Probably Optimal Graph Motifs*

Andreas Björklund[1], Petteri Kaski[2], and Łukasz Kowalik[3]

1 **Department of Computer Science, Lund University, Sweden**
andreas.bjorklund@yahoo.se
2 **Helsinki Institute for Information Technology HIIT & Department of Information and Computer Science, Aalto University, Finland**
petteri.kaski@aalto.fi
3 **Institute of Informatics, University of Warsaw, Poland**
kowalik@mimuw.edu.pl

Abstract

We show an $O^*(2^k)$-time polynomial space algorithm for the k-sized Graph Motif problem. We also introduce a new optimization variant of the problem, called Closest Graph Motif and solve it within the same time bound. The Closest Graph Motif problem encompasses several previously studied optimization variants, like Maximum Graph Motif, Min-Substitute, and Min-Add.

Moreover, we provide a piece of evidence that our result might be essentially tight: the existence of an $O^*((2-\epsilon)^k)$-time algorithm for the Graph Motif problem implies an $O((2-\epsilon')^n)$-time algorithm for Set Cover.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases graph motif, FPT algorithm

Digital Object Identifier 10.4230/LIPIcs.STACS.2013.20

1 Introduction

The Graph Motif problem is defined as follows. We are given an undirected connected graph $G = (V, E)$, a vertex coloring $c : V \to C$, and a multiset M consisting of colors in the set C. The goal is to find a subset $S \subseteq V$ such that the induced subgraph $G[S]$ is connected, and the *multiset* $c(S)$ of colors of the vertices of S is equal to M. To avoid confusion, let us stress that the input function c is arbitrary and it does not need to be a proper vertex coloring. Let $k = |S|$ denote the size of the solution (which is $|M|$ in the case of Graph Motif but may differ from $|M|$ in variants of the problem also considered in this paper).

Graph Motif was introduced by Lacroix et al. [17] and motivated by applications in bioinformatics, specifically in metabolic network analysis. It is known to be NP-hard even when the given graph is a tree of maximum degree 3 and the motif is a set [11]. However, in practice the size of M is expected to be small, what motivates the research on so-called FPT algorithms parameterized by k, that is, algorithms with running times bounded from above by a function $f(k)$ times a function polynomial in the input size, which is commonly abbreviated by $O^*(f(k))$. Indeed, Fellows et al. [10] discovered that such an algorithm exists, which was followed by a rapid series of improvements to $f(k)$ (see Table 1).

* This research was supported in part by the Academy of Finland, Grants 252083 and 256287 (P.K.), and by the National Science Centre of Poland, Grant N206 567140 (Ł.K.).

30th Symposium on Theoretical Aspects of Computer Science (STACS'13).
Editors: Natacha Portier and Thomas Wilke; pp. 20–31

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 The progress on FPT algorithms for the Graph Motif problem

Paper	Running time	Approach
Fellows et al. [10]	$O^*(87^k)$, implicit	Color-coding
Betzler et al. [1]	$O^*(4.32^k)$	Color-coding
Guillemot and Sikora [12]	$O^*(4^k)$	Multilinear detection
Koutis [15]	$O^*(2.54^k)$	Constrained multilinear detection
this work	$O^*(2^k)$	Constrained multilinear detection

The two most recent results, namely the $O^*(4^k)$ algorithm of Guillemot and Sikora [12] and the $O^*(2.54^k)$-time algorithm of Koutis [15] apply the so-called multilinear detection technique. This technique was introduced by Koutis [13] and further developed by Williams [20] and Koutis and Williams [16] to solve subgraph containment problems. Inspired by the Graph Motif application and the reduction to multilinear detection [12], Koutis [15] introduced a tailored variant of multilinear detection called constrained multilinear detection (abbreviated k-CMLD) to make further progress on the Graph Motif problem. In the k-CMLD problem, we are given a multivariate polynomial represented as an arithmetic circuit, and are asked determine whether the polynomial has a multilinear monomial of degree k with an odd coefficient, with the further constraint that the polynomial indeterminates have colors associated to them, and the monomial must not exceed a prescribed number of occurrences of each color. At a Dagstuhl seminar in 2010, Koutis [14] posed the open problem of devising an $O^*(2^k)$-time algorithm for k-CMLD. His recent paper [15] provides an $O^*(2.54^k)$-time algorithm for the problem where the worst case bound results from a budget of at most three occurrences of every color.

In this paper we indirectly show an $O^*(2^k)$-time polynomial space algorithm for k-CMLD, thereby answering Koutis's open problem in the affirmative. Since the main application of k-CMLD is the Graph Motif problem and its variants, we here present the result directly in terms of graph motifs. In the full version of this paper we will give a self-contained proof for the general k-CMLD (see also [4]). Our approach is inspired by Koutis's beautiful idea of assigning random subspaces of dimension equal to the prescribed multiplicities of the colors. Koutis used group algebras $\mathbb{F}_2[\mathbb{Z}_2^k]$ for his construction, whereas ours appear to require an extension to $\mathbb{F}_{2^\beta}[\mathbb{Z}_2^k]$ for $\beta = \Omega(\log k)$. Rather than proving the result in terms of a group algebra as Koutis suggests, we provide a construction using inclusion–exclusion over labelled indeterminates. As in [3], a paper using the technique co- authored by a subset of the present authors, we find it more convenient to work in this alternative setting.

A further contribution of the present work is to develop a generalization of Graph Motif called Closest Graph Motif. In particular, we introduce a notion of edit distance between two multisets, and the objective is to find the subset $S \subseteq V$ such that $G[S]$ is connected and the edit distance from M to $c(S)$ is minimized (see Section 3 for a precise definition). In the literature there are some more optimization variants of Graph Motif and our Closest Graph Motif generalizes three of them: Maximum Graph Motif (see Section 2), Min-Add, and Min-Substitute (see Section 3). The previous fastest algorithms for these problems are due to Koutis [15]; he shows $O^*(2.54^k)$-time algorithms for Maximum Graph Motif and Min-Add, and an $O^*(5.08^k)$-time algorithm for Min-Substitute. We present an $O^*(2^k)$-time algorithm for the general Closest Graph Motif.

Similarly to the three previous FPT improvements on Graph Motif relative to the

parameter k, our algorithm is Monte Carlo with one-sided error (with probability at most $1/2$ the algorithm asserts the absence of a solution when in fact there is one; one can get arbitrarily small error probability p by repeating the algorithm $\log_2(1/p)$ times).

In addition to our algorithmic results, we give a piece of evidence that further improvement on the running time is substantially harder. Namely, we show that for any $\epsilon > 0$ the existence of an $O^*((2-\epsilon)^k)$-time algorithm for the Graph Motif problem implies an $O((2-\epsilon')^n)$-time algorithm for the Set Cover problem, for some $\epsilon' > 0$. Thus, instead of trying to improve our algorithm one should rather attack the more generic Set Cover. After decades of research on the problem, an $O((2-\epsilon)^n)$-time algorithm for Set Cover would be a major breakthrough. Indeed, the nonexistence of such an algorithm has already been used as an assumption for proving hardness results [5]. Furthermore, it is conjectured [5] that an $O((2-\epsilon)^n)$-time algorithm for Set Cover contradicts the SETH (Strong Exponential Time Hypothesis, which states that if k-CNF SAT can be solved in $O^*(c_k^n)$ time, then $\lim_{k\to\infty} c_k = 2$). This conjecture is supported by the fact that the number of solutions to Set Cover cannot be computed in $O((2-\epsilon)^n)$ time for any $\epsilon > 0$ unless SETH fails [5]. Another consequence of such a counting algorithm would be the existence of an $O((2-\epsilon')^n)$-time algorithm to compute the permanent of an $n \times n$ integer matrix, see [2].

This paper is organized as follows. In Section 2 we describe our $O^*(2^k)$-time algorithm for Maximum Graph Motif, while in Section 3 we describe how our algorithm works for the more general Closest Graph Motif and how it encompasses earlier named variants. In Section 4 we show the reduction from Set Cover.

2 An $O^*(2^k)$-time algorithm for Maximum Graph Motif

2.1 The general approach

In this section we will study a slight generalization of Graph Motif, namely the Maximum Graph Motif problem parameterized by the solution size k. That is, for a given instance (G, c, M), the task is to decide whether there exists a subset $S \subseteq V$ such that (i) the induced subgraph $G[S]$ is connected, (ii) $|S| = k$, and (iii) the multiset inclusion $c(S) \subseteq M$ holds.

It is immediate that once we have an algorithm for the decision problem that runs in $T(n,k)$ time, we can *find* a solution in $O(nT(n,k))$ time as follows. For every vertex $v \in V$, check whether a solution exists if we remove v frmo G; if not, then put v back to G; in both cases proceed to the next vertex. After iterating over all vertices, we are left with a desired induced subgraph $G[S]$.

To solve the decision problem we use the following approach. We define a multivariate polynomial P that has two key properties: (1) P is not identically zero if and only if there is a solution S, and (2) P can be evaluated fast (that is, in $O^*(2^k)$ time) at any given point. Then an $O^*(2^k)$-time algorithm follows via the DeMillo–Lipton–Schwartz–Zippel Lemma (see Section 2.6). We note that alternatively, instead of the polynomial approach one can use the random weights approach and the Isolation Lemma (see e.g. [6]).

The main difference between the present approach and earlier works that deploy a similar polynomial sieve is that we employ the same "labels" (the universe of k elements whose subsets we use in sieving) to simultaneously accomplish two different tasks: (i) we sieve out all homomorphisms that are not injective, and (ii) we sieve out all multisets that use too many colors when compared with M.

The rest of the section is organized as follows. First we give some preliminaries and notation on branching walks and labellings in Sections 2.2 and 2.3. In Section 2.4 we define the polynomial P and we prove the property (1) as Lemma 1. In Section 2.5 we show

property (2) and finally in Section 2.6 we describe the complete algorithm and analyze its failure probability using the DeMillo–Lipton–Schwartz–Zippel Lemma.

2.2 Preliminaries on branching walks

The concept of branching walks was first introduced by Nederlof [18] to sieve for Steiner trees. Here we provide a slightly modified definition of branching walks. Let G be a graph with vertex set $V = V(G)$ and edge set $E = E(G)$. A mapping $h : V(T) \to V(G)$ is a *homomorphism* from a graph T to a graph G if for all $\{u, v\} \in E(T)$ it holds that $\{h(u), h(v)\} \in E(G)$. We adopt the convention of calling the elements of $V(T)$ *nodes* and the elements of $V(G)$ *vertices*.

A *branching walk* G is a pair $W = (T, h)$ where T is an ordered rooted tree with node set $V(T) = \{1, 2, \ldots, |V(T)|\}$ such that every node $v \in V(T)$ coincides with its rank in the preorder traversal of T, and $h : V(T) \to V(G)$ is a homomorphism from T to G. For a vertex $r \in V$, we say that W *starts* from r if $h(1) = r$.

Let $W = (T, h)$ be a branching walk in G. We define $h(T)$ to be the subgraph of G induced by the set of edges $\{\{h(u), h(v)\} : \{u, v\} \in E(T)\}$. We observe that $h(T)$ is not necessarily a tree because h need not be injective. We say that W is *simple* if $h(T)$ is injective.

It will be convenient to assume that $V(G)$ is totally ordered. Towards this end, let us assume that $V = V(G) = \{1, 2, \ldots, n\}$. We say that a branching walk $W = (T, h)$ in G is *properly ordered* if any two sibling nodes $u < v$ in T satisfy $h(u) < h(v)$.

2.3 Labelling and shading

Let (G, c, M) be the input instance of the Maximum Graph Motif problem and let $m : C \to \mathbb{N}$ be the multiplicity function for M. For each color $q \in C$ and $i = 1, 2, \ldots, m(q)$, let us call the formal pair (q, i) the *i-th shade of color q*. In particular, the number of shades for each color matches the multiplicity of the color in M. Let us write $D(q) = \{(q, i) : i = 1, 2, \ldots, m(q)\}$ for the set of all shades of color $q \in C$, and $D = \cup_{q \in C} D(q)$ for the set of all shades of all colors.

A branching walk (T, h) in G may be *labelled* with a function $\ell : V(T) \to \{1, 2, \ldots, k\}$. The three-tuple (T, h, ℓ) is called a *labelled branching walk*, and the function ℓ is a *labelling* of the branching walk.

A branching walk (T, h) in G may also be *shaded* with a function $s : V(T) \to D$. A shading may also be *partial*, that is, of the form $s : U \to D$ for a subset $U \subseteq V(T)$. We say that a (partial) shading $s : U \to D$ of a branching walk (T, h) is *consistent* with the input coloring $c : V(G) \to C$ if for every node $v \in U$ we have $s(v) \in D(c(h(v)))$. For a branching walk $W = (T, h)$ and a subset $U \subseteq V(T)$, denote by $\mathcal{S}_U(W)$ the set of all consistent (partial) shadings $s : U \to D$. Let us abbreviate $\mathcal{S}(W) = \mathcal{S}_{V(T)}(W)$.

2.4 The polynomial P

We use three different types of indeterminates in our polynomials. First, for each edge $\{a, b\} \in E(G)$, introduce two indeterminates $x_{a,b}$ and $x_{b,a}$. Second, for each vertex $a \in V(G)$ and each shade $t \in D(c(a))$, introduce an indeterminate $y_{a,t}$. Third, for each shade $t \in D$ and each label $j \in \{1, 2, \ldots, k\}$, introduce an indeterminate $z_{t,j}$. Let us write $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ for the sequences of all the $x_{a,b}$-type, $y_{a,t}$-type, and $z_{t,j}$-type variables, respectively.

Let $W = (T, h)$ be a branching walk in G, let $s : V(T) \to D$ be a consistent shading of W, and let $\ell : V(T) \to \{1, 2, \ldots, k\}$ be a labelling of W. Associate with the consistently shaded and labelled branching walk (W, s, ℓ) the monomial

$$\mathrm{mon}(W, s, \ell) = \prod_{\substack{\{u,v\} \in E(T) \\ u < v}} x_{h(u),h(v)} \prod_{v \in V(T)} y_{h(v),s(v)} z_{s(v),\ell(v)} \,.$$

Denote by $\mathcal{W}$ the set of all branching walks of $W = (T, h)$ in G that are properly ordered and satisfy $|V(T)| = k$. Let $\beta = \lceil \log k \rceil + 3$ and denote by $\mathbb{F}_{2^\beta}$ the finite field of order 2^β. Define the multivariate polynomial P with coefficients in $\mathbb{F}_{2^\beta}$ by setting

$$P(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{W=(T,h) \in \mathcal{W}} \sum_{s \in \mathcal{S}(W)} \sum_{\substack{\ell : V(T) \to \{1,2,\ldots,k\} \\ \ell \text{ bijective}}} \mathrm{mon}(W, s, \ell) \,.$$

▶ **Lemma 1.** *We have $P \not\equiv 0$ if and only if there exists a solution $S \subseteq V(G)$.*

Proof. ($\Leftarrow$) Let T_S be a spanning tree of $G[S]$. Transform T_S into a rooted tree, and make T_S an ordered tree so that the children of every vertex listed in tree order form an increasing sequence. If we replace every vertex in T_S with its rank in a preorder traversal, we obtain a properly ordered simple branching walk $W = (T, h)$. Define the shading $s : V(T) \to D$ for each node $v \in V(T)$ by setting

$$s(v) = (c(h(v)), |\{w \in S : c(h(w)) = c(h(v)) \text{ and } w \leq v\}|) \,.$$

Note that s is well-defined and consistent because S is a solution. Furthermore, observe that s is injective. Finally, choose an arbitrary bijection $\ell : V(T) \to \{1, 2, \ldots, k\}$. We must now have $P \not\equiv 0$ because we can uniquely reconstruct any three-tuple (W, s, ℓ) with a simple $W = (T, h) \in \mathcal{W}$, an injective $s \in \mathcal{S}(W)$, and a bijective $\ell : V(T) \to \{1, 2, \ldots, k\}$ from its monomial representation $\mathrm{mon}(W, s, \ell)$ – indeed, first recover $W = (T, h)$ from the $x_{a,b}$-type indeterminates using the fact W is simple, properly ordered, and starts from the unique vertex r such that $\mathrm{mon}(W, s, \ell)$ contains variables of the form $x_{r,v}$ but does not contain variables of the form $x_{v,r}$; then recover s from the $y_{a,t}$-type indeterminates using the fact that h is injective; finally recover ℓ from the $z_{s,j}$-type indeterminates using the fact that s is injective.

($\Rightarrow$) Since $P \not\equiv 0$ there is a branching walk $W = (T, h) \in \mathcal{W}$, a shading $s \in \mathcal{S}(W)$ and a bijective labelling $\ell : V(T) \to \{1, 2, \ldots, k\}$ such that the monomial $\mathrm{mon}(W, s, \ell)$ in P has a nonzero coefficient. We must derive a solution S from (W, s, ℓ).

Let us first show that $\mathrm{mon}(W, s, \ell)$ has zero coefficient in P unless h is injective. Suppose that h is not injective; that is, $h(u_0) = h(v_0)$ for some distinct nodes $u_0, v_0 \in V(T)$. If there are many such pairs, take the minimum pair in the lexicographic ordering of 2-subsets of $V(T)$. Define $\ell' : V(T) \to \{1, 2, \ldots, k\}$ for all $v \in V(T)$ by

$$\ell'(v) = \begin{cases} \ell(v_0) & \text{if } v = u_0, \\ \ell(u_0) & \text{if } v = v_0, \\ \ell(v) & \text{otherwise.} \end{cases}$$

Similarly, define $s' : V(T) \to D$ for all $v \in V(T)$ by

$$s'(v) = \begin{cases} s(v_0) & \text{if } v = u_0, \\ s(u_0) & \text{if } v = v_0, \\ s(v) & \text{otherwise.} \end{cases}$$

We observe that ℓ' is bijective and that $\ell' \neq \ell$ because ℓ is bijective. Moreover, s' is consistent because $c(h(u_0)) = c(h(v_0))$ and s is consistent. In this way, to the triple (W, s, ℓ) we associated a different triple (W, s', ℓ') such that $\mathrm{mon}(W, s, \ell) = \mathrm{mon}(W, s', \ell')$. Since $\mathbb{F}_{2^\beta}$ has characteristic 2, these two monomials cancel out in P. Conversely, if we begin from (W, s', ℓ') and follow the same association rule, we get (W, s, ℓ). Hence, the set of all triples $((T, h), s, \ell)$ in which the homomorphism h is not injective is partitioned into pairs, and the two monomials corresponding to each pair cancel out. It follows that the homomorphism h is injective in every triple $((T, h), s, \ell)$ in the preimage of a monomial with a nonzero coefficient in P.

Next suppose that h is injective but s is not injective. Then there is pair of distinct nodes $u_0, v_0 \in V(T)$ that have the same shade $s(u_0) = s(v_0)$. Again, if there are many such pairs we take the lexicographic minimum pair. Define $\ell' : V(T) \to \{1, 2, \ldots, k\}$ as before. Again, to the triple (W, s, ℓ) we assigned a different triple (W, s, ℓ') and again one can verify that $\mathrm{mon}(W, s, \ell) = \mathrm{mon}(W, s, \ell')$. By a similar argument as above, we see that the two monomials corresponding to these two triples cancel out. It follows that the shading s is injective in every triple $((T, h), s, \ell)$ in the preimage of a monomial with a nonzero coefficient in P.

So we must have that both h and s are injective in a three-tuple $((T, h), s, \ell)$ where the monomial $\mathrm{mon}(W, s, \ell)$ in P has a nonzero coefficient. Let $S = V(h(T))$. Because h is injective, $|S| = k$. Because T is connected and h is a homomorphism, we have that $h(T)$ is connected and hence so is $G[S]$. Since s is consistent and injective, S is a solution. ◀

2.5 Evaluating the polynomial in time $O^*(2^k)$

In this section we show that the polynomial P can be evaluated in a given point $(\mathbf{x}, \mathbf{y}, \mathbf{z})$ in time $O^*(2^k)$. To this end, let us rewrite P as a sum of 2^k polynomials such that each of them can be evaluated in time polynomial in the input size. For each $X \subseteq \{1, 2, \ldots, k\}$, let

$$P_X(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{W=(T,h)\in\mathcal{W}} \sum_{s\in\mathcal{S}(W)} \sum_{\ell:V(T)\to X} \mathrm{mon}(W, s, \ell)\,,$$

Note that we do not assume that the labellings in the third summation are bijective.

▶ **Lemma 2.** $P(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{X\subseteq\{1,2,\ldots,k\}} P_X(\mathbf{x}, \mathbf{y}, \mathbf{z})$.

Proof. Let us fix a branching walk $W = (T, h) \in \mathcal{W}$ such that $|V(T)| = k$ and a shading $s \in \mathcal{S}(W)$. Because $|V(T)| = k$, a function $\ell : V(T) \to \{1, 2, \ldots, k\}$ is bijective if and only if it is surjective, so

$$\sum_{\substack{\ell:V(T)\to\{1,2,\ldots,k\}\\ \ell \text{ bijective}}} \mathrm{mon}(W, s, \ell) = \sum_{\substack{\ell:V(T)\to\{1,2,\ldots,k\}\\ \ell \text{ surjective}}} \mathrm{mon}(W, s, \ell). \tag{1}$$

Observing again that $\mathbb{F}_{2^\beta}$ has characteristic 2, and hence $-1 = 1$, we have, by the Principle of Inclusion and Exclusion,

$$\sum_{\substack{\ell:V(T)\to\{1,2,\ldots,k\}\\ \ell \text{ surjective}}} \mathrm{mon}(W, s, \ell) = \sum_{X\subseteq\{1,2,\ldots,k\}} \sum_{\ell:V(T)\to X} \mathrm{mon}(W, s, \ell)\,. \tag{2}$$

From (1) and (2) we immediately obtain

$$P(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{W=(T,h)\in\mathcal{W}} \sum_{s\in\mathcal{S}(W)} \sum_{X\subseteq\{1,2,\ldots,k\}} \sum_{\ell:V(T)\to X} \mathrm{mon}(W, s, \ell)\,. \tag{3}$$

The claim follows by changing the order of summation. ◀

Now we are left with a tedious job of evaluating $P_X(\mathbf{x}, \mathbf{y}, \mathbf{z})$ in polynomial time. This is slightly technical because we consider properly ordered branching walks. To simplify notation in the running time bounds, let us write e for the number of edges in G, and $\mu = O(\beta \log \beta \log\log \beta)$ for the time needed to multiply or add two elements of $\mathbb{F}_{2^\beta}$.

▶ **Lemma 3.** *Given a nonempty subset $X \subseteq \{1, 2, \ldots, k\}$ and three vectors $\mathbf{x}, \mathbf{y}, \mathbf{z}$ of values in $\mathbb{F}_{2^\beta}$ as input, the value of $P_X(\mathbf{x}, \mathbf{y}, \mathbf{z})$ can be computed by dynamic programming in time $O(k^2 e\mu)$ and space $O(ke\beta)$.*

Proof. Recall that we assume that $V(G) = \{1, 2, \ldots, n\}$. For a vertex $a \in V(G)$, denote the ordered sequence of neighbors of a in G by $a_1 < a_2 < \cdots < a_{\deg_G(a)}$. For each $a \in V(G)$, $1 \leq i \leq \deg_G(a) + 1$, and $0 \leq l \leq k$, denote by $\mathcal{W}(a, i, l)$ the set of properly ordered branching walks $W = (T, h)$ such that (i) W starts from a, (ii) for any child node u of 1 in T it holds that $h(u) = a_j$ implies $j \geq i$, and (iii) $|V(T)| = l$.

Our objective is to compute a three-dimensional array A_X whose entries are defined by

$$A_X[a, i, l] = \sum_{W=(T,h)\in\mathcal{W}(a,i,l)} \sum_{s\in\mathcal{S}_{V(T)\setminus\{1\}}(W)} \sum_{\ell:V(T)\setminus\{1\}\to X} \prod_{\{u,v\}\in E(T)} x_{h(u),h(v)} \prod_{v\in V(T)\setminus\{1\}} y_{h(v),s(v)} z_{s(v),\ell(v)} \,.$$

The entries of A_X admit the following recurrence. For $i = \deg_G(a) + 1$ or $l = 1$, we have

$$A_X[a, i, l] = \begin{cases} 1 & \text{if } l = 1, \\ 0 & \text{otherwise.} \end{cases}$$

For $1 \leq i \leq \deg_G(a)$ and $2 \leq l \leq k$, we have

$$\begin{aligned} A_X[a, i, l] \;=\; & A_X[a, i+1, l] + \\ & x_{a,a_i} \cdot \left(\sum_{t\in D(c(a_i))} \sum_{j\in X} y_{a_i,t} z_{t,j} \right) \cdot \sum_{\substack{l_1+l_2=l \\ l_1,l_2\geq 1}} A_X[a, i+1, l_1] \cdot A_X[a_i, 1, l_2]. \end{aligned} \tag{4}$$

To see that the recurrence is correct, observe that the two lines above correspond to properly ordered branching walks in $\mathcal{W}(a, i, l)$ where either (a) there is no child node u of 1 in T such that $h(u) = a_i$ or (b) there is a unique such child. (At most one such child may exist because the branching walk is properly ordered.)

To recover the value of the polynomial $P_X(\mathbf{x}, \mathbf{y}, \mathbf{z})$, we observe that

$$P_X(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_{r\in V} \sum_{t\in D(c(r))} \sum_{j\in X} y_{r,t} z_{t,j} \cdot A_X[r, 1, k] \,. \tag{5}$$

The time bound follows by noting that the values of $\sum_{t\in D(c(a))} \sum_{j\in X} y_{a,t} z_{t,j}$ can be precomputed and tabulated in $O(k^2 n\mu)$ time to accelerate the computations for the individual entries of A_X. ◀

2.6 The decision algorithm

▶ **Lemma 4** (DeMillo and Lipton [7], Schwartz [19], Zippel [21]). *Let $P(x_1, x_2, \ldots, x_m)$ be a nonzero polynomial of degree at most d over a field $\mathbb{F}$ and let S be a finite subset of $\mathbb{F}$. Then, the probability that P evaluates to zero on a random element $(a_1, a_2, \ldots, a_m) \in S^m$ is bounded by $d/|S|$.*

▶ **Theorem 5.** *The* Maximum Graph Motif *problem admits a Monte Carlo algorithm that runs in $O(2^k k^2 e\mu)$ time and in polynomial space, with the following guarantees: (i) the algorithm always returns NO when given a NO-instance as input, (ii) the algorithm returns YES with probability at least $1/2$ when given a YES-instance as input.*

Proof. The algorithm is as follows. Select values for the variables in $\mathbf{x}, \mathbf{y}, \mathbf{z}$ independently and uniformly at random from $\mathbb{F}_{2^\beta}$. Then iterate over all $X \subseteq \{1, 2, \ldots, k\}$ and use the algorithm in Lemma 3 to evaluate the value $P_X(\mathbf{x}, \mathbf{y}, \mathbf{z})$. Accumulate the sum of the values $P_X(\mathbf{x}, \mathbf{y}, \mathbf{z})$ to obtain $P(\mathbf{x}, \mathbf{y}, \mathbf{z})$ by Lemma 2. If $P(\mathbf{x}, \mathbf{y}, \mathbf{z}) \neq 0$, answer YES and otherwise answer NO.

When the input is a NO instance, the polynomial P is the zero polynomial by Lemma 1, so the algorithm returns NO. When the input is a YES instance, by Lemma 1 the polynomial P is nonzero. The degree of P is exactly $3k - 1$, while the size of $\mathbb{F}_{2^\beta}$ is $2^{\lceil \log k \rceil + 3} \geq 8k$. Thus, by Lemma 4 the probability that P evaluated to 0 is bounded by $\frac{3k-1}{8k} < \frac{1}{2}$. ◀

3 Variants of the Graph Motif Problem

In Section 2 we described an algorithm for Maximum Graph Motif. It is easy to see that the algorithm can also be used to solve classical Graph Motif by setting $k = |M|$. Another variant of the problem studied in the literature is the list version of the problem, where every vertex $a \in V(G)$ is assigned *a set* of colors $C(a) \subseteq C$, not just one color $c(a)$, and for every vertex in a solution we can choose any of its colors to match the multiset M. It is straightforward to modify our algorithm for Maximum Graph Motif to solve the list version: in the dynamic programming of Lemma 3, in (4) instead of summing over $t \in D(c(a_i))$ we sum over $t \in \bigcup_{q \in C(a_i)} D(q)$, and similarly in (5).

Although Maximum Graph Motif (introduced in [8]) is a natural optimization version of the problem, it is not the only one. Two more optimization variants were introduced in [9]; we describe their decision versions below.

Min-Add
Input: Graph $G = (V, E)$, a coloring $c : V \to C$, a multiset of colors M, and $d \in \mathbb{N}$.
Question: Is there a subset $S \subseteq V$ such that $G[S]$ is connected, $M \subseteq c(S)$ and $|c(S) \setminus M| \leq d$?

Min-Substitute
Input: Graph $G = (V, E)$, a coloring $c : V \to C$, a multiset of colors M, and $d \in \mathbb{N}$.
Question: Is there a subset $S \subseteq V$ such that $G[S]$ is connected and $c(S)$ can be obtained from M with at most d substitutions?

In this paper we introduce a new variant, which is a generalization of Maximum Graph Motif, Min-Add and Min-Substitute. We believe that it might be useful in bioinformatics applications.

Consider the following three operations on a multiset M over a set of colors C:

1. insertion (I): adds a copy of $c \in C$ to M,

2. deletion (D): removes a copy of $c \in M$ from M,
3. substitution (S): removes a copy of $c_1 \in M$ from M and adds a copy of $c_2 \in C$ to M.

Associate with each of the three operations a nonnegative integer cost $\kappa_I, \kappa_D, \kappa_S$. Consider a sequence σ of the three operations applied to a multiset M. Let $m_{\mathrm{I}}, m_{\mathrm{D}}, m_{\mathrm{S}}$ be the numbers of insertions, deletions, and substitutions in σ. Then the *cost* of σ is defined as $m_{\mathrm{I}}\kappa_{\mathrm{I}} + m_{\mathrm{D}}\kappa_{\mathrm{D}} + m_{\mathrm{S}}\kappa_{\mathrm{S}}$. Moreover, for two multisets M and M', the *weighted edit distance* is defined as the minimum cost $\kappa(M, M')$ of a sequence of operations that turns M into M'.

Closest Graph Motif
Input: Graph $G = (V, E)$, a coloring $c : V \to C$, a multiset of colors M, and numbers $d, \kappa_{\mathrm{I}}, \kappa_{\mathrm{D}}, \kappa_{\mathrm{S}} \in \mathbb{N}$.
Question: Is there a subset $S \subseteq V$ such that $G[S]$ is connected and $\kappa(M, c(S)) \leq d$?

Note that Min Add reduces to Closest Graph Motif by settting $\kappa_{\mathrm{I}} = 1$, $\kappa_{\mathrm{D}} = \kappa_{\mathrm{S}} = d + 1$. Similarly, for Min Substitute set $\kappa_{\mathrm{S}} = 1$, $\kappa_{\mathrm{I}} = \kappa_{\mathrm{D}} = d + 1$.

We next describe an algorithm for Closest Graph Motif subject to the parameterization that we are given an additional integer $k \in \mathbb{N}$ as input and the subset S must satisfy $|S| = k$. We will present an $O^*(2^k)$-time polynomial space algorithm, assuming that d is bounded by a polynomial function in n. Note that when parameterized by the edit distance (which also seems natural) the problem is unlikely to admit an FPT algorithm since Graph Motif is NP-hard. Since the algorithm is a rather straightforward extension of the algorithm for Maximum Graph Motif presented in Section 2, we only sketch it by describing the modifications needed to handle the more general problem.

We proceed to define an analog of the polynomial P from Section 2. We use the same indeterminates as before, with additional indeterminates for tracking substitutions and the edit distance. Towards this end, for each $a \in V(G)$, introduce the indeterminate w_a. Denote by $\mathbf{w}$ the sequence of all such indeterminates. Introduce one further indeterminate η for tracking the edit distance.

Recall that in the polynomial P, every monomial corresponds to a consistently shaded and bijectively labelled branching walk (W, s, ℓ). In the new polynomial Q, every monomial corresponds to a quadruple (W, f, s, ℓ), where $f : V(T) \to \{0, 1\}$ is an indicator function for substitutions. That is, $f(v) = 1$ and $s(v) = (q, i)$ for a node v means that a copy of color q (the copy corresponding to the ith shade of q) is substituted by the color $c(h(v))$. We need to modify the set of shades in order to make it accept insertions. For a color $q \in C$ let $D'(q) = \{(q, i) \ : \ i = 1, 2, \ldots, m(q) + \lfloor d/\kappa_{\mathrm{I}} \rfloor\}$ and let $D' = \bigcup_{q \in C} D'(q)$. In Q the shading function s maps $V(T)$ to D'. The meaning of this modification is that the shade (q, i) for $i > m(q)$ corresponds to an inserted shade of color q. Accordingly, the notion of consistency of a shading requires a modification in order to accept substitutions. We say that a (partial) shading $s : U \to D'$ of a branching walk $W = (T, h)$ and a substitution indicator $f : V(T) \to \{0, 1\}$ is *consistent* if for every node $v \in U \subseteq V(T)$ one of the following conditions holds: (i) if $f(v) = 0$ then $s(v) = (c(h(v)), i)$ for some i, or (ii) if $f(v) = 1$ then $s(v) \in D$. For a given branching walk $W = (T, h)$, a substitution indicator $f : V(T) \to \{0, 1\}$, and a set $U \subseteq V(T)$, denote by $\mathcal{S}_U(W, f)$ the set of all (partial) shadings $s : U \to D'$ of W that are consistent. Let us abbreviate $\mathcal{S}(W, f) = \mathcal{S}_{V(T)}(W, f)$.

The following lemma will be useful in the construction of Q.

▶ **Lemma 6.** *Consider a sequence σ of m_I insertions, m_S substitutions and a number of deletions which transforms a multiset M into a multiset M' of size k. Then, the cost of σ is equal to $m_I(\kappa_I + \kappa_D) + m_S\kappa_S + (|M| - k)\kappa_D$.*

Proof. Observe that σ contains $|M| + m_{\mathrm{I}} - k$ deletions. ◀

By the above lemma, given a quadruple (W, f, s, ℓ) the cost of the sequence of operations corresponding to this quadruple is

$$\kappa(f,s) = \left(\sum_{q \in C} \sum_{i=1}^{\lfloor d/\kappa_{\mathrm{I}} \rfloor} |s^{-1}((q, m(q)+i))| \right) (\kappa_{\mathrm{I}} + \kappa_{\mathrm{D}}) + |f^{-1}(1)|\kappa_{\mathrm{S}} + (|M| - k)\kappa_{\mathrm{D}} \,.$$

For a substitution indicator function $f : V(T) \to \{0,1\}$ and a homomorphism $h : V(T) \to V(G)$, let us write $\mathbf{w}_h^f = \prod_{u \in V(T)} w_{h(u)}^{f(u)}$ for the indicator monomial of f given h.

Now we are ready to define the polynomial

$$Q(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \eta) = \sum_{W=(T,h) \in \mathcal{W}} \sum_{f : V(T) \to \{0,1\}} \sum_{s \in \mathcal{S}(W,f)} \sum_{\substack{\ell : V(T) \to \{1,2,\ldots,k\} \\ \ell \text{ bijective}}} \mathrm{mon}(W, s, \ell) \mathbf{w}_h^f \eta^{\kappa(f,s)} \,.$$

▶ **Lemma 7.** *Let $Q(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \eta) = \sum_{i \geq 0} Q_i(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w})\eta^i$. Then, we have $Q_i \not\equiv 0$ for an $i \leq d$ if and only if there exists a solution $S \subseteq V(G)$ with $\kappa(M, c(S)) \leq d$.*

Proof. Analogous to the proof of Lemma 1, with the following modifications to handle the substitution indicator f.

($\Leftarrow$) Recover h as before, use the editing sequence for $i = \kappa(M, c(S)) \leq d$ to construct a substitution indicator f, then define s and ℓ. To conclude that $Q_i \not\equiv 0$, reconstruct a quadruple (W, s, f, ℓ) from its monomial representation as before but with the additional observation that when h is injective we can recover f from $\mathbf{w}_h^f$.

($\Rightarrow$) When h is not injective, define f' from f by transposing the images of u_0 and v_0 under f. When h is injective but s is not injective, proceed as before. Construct S as before. From f and s we can read off a sequence of edits to transform M to $c(S)$ with cost $i \leq d$. ◀

It is immediate that once we can evaluate polynomial Q in a given vector $(\mathbf{x}, \mathbf{y}, \mathbf{z}, \mathbf{w}, \eta)$ we can test whether the polynomials Q_i are nonzero using the DeMillo–Lipton–Schwartz–Zippel Lemma and Lagrange interpolation. By Lemma 2, the task of evaluating Q in time $O^*(2^k)$ boils down to evaluating Q_X in polynomial time (where Q_X is defined analogously as P_X). The evaluation proceeds as in Lemma 3, with minor changes to the dynamic programming recurrence. In particular, in (4) we change the expression $\sum_{s \in D(c(a_i))} \sum_{j \in X} y_{a_i,s} z_{s,j}$ into

$$\sum_{s \in D(c(a_i))} \sum_{j \in X} y_{a_i,s} z_{s,j} + \sum_{p=m(c(a_i))+1}^{m(c(a_i))+\lfloor d/\kappa_{\mathrm{I}} \rfloor} \sum_{j \in X} y_{a_i,(c(a_i),p)} z_{(c(a_i),p),j} \eta^{\kappa_{\mathrm{I}}+\kappa_{\mathrm{D}}} + \sum_{s \in D} \sum_{j \in X} y_{a_i,s} z_{s,j} w_{a_i} \eta^{\kappa_{\mathrm{S}}} \,.$$

The three summands correspond to the three possibilities: (i) a_i just gets a color from M, (ii) a_i gets a new copy of the color $c(a_i)$ that is inserted into M, (iii) one copy of a color from M is substituted by one copy of the color $c(a_i)$. A similar change is required for the expression (5), including also multiplication of the whole expression by $\eta^{(|M|-k)\kappa_{\mathrm{D}}}$. (Alternatively, one may offset the final edit distance by $(|M| - k)\kappa_{\mathrm{D}}$.) Precomputing the above expression takes $O(nk(k + d + |M|)\mu)$ time so single evaluation of Q_X is performed in time $O(k^2 e\mu + nk(k + d + |M|)\mu) = O((ke + nd + n|M|)k\mu)$.

We conclude with the following theorem which follows from considerations in this section. The additional factor of $(k + |M|)d$ in the running time is caused by the use of Lagrange interpolation, which requires $O((k + |M|)d)$ evaluations of Q. Note that the degree of η in Q is bounded by $(k + |M|)(\kappa_{\mathrm{I}} + \kappa_{\mathrm{D}} + \kappa_{\mathrm{S}}) = O((k + |M|)d)$ because we can assume $\kappa_{\mathrm{I}}, \kappa_{\mathrm{D}}, \kappa_{\mathrm{S}} \leq d + 1$. We also note that for every $i \geq 0$, we have $\deg(Q_i) \leq (3k - 1)k$, so to get the bound on the error probability as before, the size of the finite field $\mathbb{F}_{2^\beta}$ should be at least $2(3k - 1)k$. On the other hand, Lagrange interpolation requires $\mathbb{F}_{2^\beta}$ to have size at least $(k + |M|)(\kappa_{\mathrm{I}} + \kappa_{\mathrm{D}} + \kappa_{\mathrm{S}})$. It suffices to put $\beta = \max\{\lceil 2 \log_2 k \rceil, \lceil \log_2 (k + |M|)d \rceil\} + 3$.

▶ **Theorem 8.** *The* Closest Graph Motif *problem admits a Monte Carlo algorithm that runs in* $O(2^k(ke + nd + n|M|)(k + |M|)dk\mu)$ *time and in polynomial space.*

4 A reduction from Set Cover

In the Set Cover problem we are given an integer t and a family of sets $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$ over *the universe* $U = \bigcup_{j=1}^{m} S_j$ with $n = |U|$. The task is to determine whether there is a subfamily of t sets $S_{i_1}, S_{i_2}, \ldots, S_{i_t}$ such that $U = \bigcup_{j=1}^{t} S_{i_j}$.

Cygan *et al.* proved the following result (see Theorem 4.4 in [5][1]).

▶ **Theorem 9** (Cygan et al. [5])**.** *If* Set Cover *can be solved in* $O((2-\epsilon)^{n+t})$ *time for some* $\epsilon > 0$ *then it can also be solved in* $O((2-\epsilon')^n)$ *time, for some* $\epsilon' > 0$.

We use Theorem 9 to show the following.

▶ **Theorem 10.** *If* Graph Motif *can be solved in* $O((2-\epsilon)^k)$ *time for some* $\epsilon > 0$ *then* Set Cover *can be solved in* $O((2-\epsilon')^n)$ *time, for some* $\epsilon' > 0$. *Moreover, this holds even for* Graph Motif *restricted to one of the following two extreme cases:*

(*i*) *M is a set,*
(*ii*) *M has only two distinct colors.*

Proof. Let $(\mathcal{S}, t)$ be an instance of Set Cover. We are going to show a polynomial-time reduction to Graph Motif so that in the resulting instance (G, c, M) the multiset M has cardinality $n + t + 1$. Clearly, combined with Theorem 9, this will prove our claim.

Graph $G = (V, E)$ is defined as follows. The vertex set consists of U, t copies of the family $\mathcal{S}$ and a special vertex r, that is, $V = U \cup \{s_i^j : i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, t\} \cup \{r\}$. Moreover, $E = \{es_i^j : e \in S_i\} \cup \{rs_i^j : i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, t\}$.

To establish case (i), let $M = \{1, 2, \ldots, n + t + 1\}$. Moreover we put $c(s_i^j) = j$ for every $i = 1, 2, \ldots, m,\ j = 1, 2, \ldots, t$. Further, $c(r) = t + 1$. The n colors $t + 2, t + 3, \ldots, n + t + 1$ are assigned bijectively to the vertices from U. Now we show that $(\mathcal{S}, t)$ is a YES-instance of Set Cover iff (G, c, M) is a YES-instance of Graph Motif. Assume $S_{i_1}, S_{i_2}, \ldots, S_{i_t}$ is a solution to Set Cover. Then let $S = \{r\} \cup U \cup \{s_{i_j}^j : j = 1, 2, \ldots, t\}$. It is clear that the multiset of colors on S matches M. Obviously, $G[\{r\} \cup \{s_{i_j}^j : j = 1, 2, \ldots, t\}]$ is connected. Since for every $e \in U$ there is $j = 1, 2, \ldots, t$ such that $e \in S_{i_j}$, so $es_{i_j}^j \in E(G[S])$. It follows that $G[S]$ is connected, and hence S is a solution for Graph Motif. Conversely, if S is a solution for Graph Motif in (G, c, M) then for every $j = 1, 2, \ldots, t$ there is exactly one $i_j \in \{1, 2, \ldots, m\}$ such that $s_{i_j}^j \in S$, since the colors of S match M. Moreover, since $G[S]$ is connected we infer that for every $e \in U$ there is $j = 1, 2, \ldots, t$ such that $es_{i_j}^j \in E(G[S])$. However, then $e \in S_{i_j}$ and it follows that $S_{i_1}, 2, \ldots, S_{i_t}$ is a solution for Set Cover.

To establish case (ii), let M consist of $n + 1$ copies of color 1 and t copies of color 2. We put $c(r) = 1$ and $c(e) = 1$ for every $e \in U$. All the remaining vertices are colored with 2. The equivalence can be shown very similarly to the case (i), we skip the details. ◀

Acknowledgments

We thank reviewers for helpful comments. The third author thanks Sylwia Antoniuk, Marek Cygan, Michal Debski and Matthias Mnich for helpful discussions on related topics.

[1] Actually Theorem 4.4 is stated in a slightly different way, taking into account the maximum size of sets S_i, but Theorem 9 follows immediately from their proof.

References

1 N. Betzler, M. R. Fellows, C. Komusiewicz, and R. Niedermeier. Parameterized algorithms and hardness results for some graph motif problems. In *Proc. CPM'08*, volume 5029 of *LNCS*, pages 31–43, 2008.

2 A. Björklund. Counting perfect matchings as fast as Ryser. In *Proc. SODA'12*, pages 914–921, 2012.

3 A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto. Narrow sieves for parameterized paths and packings. *CoRR*, abs/1007.1161, 2010.

4 A. Björklund, P. Kaski, and Ł. Kowalik. Probably optimal graph motifs. *CoRR*, abs/1209.1082, 2012.

5 M. Cygan, H. Dell, D. Lokshtanov, D. Marx, J. Nederlof, Y. Okamoto, R. Paturi, S. Saurabh, and M. Wahlström. On problems as hard as CNF-SAT. In *IEEE Conference on Computational Complexity*, pages 74–84, 2012.

6 M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proc. FOCS'11*, pages 150–159, 2011.

7 R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7:193–195, 1978.

8 R. Dondi, G. Fertin, and S. Vialette. Maximum motif problem in vertex-colored graphs. In *Proc. CPM'09*, volume 5577 of *LNCS*, pages 221–235, 2009.

9 R. Dondi, G. Fertin, and S. Vialette. Finding approximate and constrained motifs in graphs. In *Proc. CPM'11*, volume 6661 of *LNCS*, pages 388–401, 2011.

10 M. R. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In *Proc. ICALP'07*, volume 4596 of *LNCS*, pages 340–351, 2007.

11 M. R. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Upper and lower bounds for finding connected motifs in vertex-colored graphs. *J. Comput. Syst. Sci.*, 77(4):799–811, 2011.

12 S. Guillemot and F. Sikora. Finding and counting vertex-colored subtrees. In *Proc. MFCS'10*, volume 6281 of *LNCS*, pages 405–416, 2010.

13 I. Koutis. Faster algebraic algorithms for path and packing problems. In *Proc. ICALP'08*, volume 5125 of *LNCS*, pages 575–586, 2008.

14 I. Koutis. The power of group algebras for constrained multilinear monomial detection. *Dagstuhl meeting 10441*, 2010.

15 I. Koutis. Constrained multilinear detection for faster functional motif discovery. *Information Processing Letters*, 112(22):889 – 892, 2012.

16 I. Koutis and R. Williams. Limits and applications of group algebras for parameterized problems. In *ICALP (1)*, volume 5555 of *LNCS*, pages 653–664, 2009.

17 V. Lacroix, C. G. Fernandes, and M.-F. Sagot. Motif search in graphs: Application to metabolic networks. *IEEE/ACM Trans. Comput. Biology Bioinform.*, 3(4):360–368, 2006.

18 J. Nederlof. Fast polynomial-space algorithms using Möbius inversion: Improving on Steiner tree and related problems. In *Proc. ICALP'09*, volume 5555 of *LNCS*, pages 713–725, 2009.

19 J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.

20 R. Williams. Finding paths of length k in $O^*(2^k)$ time. *Inf. Process. Lett.*, 109(6):315–318, 2009.

21 R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proc. International Symposium on Symbolic and Algebraic Computation*, volume 72 of *LNCS*, pages 216–226, 1979.