instance then Lemma 1 applies. In the following, we describe this transformation in detail. Let M be the motif and let $\mathrm{occ}(c)$ denote the number of occurrences of a color c in M. For each color c with $\mathrm{occ}(c) \geq 2$ we introduce $\mathrm{occ}(c)$ new colors $c_1, c_2, \ldots, c_{\mathrm{occ}(c)}$. Then, we randomly recolor each vertex that has color c with one of the new colors, where the probability for each color is exactly $1/\mathit{occ}(c)$ (uniform distribution). Let M' be the set of colors that contains the colors that occurred only once in M together with the colors $\{c_1, c_2, \ldots, c_{\mathrm{occ}(c)}\}$ for every color c with $\mathrm{occ}(c) \geq 2$. Furthermore, let S be an occurrence of M. We say that S achieves a *colorful recoloring* if $\mathrm{col}(S)$ is colorful after the recoloring procedure. Clearly, if S achieves a colorful recoloring, then $\mathrm{col}(S) = M'$. An occurrence of M' can be found via dynamic programming by Lemma 1. This procedure of recoloring with subsequent dynamic programming is repeated until either an occurrence of M is found, or the probability that there is an S that has not achieved a colorful recoloring is acceptably low.

Proposition 1. Graph Motif *can be solved with error probability ϵ within $O(|\ln(\epsilon)| \cdot 8.16^k \cdot m)$ time.*

Proof. By Lemma 1, we can find an occurrence of a colorful motif in $O(3^k \cdot m)$ time. Therefore, the total running time of the algorithm is $O(t(\epsilon) \cdot 3^k \cdot m)$, where $t(\epsilon)$ denotes the number of trials that is needed in order to achieve a colorful recoloring of the vertices of the motif in at least one of the trials with a probability of at least $1 - \epsilon$. For each color $c \in M$, the probability P_c that the $\mathrm{occ}(c)$ vertices in S that have color c receive a colorful

recoloring is $(\mathrm{occ}(c))!/\mathrm{occ}(c)^{\mathrm{occ}(c)}$, because each coloring has the same probability and $(\mathrm{occ}(c))!$ colorings of the $\mathrm{occ}(c)^{\mathrm{occ}(c)}$ possible colorings are colorful. Using Stirling's approximation for factorials we can show that $\mathrm{occ}(c)!/\mathrm{occ}(c)^{\mathrm{occ}(c)} > \sqrt{2 \cdot \pi \cdot \mathrm{occ}(c)} \cdot e^{-\mathrm{occ}(c)}$. For two colors c_1 and c_2 the probabilities P_{c_1} and P_{c_2} are independent. Therefore, the probability $P_{c_1 \wedge c_2}$ that the vertices of both color classes achieve a colorful recoloring is

$$P_{c_1 \wedge c_2} = P_{c_1} \cdot P_{c_2} > \sqrt{2 \cdot \pi \cdot (\mathrm{occ}(c_1) + \mathrm{occ}(c_2))} \cdot e^{-(\mathrm{occ}(c_1)+\mathrm{occ}(c_2))}.$$

The probability P_M that an occurrence of M receives a colorful recoloring thus is

$$P_M = \prod_{c \in \mathrm{col}(M)} P_c > \sqrt{2 \cdot \pi \cdot k} \cdot e^{-\sum_{c \in \mathrm{col}(M)} \mathrm{occ}(c)} > e^{-k}.$$

After t trials the error probability, that is, the probability that a colorful recoloring was not achieved, is $(1-P_M)^t$. Therefore, the number of trials $t(\epsilon)$ to achieve an error probability of at most ϵ is $t(\epsilon) = \lceil |\ln(\epsilon)|/\ln(1-P_M) \rceil = |\ln(\epsilon)| \cdot O(e^k)$. Hence, the total running time of the algorithm when an error probability of at most ϵ is allowed is $O(|\ln(\epsilon)| \cdot e^k \cdot 3^k \cdot m) = O(|\ln(\epsilon)| \cdot 8.16^k \cdot m)$. □

Applying two speed-up techniques, we can further improve the running time of the algorithm. First, as proposed by Hüffner et al. [10], we can increase the number of colors that are used for color-coding in order to increase the probability of an occurrence of M to receive a colorful recoloring[5]. Second, we can speed up the dynamic programming procedure of Lemma 1 by using the technique of *fast subset convolution*. This novel technique was developed by Björklund et al. [3], who used it to speed up several dynamic programming algorithms including the algorithm by Scott et al. [14] for computing minimum weight size k trees in signalling networks.

Let f and g be functions defined on the power set of a finite set N with $|N| = n$, that is, $f, g : \mathcal{P}(N) \to I$. For any ring over I that defines addition and multiplication on elements of I, the *subset convolution* of f and g, denoted by $f * g$, is defined for each $S \subseteq N$ as

$$f * g : \mathcal{P}(N) \to I, \quad (f * g)(S) = \sum_{T \subseteq S} f(T) g(S \setminus T).$$

To calculate the subset convolution means to determine the value of $f * g$ for all 2^n possible inputs, assuming that f and g can be evaluated in constant time (typically by being stored in a table). A naive algorithm that calculates each value independently needs $O(\sum_{i=0}^{n} \binom{n}{i} 2^i) = O(3^n)$ ring operations. The following result shows a substantial improvement.

Theorem 1 (Björklund et al. [3]). *The subset convolution over an arbitrary ring can be computed with $O(2^n \cdot n^2)$ ring operations.*

[5] Increasing the number of colors has been independently examined by Deshpande et al. [5]. Hüffner et al. [10] derive a better bound on the worst-case running time.

Björklund et al. [3] showed how to apply Theorem 1 to also calculate the subset convolution for the integer min-sum semiring

$$f * g : \mathcal{P}(N) \to \mathbb{Z}, \quad (f * g)(S) = \min_{T \subseteq S} f(T) + g(S \setminus T)$$

by embedding it into the standard integer sum-product ring.[6] Recall the recurrence of the dynamic programming procedure for colorful motifs:

$$D_{v,C} = \min_{u \in N(v),\, C' \subset C} \left\{ \begin{array}{l} D_{u,C \setminus \{\mathrm{col}(v)\}}, \\ D_{v,C' \cup \{\mathrm{col}(v)\}} + D_{v,(C \setminus C') \cup \{\mathrm{col}(v)\}} \end{array} \right\}.$$

The first part of the recurrence can be evaluated in $O(2^k \cdot m)$ time. For the second part we can use fast subset convolution and can thus compute the recurrence in $O(2^k \cdot k^2 \cdot n)$ time, because each ring operation can be performed in constant time, since the maximum weight that is used for the basic table entries is 1. Clearly, the graph G has an occurrence of M if there is a table entry $D_{v,M} = 0$ in the final table. The actual occurrence of the motif can be computed in $O(2^k \cdot k \cdot m)$ time by traceback. In the following theorem, we upper-bound the running time of the algorithm that is obtained from combining the two described speed-up techniques.

Theorem 2. Graph Motif *can be solved with error probability* ϵ *in* $O(|\ln(\epsilon)| \cdot 4.32^k \cdot k^2 \cdot m)$ *time.*

Proof. Hüffner et al. [10] showed that when using $1.3 \cdot k$ colors, the number of trials that is needed to obtain error probability ϵ is $O(|\ln(\epsilon)| \cdot 1.752^k)$. However, this increases the running time of the dynamic programming procedure, since now the color set has size $1.3 \cdot k$. The modified dynamic programming procedure then has a running time of $O(2^{1.3 \cdot k} \cdot k^2 \cdot m)$. Overall, the running time amounts to $O(|\ln(\epsilon)| \cdot 1.752^k \cdot 2^{1.3 \cdot k} \cdot k^2 \cdot m) = O(4.32^k \cdot k^2 \cdot m)$. □

A drawback of using $1.3 \cdot k$ colors is that the memory requirement increases from $O(2^k \cdot m)$ to $O(2^{1.3k} \cdot m) = O(2.47^k \cdot m)$. However, it was shown that the running time improvement is enormous in practice [10]. In some special cases, we need even less trials to achieve an exponentially low error probability. For example, if every color in the motif occurs at most twice, then we have to use at most two colors per vertex. Furthermore, there can be at most $k/2$ colors that appear twice in the motif. Using two colors for each color c that appears twice in M, the two vertices in an occurrence of M that have a color c receive different colors with probability $1/2$. Hence, the probability that a recoloring is a colorful recoloring is $2^{-k/2} = (\sqrt{2})^{-k}$. The number of trials needed to achieve exponentially low error probability then is $O((\sqrt{2})^k)$ and the total running time $O((\sqrt{2})^k \cdot 2^k \cdot k^2 \cdot m) = O(2.83^k \cdot k^2 \cdot m)$.

[6] Björklund et al. [3] also considered the variant where we do not have disjoint sets T and $S \setminus T$ but allow one element occurring in both sets (as we make use of in the following).

Two Natural Graph Motif Variants. We extend our randomized algorithm for the basic Graph Motif problem to two practically interesting problem variants. The original formulation of Graph Motif allows multiple colors per vertex [12]. This makes sense in a biological context in order to model multiple functionalities of one element. The input graph can then be formalized as a *list-colored graph*, in which a *list* of colors is attached to every vertex of the graph. In other words, for a vertex $v \in V$ of a list-colored graph, $\mathrm{col}(v)$ denotes a set of colors instead of a single color.

List-colored Graph Motif: Input: A list-colored undirected graph $G = (V, E)$ and a multiset of colors M. Question: Does there exist a vertex subset $S \subseteq V$ such that the induced subgraph $G[S]$ is connected and there is a bijection $f : S \to M$ such that $\forall v \in S : f(v) \in col(v)$?

Unfortunately, we cannot use our above algorithm for List-colored Graph Motif. The difficulty is that in list-colored graphs we do not have a one-to-one correspondence between vertices and colors; hence, two disjoint color sets do not imply two disjoint vertex sets. However, we can apply a different color-coding procedure, partially resembling the algorithm by Fellows et al. [9].

Theorem 3. List-colored Graph Motif *can be solved with error probability* ϵ *in* $O(|\ln(\epsilon)| \cdot 10.88^k \cdot m)$ *time.*

Proof. We use color-coding. To avoid ambiguities, we call the random colors assigned by the color-coding procedure *labels*, and the term color only refers to the colors of the list-colored graph. Let $L = \{l_1, l_2, \ldots, l_k\}$ denote a set of k distinct labels. We randomly assign (uniformly distributed) the labels of L to the vertices of the graph and solve the problem of finding an occurrence of the motif M under the assumption that all vertices of the occurrence have received a different label. Without loss of generality, assume that M is colorful. Otherwise, we transform M and G as follows: For each color c that occurs $\mathrm{occ}(c)$ times, we add $\mathrm{occ}(c)$ new colors to M and completely remove c from G. Furthermore, for every vertex v in G with $c \in \mathrm{col}(v)$, we remove c from $\mathrm{col}(v)$ and add the $\mathrm{occ}(c)$ new colors to $\mathrm{col}(v)$. Let M' and G' be the thus modified motif and graph, respectively. We now solve the problem of finding an occurrence of M' in G'. Each such occurrence clearly corresponds to an occurrence of M in G.

The problem of finding a colorful occurrence of M that has the labels of L is solved by dynamic programming. First, we extend our notion of occurrence. Let $F \subseteq (L \cup M)$ be a set that contains labels as well as colors. An occurrence of F is defined as a set of vertices S such that the vertices of S have exactly the labels of $F \cap L$, and there is a bijection $f : S \to F \cap M$, such that for each vertex v $f(v) \in \mathrm{col}(v)$. An entry $D_{v,F}$ of the dynamic programming table denotes the "score" of an occurrence of F that contains v. We initialize the table as follows:

$$D_{v,\{c,l\}} = \begin{cases} 0, & c \in \mathrm{col}(v) \wedge l = \mathrm{label}(v), \\ 1, & \text{otherwise.} \end{cases}$$

Furthermore, we assign weight 1 to all entries $D_{v,\{c\}}$ and $D_{\{l\}}$. The recurrence reads

$$D_{v,F} = \min_{u \in N(v),\ c \in \mathrm{col}(v),\ F' \subset F} \left\{ \begin{array}{l} D_{u,F\setminus\{c,\mathrm{label}(v)\}}, \\ D_{v,F'\cup\{c,\mathrm{label}(v)\}} + D_{v,(F\setminus F')\cup\{c,\mathrm{label}(v)\}} \end{array} \right\}.$$

We calculate the score for sets $F \subseteq M$ of increasing cardinality. Note that by initializing the entries $D_{v,\{c\}}$ and $D_{v,\{l\}}$ with 1, we make sure that a score of 0 of an "occurrence" of F can only be achieved when there is a one-to-one correspondence between labels and colors of the occurrence. Therefore, if there is a $v \in V$ such that $D_{v,L\cup M} = 0$, then there is an occurrence of $L \cup M$ in G. An actual occurrence then can be computed by traceback.

For the running time consider the following. Clearly $|L \cup M| = 2 \cdot k$. The recursion is similar to the recursion in the proof of Theorem 2. Hence, we can also apply subset convolution and obtain a running time of $O(2^{2\cdot k} \cdot (2 \cdot k)^2 \cdot m) = O(4^k \cdot k^2 \cdot m)$ for the dynamic programming procedure. The number of trials that is needed to obtain a *good labelling* with probability at least $1-\epsilon$ is $O(|\ln(\epsilon)| \cdot e^k)$. The total running time thus amounts to $O(|\ln(\epsilon)| \cdot e^k \cdot 4^k \cdot k^2 \cdot m) = O(10.88^k \cdot k^2 \cdot m)$. □

Our second variant of Graph Motif has been introduced by Dondi et al. [6], who proposed a generalization of Graph Motif in which it is no longer demanded that the motif is connected.

Min-CC Graph Motif: Input: A vertex-colored undirected graph $G = (V, E)$, a multiset of colors M with $|M| = k$, and a nonnegative integer d. Question: Does there exist an $S \subseteq V$ such that $G[S]$ has at most d components, and there is a bijection between the colors of the vertices in S and M?

Clearly, Graph Motif is Min-CC Graph Motif with $d = 1$. Among other results, Dondi et al. [6] showed that the algorithms for Graph Motif by Fellows et al. [9] can be adapted to solve Min-CC Graph Motif. We can also modify our Graph Motif algorithm to solve Min-CC Graph Motif.

Theorem 4. Min-CC Graph Motif *can be solved with error probability* ϵ *in* $O(|\ln(\epsilon)| \cdot 4.32^k \cdot k^2 \cdot m)$ *time.*

3 Parameterized Hardness Results

Lacroix et al. [12] motivated the study of (variants) of the Graph Motif problem by considerations comparing "topological motifs" with "functional motifs". The Graph Motif problem only poses a minimal demand on the motif topology by requiring connectedness. The natural question arises what happens if we ask for somewhat "more robust" motifs, replacing the connectedness demand by demands for biconnectivity, bridge-connectivity and the like. As we will show in this section, these seemingly small steps towards topologically more constrained motifs already lead to W[1]-completeness. Finally, the only time considering a parameter other than motif size, we answer an open question of Dondi et al. [6]

Fig. 1: An example of the transformation of a Clique instance with $k = 3$ into a Biconnected Subgraph instance with $k' = 15$. White vertices in G' belong to V_1, black vertices to V_2.

by showing that the parameter "number of connected components" in a graph motif leads to a W[1]-hard problem.

Biconnected Graph Motif: Input: A vertex-colored undirected graph $G = (V, E)$ and a multiset of colors M. Question: Does there exist an $S \subseteq V$ such that the induced subgraph $G[S]$ is biconnected and there is a bijection between the colors of the vertices in S and M?

We will show that Biconnected Graph Motif is W[1]-complete when parameterized by the size of the motif M. In fact, we prove an even stronger result. Consider the special case that M contains only one color c, $|M| = k$, and that all vertices in G have color c. Then, the remaining problem to find a biconnected subgraph of size *exactly* k is denoted as:

Biconnected Subgraph: Input: An undirected graph $G = (V, E)$ and a nonnegative integer k. Question: Does there exist an $S \subseteq V$ of size k such that the induced subgraph $G[S]$ is biconnected?

Note that looking for a biconnected subgraph of size *at least* k is solvable in linear time [17]. However, restricting the size of the biconnected subgraph to exactly k makes the problem surprisingly hard. We prove the parameterized hardness by reduction from the Clique problem, which is known to be W[1]-complete [8] with respect to the size of the clique searched for.

Clique: Input: An undirected graph G and a nonnegative integer k. Question: Is there a complete subgraph of size k in G?

Theorem 5. Biconnected Subgraph *is W[1]-complete with respect to* k.

Proof. To show the W[1]-hardness, we give a parameterized reduction from Clique to Biconnected Subgraph. Let (G, k) be a Clique instance. We construct a graph G' from G by replacing every edge e of G with a simple path p_e that has $\binom{k}{2} + 1$ internal new vertices. The vertex set of G' can be partitioned into two vertex sets V_1 and V_2, where V_1 contains the vertices that correspond to vertices of the original graph G and V_2 contains the new internal path vertices. An example of this reduction is shown in Figure 1.

We prove in the following that G has a clique of size k iff G' has a biconnected subgraph of size $k' = k + \binom{k}{2} \cdot (\binom{k}{2} + 1)$. If G has a clique C of size k, then the subgraph that is induced by the k vertices of C and by the vertices on the $\binom{k}{2}$ paths that were created from the $\binom{k}{2}$ clique edges of C in G has size exactly $k + \binom{k}{2} \cdot (\binom{k}{2} + 1)$. Clearly, this subgraph is also biconnected.

It remains to show that if G' has a biconnected subgraph of size $k' = k + \binom{k}{2} \cdot (\binom{k}{2}+1)$, then G has a clique of size k. Let G' have a biconnected subgraph $G'[S]$ of size k. If S contains one vertex of a path p_e, then it must contain all vertices from p_e, because otherwise $G'[S]$ would not be biconnected. Hence, the number of vertices k' in S can be expressed as $k' = a + b \cdot (\binom{k}{2} + 1)$, where $a = |S \cap V_1|$ and b denotes the number of paths in G' that correspond to edges of G.

We distinguish two main cases. In the first case, let $a = k$. Then, $G'[S]$ must contain exactly $\binom{k}{2}$ paths that correspond to edges in G. Let $e = \{u, v\}$ be an edge of G, and let $A := S \cap V_1$. Since $G'[S]$ is biconnected, if a path p_e is contained in S, then $\{u, v\} \subseteq A$. Since $G'[S]$ contains exactly $\binom{k}{2}$ paths consisting of vertices from V_2 and each path must connect two vertices of A, all vertices of A are pairwise connected via a path of length $\binom{k}{2}$. Hence, the subgraph $G[A]$ must be a size-k clique since it contains exactly k vertices and exactly $\binom{k}{2}$ edges.

We now consider the case $a \neq k$ and show that in this case, either $a + b \cdot \left(\binom{k}{2} + 1\right) \neq k + \binom{k}{2} \cdot \left(\binom{k}{2} + 1\right) = k'$ or $G'[S]$ cannot be biconnected. Clearly, if $b = \binom{k}{2}$, then

$$a + b \cdot \left(\binom{k}{2} + 1\right) = a + \binom{k}{2} \cdot \left(\binom{k}{2} + 1\right) \neq k + \binom{k}{2} \cdot \left(\binom{k}{2} + 1\right) = k'.$$

Therefore, we can assume that $b \neq \binom{k}{2}$. In the following, we list all remaining cases and show that either $a + b \cdot \binom{k}{2} \neq k'$ or $G'[S]$ is not biconnected.
Case 1: $b > \binom{k}{2}$.

$$a + b \cdot \left(\binom{k}{2} + 1\right) \geq a + \left(\binom{k}{2} + 1\right) \cdot \left(\binom{k}{2} + 1\right) > k + \binom{k}{2} \cdot \left(\binom{k}{2} + 1\right)$$

Case 2.1 : $b < \binom{k}{2}$ and $a < \binom{k}{2}$.

$$a + b \cdot \left(\binom{k}{2} + 1\right) < \binom{k}{2} + \left(\binom{k}{2} - 1\right) \cdot \left(\binom{k}{2} + 1\right) < k + \binom{k}{2} \cdot \left(\binom{k}{2} + 1\right)$$

Case 2.2 : $b < \binom{k}{2}$ and $a \geq \binom{k}{2}$.
In this case, $G'[S]$ cannot be biconnected: S consists of at least $a \geq \binom{k}{2}$ vertices from V_1 and less than $\binom{k}{2}$ paths that correspond to edges of G. Therefore, at least one of the $\binom{k}{2}$ vertices from V_1 is connected to at most one path. By construction, vertices in V_1 may only be adjacent to vertices in V_2. Hence, $G'[S]$ is not biconnected.

Summarizing, G has a clique of size k iff G' has a biconnected subgraph of size $k \cdot \binom{k}{2} \cdot (\binom{k}{2} + 1)$. The reduction can be clearly performed in polynomial time. We omit the proof for containment in W[1]. □

A second natural way to heighten the robustness demands for Graph Motif is to search for bridge-connected motifs. We define Bridge-Connected Subgraph in complete analogy to Biconnected-Connected Subgraph, simply replacing the demand for biconnectivity by the demand for bridge-connectivity.

The reduction from Clique as used in the proof of Theorem 5 works also for bridge-connected subgraphs.

Theorem 6. Bridge-Connected Subgraph *is W[1]-complete with respect to k (number of subgraph vertices).*

Further, we can generalize the hardness results to graph motifs of higher connectivity. To this end, consider the following problem.

p-(Edge) Connected Subgraph: Input: An undirected graph G and a nonnegative integer k. Question: Does there exist an $S \subseteq V$ of size k such that the induced subgraph $G[S]$ is p-(edge) connected?

Theorem 7. p-(Edge) Connected Subgraph *is W[1]-complete with respect to k (number of subgraph vertices).*

The following theorem answers an open question of Dondi et al. [6].

Theorem 8. Min-CC Graph Motif *restricted to paths is W[1]-hard with respect to the parameter "number of components".*

Proof. (Construction) We reduce from the W[1]-complete Perfect Code [4] problem: Given an undirected graph $G = (V, E)$ and a positive integer k, is there is a size-k-subset $V' \subseteq V$ such that for every vertex $v \in V$ there is exactly one vertex in $N[v] \cap V'$. Given a Perfect Code instance $(G = (V, E), k)$, we construct a Min-CC Graph Motif instance consisting of a path P and a motif M. It asks for the existence of a solution consisting of k connected components. The vertex set of P consists of $|N[v]|$ vertices with color c_v for every $v \in V$, $n-1$ "separator" vertices with color s each, and $2 \cdot n$ "end" vertices with color e each. Now, we describe the order of the vertices in the path P. For this, let a "subpath" of P denote a connected path that is part of P. Then, for every vertex $v \in V$ there is a subpath containing $|N[v]|$ vertices colored by $\{c_w \mid w \in N[v]\}$ in an arbitrary order. At both ends of every subpath we add an end vertex with color e. Finally, we connect all subpaths in an arbitrary order such that two neighboring subpaths are connected through a separator vertex with color s. The motif set M consists of $2 \cdot k$ times the color e and $\{c_v \mid v \in V\}$.

We omit to show that G has a perfect code of size k iff the there are k subpaths $P_1, \ldots, P_k$ such that there is a bijection between the colors of their vertices and the colors of M. □

4 Conclusion

Graph Motif and its variants are natural graph-theoretic pattern matching problems with prospective applications. Our positive algorithmic results should support implementation and experimental work, similarly to previous positive experiences with color-coding based graph algorithms [5, 7, 10, 11, 14]. It is particularly interesting whether the recently introduced subset convolution technique [3], which so far has been studied purely from a theoretical point of view, also yields a significant speed-up in practice.

Acknowledgments. We are grateful to Jiong Guo (hinting to Theorem 8) and Frances Rosamond for helpful comments.

References

[1] E. Alm and A. P. Arkin. Biological networks. *Curr. Opin. Struc. Biol.*, 13(2): 193–202, 2003.

[2] N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. ACM*, 42(4):844–856, 1995.

[3] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Fourier meets Möbius: fast subset convolution. In *Proc. 39th STOC*, pages 67–74. ACM, 2007.

[4] M. Cesati. Perfect code is W[1]-complete. *Inform. Process. Lett.*, 81:163–168, 2002.

[5] P. Deshpande, R. Barzilay, and D. R. Karger. Randomized decoding for selection-and-ordering problems. In *Proc. NAACL HLT '07*, pages 444–451. Association for Computational Linguistics, 2007.

[6] R. Dondi, G. Fertin, and S. Vialette. Weak pattern matching in colored graphs: Minimizing the number of connected components. In *Proc. 10th ICTCS*, volume 4596 of *WSPC*, pages 27–38. World Scientific, 2007.

[7] B. Dost, T. Shlomi, N. Gupta, E. Ruppin, V. Bafna, and R. Sharan. QNet: A tool for querying protein interaction networks. In *Proc. 11th RECOMB*, volume 4453 of *LNCS*, pages 1–15. Springer, 2007.

[8] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[9] M. R. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In *Proc. 34th ICALP*, volume 4596 of *LNCS*, pages 340–351. Springer, 2007.

[10] F. Hüffner, S. Wernicke, and T. Zichner. Algorithm engineering for color-coding to facilitate signaling pathway detection. In *Proc. 5th APBC*, volume 5 of *Advances in Bioinf. and Comput. Biol.*, pages 277–286. Imperial College Press, 2007. Extended version to appear in *Algorithmica*.

[11] F. Hüffner, S. Wernicke, and T. Zichner. FASPAD: fast signaling pathway detection. *Bioinformatics*, 23(13):1708–1709, 2007.

[12] V. Lacroix, C. G. Fernandes, and M.-F. Sagot. Reaction motifs in metabolic networks. In *Proc. 5th WABI*, volume 3692 of *LNCS*, pages 178–191. Springer, 2005.

[13] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[14] J. Scott, T. Ideker, R. M. Karp, and R. Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.*, 13(2):133–144, 2006.

[15] R. Sharan and T. Ideker. Modeling cellular machinery through biological network comparison. *Nat. Biotechnol.*, 24:427–433, April 2006.

[16] S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nat. Genet.*, 31(1):64–68, 2002.

[17] R. E. Tarjan. Depth first search and linear graph algorithms. *SIAM J. Comp.*, (1):146–160, 1972.

[18] S. Wernicke. Efficient detection of network motifs. *IEEE ACM T. Comput. Bi.*, 3(4):347–359, 2006.