
This is an electronic reprint of the original article.
This reprint may differ from the original in pagination and typographic detail.

Björklund, Andreas; Kaski, Petteri; Kowalik, Łukasz

Constrained Multilinear Detection and Generalized Graph Motifs

Published in:
Algorithmica

DOI:
[10.1007/s00453-015-9981-1](https://doi.org/10.1007/s00453-015-9981-1)

Published: 01/02/2016

Document Version
Publisher's PDF, also known as Version of record

Published under the following license:
CC BY

Please cite the original version:
Björklund, A., Kaski, P., & Kowalik, . (2016). Constrained Multilinear Detection and Generalized Graph Motifs. *Algorithmica*, 74(2), 947-967. <https://doi.org/10.1007/s00453-015-9981-1>

Constrained Multilinear Detection and Generalized Graph Motifs

Andreas Björklund · Petteri Kaski ·
Łukasz Kowalik

Received: 14 May 2013 / Accepted: 23 February 2015 / Published online: 4 March 2015
© The Author(s) 2015. This article is published with open access at Springerlink.com

Abstract We introduce a new algebraic sieving technique to detect constrained multilinear monomials in multivariate polynomial generating functions given by an evaluation oracle. The polynomials are assumed to have coefficients from a field of characteristic two. As applications of the technique, 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 GRAPH MOTIF, and MIN-ADD GRAPH MOTIF. Finally, 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.

A preliminary conference abstract of this work has appeared as A. Björklund, P. Kaski, and Ł. Kowalik, “Probably optimal graph motifs,” Proceedings of the 30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013, Kiel, February 27–March 2, 2013), Leibniz International Proceedings in Informatics 20, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2013, pp. 20–31.

A. Björklund

Department of Computer Science, Lund University, P.O. Box 118, 22100 Lund, Sweden
e-mail: andreas.bjorklund@yahoo.se

P. Kaski

Department of Information and Computer Science, Helsinki Institute for Information Technology HIIT, Aalto University, P.O. Box 15400, 00076 Aalto, Finland
e-mail: petteri.kaski@aalto.fi

Ł. Kowalik (✉)

Institute of Informatics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland
e-mail: kowalik@mimuw.edu.pl

Keywords Constrained multilinear detection · Graph motif · FPT algorithm

1 Introduction

Many hard combinatorial problems can be reduced to the framework of detecting whether a multivariate polynomial $P(\mathbf{x}) = P(x_1, x_2, \dots, x_n)$ has a monomial with specific properties of interest. In such a setup, $P(\mathbf{x})$ is not available in explicit symbolic form but is implicitly defined by the problem instance at hand, and our access to $P(\mathbf{x})$ is restricted to having an efficient algorithm for computing values of $P(\mathbf{x})$ at points of our choosing. This framework was pioneered by Koutis [14], Williams [21], and Koutis and Williams [17] for use in the domain of parameterized subgraph containment problems, and it currently underlies the fastest known parameterized algorithms for many basic tasks such as path and packing problems [4].

The present paper is motivated by recent works of Guillemot and Sikora [13] and Koutis [16], who observed that functional motif discovery problems in bioinformatics are also amenable to efficient parameterized solution in the polynomial framework. Following Koutis [16], applications in this domain require one to detect monomials in $P(\mathbf{x})$ that are both *multilinear* and further *constrained* by means of colors assigned to variables \mathbf{x} , so that the combined degree of variables of each color in the monomial may not exceed a given maximum multiplicity for that color. Our objectives in this paper are to (1) present an improved algebraic technique for constrained multilinear detection in polynomials over fields of characteristic 2, (2) generalize the technique to allow for approximate matching at cost, and (3) derive improved algorithms for graph motif problems, together with evidence that our algorithms may be optimal in the exponential part of their running time. We also introduce a new common generalization—the *closest graph motif problem*—that tracks the weighted edit distance between the target motif and each candidate pattern; this in particular generalizes both the minimum substitution and minimum addition variants of the graph motif problem introduced by Dondi et al. [10].

Let us now describe our main results in more detail, starting with algebraic contributions and then proceeding to applications in graph motifs. All the algebraic contributions rely essentially on what can be called the “substitution-sieving” method in characteristic 2 [2,4].

1.1 Multilinearity

To ease the exposition and the subsequent proofs, it will be convenient to start with a known, non-constrained version of the substitution sieve that exposes multilinear monomials.

Let $P(\mathbf{x}) = P(x_1, x_2, \dots, x_n)$ be a multivariate polynomial over a field of characteristic 2 such that every monomial $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$ has total degree $d_1 + d_2 + \cdots + d_n = k$. A monomial is *multilinear* if $d_1, d_2, \dots, d_n \in \{0, 1\}$.

For an integer n , let us write $[n] = \{1, 2, \dots, n\}$. Let L be a set of k labels. For each index $i \in [n]$ and label $j \in L$, introduce a new variable $z_{i,j}$. Denote by \mathbf{z} the vector of all variables $z_{i,j}$.

Lemma 1 (Non-constrained multilinear detection [2,4])

The polynomial $P(\mathbf{x})$ has at least one multilinear monomial if and only if the polynomial

$$Q(\mathbf{z}) = \sum_{A \subseteq L} P(z_1^A, z_2^A, \dots, z_n^A) \quad (1)$$

is not identically zero, where $z_i^A = \sum_{j \in A} z_{i,j}$ for all $i \in [n]$ and $A \subseteq L$.

Remark We can now observe the basic structure of the sieve (1): by making 2^k substitutions of the new variables \mathbf{z} into $P(\mathbf{x})$, we reduce the question of existence of a multilinear monomial in $P(\mathbf{x})$ into the question whether the polynomial $Q(\mathbf{z})$ is not identically zero. The latter can be tested probabilistically by one *evaluation* of $Q(\mathbf{z})$ at a random point, which reduces via (1) into *evaluations* of $P(\mathbf{x})$ at 2^k points. This will be the basic structure in all our subsequent algorithm designs.

1.2 Constrained Multilinearity

We are now ready to state our main algebraic contribution. Let C be a set of at most n colors such that each color $q \in C$ has a maximum multiplicity $m(q) \in \{0, 1, \dots, n\}$. Associate with each index $i \in [n]$ a color $c(i) \in C$. Let us say that a monomial $x_1^{d_1} x_2^{d_2} \dots x_n^{d_n}$ is *properly colored* if the number of occurrences of each color is at most its maximum multiplicity, or equivalently, for all $q \in C$ it holds that $\sum_{i \in c^{-1}(q)} d_i \leq m(q)$.

Associate with each color $q \in C$ a set S_q of $m(q)$ shades of the color q , such that S_q and $S_{q'}$ are disjoint whenever $q \neq q'$. Let $S = \cup_{q \in C} S_q$.

For each index $i \in [n]$ and each shade $d \in S_{c(i)}$, introduce a new variable $v_{i,d}$. For each shade $d \in S$ and each label $j \in L$, introduce a new variable $w_{d,j}$.

Lemma 2 (Constrained multilinear detection) The polynomial $P(\mathbf{x})$ has at least one monomial that is both multilinear and properly colored if and only if the polynomial

$$Q(\mathbf{v}, \mathbf{w}) = \sum_{A \subseteq L} P(u_1^A, u_2^A, \dots, u_n^A) \quad (2)$$

is not identically zero, where $u_i^A = \sum_{j \in A} u_{i,j}$ and $u_{i,j} = \sum_{d \in S_{c(i)}} v_{i,d} w_{d,j}$ for all $i \in [n]$, $j \in L$, and $A \subseteq L$.

Remark This lemma enables us to (probabilistically) detect a constrained multilinear monomial of degree k using 2^k evaluations of $P(\mathbf{x})$, assuming that we are working over a sufficiently large field of characteristic 2. This solves an open problem posed by Koutis at a Dagstuhl seminar in 2010 [15], and forms the core of our algorithm in Theorem 4.

1.3 Cost-Constrained Multilinearity

The previous setting admits a generalization where we associate *costs* to decisions to arrive at a proper coloring. Accordingly, we assume that no coloring $c : [n] \rightarrow C$ has

been fixed *a priori*, but instead associate with each index $i \in [n]$ and each color $q \in C$ a nonnegative integer $\kappa_i(q)$, the *cost* of assigning the color q to i .

Once a coloring $c : [n] \rightarrow C$ has been assigned, the *cost* of a monomial $x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$ in the assigned coloring is $\sum_{i \in [n]} d_i \kappa_i(c(i))$. The objective now becomes to detect a multilinear monomial that has the minimum cost under a proper coloring.

For each index $i \in [n]$ and each shade $d \in S$, introduce a new variable $v_{i,d}$. For each shade $d \in S$ and each label $j \in L$, introduce a new variable $w_{d,j}$. Introduce a new variable η .

Lemma 3 (Cost-constrained multilinear detection) *The polynomial $P(\mathbf{x})$ has at least one monomial that is both multilinear and admits a proper coloring with cost σ if and only if the polynomial*

$$Q(\mathbf{v}, \mathbf{w}, \eta) = \sum_{A \subseteq L} P(u_1^A, u_2^A, \dots, u_n^A) \quad (3)$$

has at least one monomial whose degree in the variable η is σ , where $u_i^A = \sum_{j \in A} u_{i,j}$ and

$$u_{i,j} = \sum_{q \in C} \eta^{\kappa_i(q)} \sum_{d \in S_q} v_{i,d} w_{d,j} \quad (4)$$

for all $i \in [n]$, $j \in L$, and $A \subseteq L$.

Remark The previous lemma may be extended to track multiple cost parameters η_1, η_2, \dots simultaneously. In fact, this will be convenient in our algorithm underlying Theorem 5. We also observe that in applications one typically works with a (random) evaluation in the variables \mathbf{v} and \mathbf{w} , but seeks to recover an explicit polynomial in η as the output of the sieve, typically by a sequence of evaluations at distinct points, followed by interpolation to recover the polynomial in η .

1.4 Graph Motif Problems

The application protagonist for our algebraic tools will be the following problem and its generalization.

MAXIMUM GRAPH MOTIF [9]

Input: A connected, undirected host graph H with n vertices and e edges, a multiset M of colors over a base color set C , a coloring $c : V(H) \rightarrow C$ for the vertices of H , and a positive integer k .

Question: Is there a subset $K \subseteq V(H)$ of size k such that (a) the subgraph induced by K in H is connected, and (b) the multiset $c(K)$ of colors is a subset of M , taking multiplicities into account?

Background Graph motif problems were introduced by Lacroix et al. [18] and motivated by applications in bioinformatics, specifically in metabolic network analysis. The MAXIMUM GRAPH MOTIF problem was introduced by Dondi et al. [9]. It is known

Table 1 Progress on FPT algorithms for the k -sized graph motif problem

Paper	Running time	Approach
Fellows et al. [11]	$O^*(87^k)$, implicit	Color-coding
Betzler et al. [1]	$O^*(4.32^k)$	Color-coding
Guillemot and Sikora [13]	$O^*(4^k)$	Multilinear detection
Koutis [16]	$O^*(2.54^k)$	Constrained multilinear detection
This work	$O^*(2^k)$	Constrained multilinear detection

to be NP-hard even when the given graph is a tree of maximum degree 3 and each color may occur at most once [12]. However, in practice the parameter k 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. [11] discovered that such an algorithm exists, which was followed by a rapid series of improvements to $f(k)$ [1, 11, 13], culminating in the $O^*(2.54^k)$ -time algorithm of Koutis [16] (see Table 1).

From a high-level perspective the two key ideas underlying our main theorem in this section are (1) an observation of Guillemot and Sikora [13] that *branching walks* [19] yield an efficient polynomial generating function for connected sets, and (2) Lemma 2 that builds on work by Koutis [16]. More precisely, 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 based on inclusion–exclusion principle and labelled indeterminates. As in [4], 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.

Our results. The coefficient $\mu = O(\log k \log \log k \log \log \log k)$ in the following theorem reflects the time complexity of basic arithmetic (addition, multiplication) in a finite field of size $O(k)$ and characteristic 2 [6].

Theorem 4 *There exists a Monte Carlo algorithm for MAXIMUM GRAPH MOTIF 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.*

Remark We observe that the algorithm in Theorem 4 runs in linear time in the number of edges e in the host graph H . Furthermore, the exponential part 2^k of the running time is caused by the sieve (2), implying that the algorithm can be executed in parallel on up to 2^k processors with essentially linear speedup. A caveat of the algorithm is that it solves only the YES/NO-decision problem, however, it can be extended to extract a solution set K at additional multiplicative cost k to the running time; this extension will be pursued elsewhere.

1.5 Weighted Edit Distance and the Closest Motif Problem

A natural generalization of the basic graph motif framework is to allow for weighted inexact matches between the “target” motif M and a connected induced subgraph. Such variants have been studied in the literature, in particular by Dondi et al. [10] in the context of either (a) addition of colors to M or (b) substitutions of colors in M . We state both problems below as decision problems parameterized by k .

MIN-ADD GRAPH MOTIF [10]

Input: A connected, undirected host graph H , a multiset M of colors over a base color set C , a coloring $c : V(H) \rightarrow C$ for the vertices of H , a positive integer k , and a nonnegative integer d .

Question: Is there a subset $K \subseteq V(H)$ of size k such that (a) the subgraph induced by K in H is connected, and (b) it holds that $M \subseteq c(K)$ and $|c(K) \setminus M| \leq d$, taking multiplicities into account?

MIN-SUBSTITUTE GRAPH MOTIF [10]

Input: A connected, undirected host graph H , a multiset M of colors over a base color set C , a coloring $c : V(H) \rightarrow C$ for the vertices of H , a positive integer k , and a nonnegative integer d .

Question: Is there a subset $K \subseteq V(H)$ of size k such that (a) the subgraph induced by K in H is connected, and (b) it holds that M can be transformed to $c(K)$ by at most d substitutions of colors, taking multiplicities into account?

Koutis [16] gives an $O^*(2.54^k)$ -time algorithm for MIN-ADD GRAPH MOTIF and an $O^*(5.08^k)$ -time algorithm for MIN-SUBSTITUTE GRAPH MOTIF.

Our objective here is to generalize the graph motif framework to weighted edit distance between M and $c(K)$ by introducing a common generalization, the CLOSEST GRAPH MOTIF problem. We then use Lemma 3 to obtain an $O^*(2^k)$ -time algorithm for the problem.

We start with some preliminaries to give a precise meaning to “closest” via the weighted edit distance. Let M be a multiset over a base set of colors C_0 . Let us allow to change M by means of three basic operations:

- (S) substitute one occurrence of a color $q \in M$ with a color $q' \in C_0$,
- (I) insert one occurrence of a color $q \in C_0$ to M , and
- (D) delete one occurrence of a color $q \in M$ from M .

Associate with each basic operation (S), (I), (D) an nonnegative integer *cost* $\sigma_S, \sigma_I, \sigma_D$.

For multisets M and N over C_0 , the *cost* (or *weighted edit distance*) to match M with N is the minimum cost of a sequence of basic operations that transforms M to N , where the cost of the sequence is the sum of costs of the basic operations in the sequence.

CLOSEST GRAPH MOTIF

Input: A connected, undirected host graph H with n vertices and e edges, a multiset M of colors over a base color set C_0 , a coloring $c : V(H) \rightarrow C_0$ for the vertices of H , nonnegative integer costs $\sigma_S, \sigma_I, \sigma_D$, a threshold cost τ , and a positive integer k .

Question: Is there a subset $K \subseteq V(H)$ of size k such that (a) the subgraph induced by K in H is connected and (b) the cost to transform the multiset M into the multiset $c(K)$ is at most τ ?

Our results Our main result in this section is as follows.

Theorem 5 *There exists a Monte Carlo algorithm for CLOSEST GRAPH MOTIF that runs in $O((2^k k^4 + |C_0| k^3) e \mu)$ 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.*

Remark Similar remarks apply to Theorem 5 as with Theorem 4. In particular, the implementation of (3) with two cost parameters enables essentially linear parallel speedup on up to $2^k k^2$ processors.

1.6 A Lower Bound

There is some evidence that the exponential part 2^k in the running time of the algorithms in Theorem 4 and Theorem 5 may be the best possible. Our approach is to proceed by reduction from the set cover problem.

SET COVER

Input: An integer t and a family of sets $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ over the universe $U = \bigcup_{j=1}^m S_j$ with $n = |U|$.

Question: Is there a subfamily of t sets $S_{i_1}, S_{i_2}, \dots, S_{i_t}$ such that $U = \bigcup_{j=1}^t S_{i_j}$?

We show that for any $\epsilon > 0$ the existence of an $O^*((2 - \epsilon)^k)$ -time algorithm for MAXIMUM GRAPH MOTIF implies an $O((2 - \epsilon')^n)$ -time algorithm for SET COVER, for some $\epsilon' > 0$. Thus, instead of trying to improve our algorithm one should rather directly attack SET COVER, for which all attempts to obtain a $O((2 - \epsilon)^n)$ -time algorithm have failed, despite extensive effort. Indeed, the nonexistence of such an algorithm is already used as a basis for hardness results [7]. Furthermore, it is conjectured [7] that an $O((2 - \epsilon)^n)$ -time algorithm for SET COVER contradicts the Strong Exponential Time Hypothesis (SETH), which states that if k -CNF SAT can be solved in $O^*(c_k^n)$ time, then $\lim_{k \rightarrow \infty} c_k = 2$. This conjecture is further supported by the fact that the number of solutions to an instance of SET COVER cannot be computed in $O((2 - \epsilon)^n)$ time for any $\epsilon > 0$ unless SETH fails [7]. A yet further 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 [3].

Theorem 6 *If MAXIMUM 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 instances of MAXIMUM GRAPH MOTIF restricted to one of the following two extreme cases:*

1. *each color may occur at most once, or*
2. *there are exactly two colors.*

1.7 Organization

Our two main lemmas, Lemma 2 and Lemma 3, are proved in Sect. 2. Theorem 4 is proved in Sect. 3. Theorem 5 is proved in Sect. 4. Theorem 6 is proved in Sect. 5.

2 Algebraic Tools

This section proves Lemma 2 and Lemma 3. We start with a proof of Lemma 1 that will act as a building block of both proofs.

2.1 Proof of Lemma 1.

It will be convenient to work with a polynomial consisting of a single monomial with coefficient 1, after which it will be easy to extend the analysis to an arbitrary polynomial. So suppose that

$$P(x_1, x_2, \dots, x_n) = x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$$

with $d_1 + d_2 + \cdots + d_n = k$. We must show that the expression $\sum_{A \subseteq L} P(z_1^A, z_2^A, \dots, z_n^A)$ is not identically zero in characteristic 2 if and only if $d_1, d_2, \dots, d_n \in \{0, 1\}$.

Let us start by simplifying the expression into a more convenient form. Recalling that $z_i^A = \sum_{j \in A} z_{i,j}$ for $i \in [n]$ and expanding the product–sum into a sum–product, we have

$$\begin{aligned} \sum_{A \subseteq L} P(z_1^A, z_2^A, \dots, z_n^A) &= \sum_{A \subseteq L} \prod_{i=1}^n \left(\sum_{j \in A} z_{i,j} \right)^{d_i} \\ &= \sum_{A \subseteq L} \prod_{i=1}^n \sum_{f_i: [d_i] \rightarrow A} \prod_{\ell=1}^{d_i} z_{i, f_i(\ell)} \\ &= \sum_{A \subseteq L} \sum_{f_1: [d_1] \rightarrow A} \sum_{f_2: [d_2] \rightarrow A} \cdots \sum_{f_n: [d_n] \rightarrow A} \prod_{i=1}^n \prod_{\ell=1}^{d_i} z_{i, f_i(\ell)}. \quad (5) \end{aligned}$$

The outer sum in (5) is over all subsets $A \subseteq L$ and the inner sums range over all n -tuples $f = (f_1, f_2, \dots, f_n)$ of functions $f_i: [d_i] \rightarrow A$ with $i \in [n]$.

Let us fix an arbitrary n -tuple $f = (f_1, f_2, \dots, f_n)$ of functions $f_i: [d_i] \rightarrow L$ with $i \in [n]$. Let us define the *image* of f by

$$I(f) = f_1([d_1]) \cup f_2([d_2]) \cup \cdots \cup f_n([d_n]).$$

Now let us consider the outer sum over subsets $A \subseteq L$ in (5). Observe that for a fixed $A \subseteq L$, our fixed n -tuple $f = (f_1, f_2, \dots, f_n)$ occurs exactly once in the inner sums of (5) if and only if $I(f) \subseteq A$. That is to say, the fixed f occurs exactly once for each A with $I(f) \subseteq A \subseteq L$. The number of such A is $2^{|L|-|I(f)|}$, which is even—and hence cancels in characteristic 2—if and only if $I(f) \neq L$.

Let us say that f is *surjective* if $I(f) = L$. Since all but surjective f cancel, from (5) and the previous analysis we thus have

$$\sum_{A \subseteq L} P(z_1^A, z_2^A, \dots, z_n^A) = \sum_{\substack{f=(f_1, f_2, \dots, f_n) \\ f \text{ surjective}}} \prod_{i=1}^n \prod_{\ell=1}^{d_i} z_{i, f_i(\ell)}. \quad (6)$$

Next we show that (6) is identically zero unless $d_1, d_2, \dots, d_n \in \{0, 1\}$.

So suppose there exists at least one bad index $b \in [n]$ with $d_b \geq 2$. Let us fix b to be the minimum such index. Consider an arbitrary surjective n -tuple $f = (f_1, f_2, \dots, f_n)$. Since $|L| = k = d_1 + d_2 + \cdots + d_n$ and f is surjective, we must have that for every $i \in [n]$ the function f_i is bijective, in particular thus $f_b(1) \neq f_b(2)$.

Define the *mate* f' of f by setting $f'_i = f_i$ for all $i \in [n] \setminus \{b\}$ and

$$f'_b(\ell) = \begin{cases} f_b(2) & \text{if } \ell = 1; \\ f_b(1) & \text{if } \ell = 2; \\ f_b(\ell) & \text{otherwise.} \end{cases}$$

Observe that $f' \neq f$ and that $f'' = f$. Thus, the set of all surjective f partitions into disjoint pairs $\{f, f'\}$ with

$$\prod_{i=1}^n \prod_{\ell=1}^{d_i} z_{i, f_i(\ell)} = \prod_{i=1}^n \prod_{\ell=1}^{d_i} z_{i, f'_i(\ell)}.$$

Thus, all monomials in (6) have an even coefficient and hence cancel in characteristic 2 unless $d_1, d_2, \dots, d_n \in \{0, 1\}$.

So suppose that $d_1, d_2, \dots, d_n \in \{0, 1\}$. Since $d_1 + d_2 + \cdots + d_n = k$, we have that the set $K = \{i \in [n] : d_i = 1\}$ has size k . Furthermore, associated with each surjective f there is a unique bijection $g : K \rightarrow L$ defined for all $i \in K$ by $g(i) = f_i(1)$. We thus have

$$\sum_{A \subseteq L} P(z_1^A, z_2^A, \dots, z_n^A) = \sum_{\substack{g: K \rightarrow L \\ g \text{ bijective}}} \prod_{i \in K} z_{i, g(i)}. \quad (7)$$

In particular, from each monomial $\prod_{i \in K} z_{i, g(i)}$ we can recover both the set K and the bijection $g : K \rightarrow L$, implying that no cancellation happens in characteristic 2. Furthermore, from K we can recover $P(x_1, x_2, \dots, x_n) = \prod_{i \in K} x_i$.

The lemma now follows by linearity. Indeed, an arbitrary multivariate polynomial $P(x_1, x_2, \dots, x_n)$ is a sum of monomials of the form $\alpha x_1^{d_1} x_2^{d_2} \cdots x_n^{d_n}$, where α is a coefficient from the considered field of characteristic two. \square

2.2 Proof of Lemma 2.

We obtain cancellation in characteristic 2 using identical arguments to the proof of Lemma 1, up to and including adapting (7) to the setting of Lemma 2. That is,

$$\sum_{A \subseteq L} P(u_1^A, u_2^A, \dots, u_n^A) = \sum_{\substack{g: K \rightarrow L \\ g \text{ bijective}}} \prod_{i \in K} u_{i, g(i)}. \quad (8)$$

We proceed to show that the right-hand side of (8) is not identically zero if and only if the multilinear monomial $\prod_{i \in K} x_i$ is properly colored.

Let us say that a function $h: K \rightarrow S$ that associates a shade $h(i) \in S$ to each $i \in K$ is *valid* if it holds that $h(i) \in S_{c(i)}$ for all $i \in K$. Observe in particular that an *injective* valid $h: K \rightarrow S$ exists if and only if $\prod_{i \in K} x_i$ is properly colored.

We are now ready to start simplifying the right-hand side of (8). Recalling that $u_{i,j} = \sum_{d \in S_{c(i)}} v_{i,d} w_{d,j}$, expanding the product–sum into a sum–product, and changing the order of summation, we have

$$\begin{aligned} \sum_{\substack{g: K \rightarrow L \\ g \text{ bijective}}} \prod_{i \in K} u_{i, g(i)} &= \sum_{\substack{g: K \rightarrow L \\ g \text{ bijective}}} \prod_{i \in K} \left(\sum_{d \in S_{c(i)}} v_{i,d} w_{d, g(i)} \right) \\ &= \sum_{\substack{g: K \rightarrow L \\ g \text{ bijective}}} \sum_{h: K \rightarrow S} \prod_{\substack{i \in K \\ h \text{ valid}}} v_{i, h(i)} w_{h(i), g(i)} \\ &= \sum_{\substack{h: K \rightarrow S \\ h \text{ valid}}} \sum_{\substack{g: K \rightarrow L \\ g \text{ bijective}}} \prod_{i \in K} v_{i, h(i)} w_{h(i), g(i)}. \end{aligned} \quad (9)$$

The outer sum in (9) ranges over all valid functions $h: K \rightarrow S$.

Now, let us fix an arbitrary valid $h: K \rightarrow S$. We will show that the inner sum in (9) evaluates to zero in characteristic 2 unless h is injective.

So suppose that h is not injective. In particular, there exists at least one pair $b_1, b_2 \in K$ with $h(b_1) = h(b_2)$ and $b_1 \neq b_2$. Let us fix (b_1, b_2) to be the lexicographically minimum such pair. Consider an arbitrary bijective $g: K \rightarrow L$. Define the *mate* g' of g by setting

$$g'(i) = \begin{cases} g(b_2) & \text{if } i = b_1; \\ g(b_1) & \text{if } i = b_2; \\ g(i) & \text{otherwise.} \end{cases}$$

Since g is bijective, we have $g' \neq g$ and $g'' = g$. Thus, the set of all bijections $g: K \rightarrow L$ partitions into disjoint pairs $\{g, g'\}$ with

$$\prod_{i \in K} v_{i,h(i)} w_{h(i),g(i)} = \prod_{i \in K} v_{i,h(i)} w_{h(i),g'(i)} .$$

Thus, for each valid $h : K \rightarrow S$ that is not injective, the monomials in the inner sum in (9) have an even coefficient and hence vanish in characteristic 2.

So suppose that h is injective (Recall that such an h exists if and only if K defines a properly colored multilinear monomial.) Let us study the inner sum in (9). Fix an arbitrary bijective $g : K \rightarrow L$ and study the inner monomial $\prod_{i \in K} v_{i,h(i)} w_{h(i),g(i)}$. From the variables $v_{i,d}$ in the monomial we can reconstruct the set K and the mapping h . Because h is injective, we can reconstruct the mapping g from the variables $w_{d,j}$ in the monomial by setting $g(h^{-1}(d)) = j$ for each relevant pair (d, j) . Since the three-tuple (K, h, g) can be reconstructed from the inner monomial, no cancellation happens in characteristic 2.

The lemma follows again by linearity. \square

2.3 Proof of Lemma 3

Let $\pi : S \rightarrow C$ be the mapping that projects each shade $d \in S_q$ to its underlying color $\pi(d) = q$. Imitating the proof of Lemma 1 and expanding (4) over $i \in K$ as in (9), we obtain cancellation in characteristic 2, except possibly for the monomials

$$\sum_{h:K \rightarrow S} \sum_{\substack{g:K \rightarrow L \\ g \text{ bijective}}} \eta^{\sum_{i \in K} \kappa_i(\pi(h(i)))} \prod_{i \in K} v_{i,h(i)} w_{h(i),g(i)} . \quad (10)$$

Imitating the proof of Lemma 2, we obtain further cancellation in characteristic 2 unless the mapping h is injective.

So suppose that h is injective. Observe that we can reconstruct the three-tuple (K, h, g) from the corresponding monomial in (10) exactly as in the proof of Lemma 2, and thus no further cancellation happens in characteristic 2. The degree of η is clearly the cost of the monomial $\prod_{i \in K} x_i$ in its coloring $c = \pi h$. In particular, we have that $\prod_{i \in K} x_i$ is properly colored in c since h is injective.

The lemma follows again by linearity. \square

2.4 Remarks

It is immediate from the proofs that the polynomial $P(\mathbf{x})$ may have additional variables $P(\mathbf{x}, \mathbf{y})$ without changing the conclusion as regards multilinearity and proper coloring of the monomials when restricted to the variables \mathbf{x} . Furthermore, any monomial that has total degree less than k in the variables \mathbf{x} will cancel.

We observe that Lemma 3 subsumes Lemma 2. Indeed, given a coloring $c : [n] \rightarrow C$ we can set the costs for Lemma 3 so that $\kappa_i(q) = 0$ if $c(i) = q$ and $\kappa_i(q) = 1$ otherwise. Then, $P(\mathbf{x})$ has at least one monomial that is both multilinear and properly colored if and only if $Q(\mathbf{v}, \mathbf{w}, \eta)$ has at least one monomial whose degree in the variable η is $\sigma = 0$.

3 An Algorithm for the Maximum Graph Motif Problem

This section illustrates the use of Lemma 2 in a concrete algorithm design for MAXIMUM GRAPH MOTIF. In particular, we proceed to give a proof of Theorem 4.

Consider an instance (H, M, C, c, k) of MAXIMUM GRAPH MOTIF. Let us write $m(q)$ for the number of occurrences of color $q \in C$ in the multiset M . Also recall that we assume that the host graph H is connected with n vertices and e edges; in particular, $e \geq n - 1$. By preprocessing we may assume that $m(q) \leq k$ for each $q \in C$.

Our first objective is to arrive at a generating polynomial $P_k(\mathbf{x}, \mathbf{y})$ that we can use with Lemma 2. There are two key aspects to this quest: (i) the multilinear monomials need to reflect the connected vertex sets of size k in H , and (ii) we must have a fast algorithm for evaluating the polynomial at specific points.

3.1 Branching Walks

The concept of branching walks was first introduced by Nederlof [19] to sieve for Steiner trees, followed by Guillemot and Sikora [13] who observed that branching walks can also be employed to span connected vertex sets of size k in the host graph H . Our approach here is to capitalize on this observation and span connected sets via branching walks.

Let us write $V = V(H) = \{1, 2, \dots, n\}$ for the vertex set and $E = E(H)$ for the edge set of the host graph H . A mapping $\varphi : V(T) \rightarrow V(H)$ is a *homomorphism* from a graph T to the host H if for all $\{a, b\} \in E(T)$ it holds that $\{\varphi(a), \varphi(b)\} \in E(H)$. We adopt the convention of calling the elements of $V(T)$ *nodes* and the elements of $V(H)$ *vertices*.

A *branching walk* in H is a pair $W = (T, \varphi)$ where T is an ordered rooted tree with node set $V(T) = \{1, 2, \dots, |V(T)|\}$ such that every node $a \in V(T)$ coincides with its rank in the preorder traversal of T , and $\varphi : V(T) \rightarrow V(H)$ is a homomorphism from T to H .

Let $W = (T, \varphi)$ be a branching walk in H . The walk *starts* from the vertex $\varphi(1)$ in H . The walk *spans* the vertices $\varphi(V(T))$ in H . The *size* of the walk is $|V(T)|$. The walk is *simple* if φ is injective. Finally, the walk is *properly ordered* if any two sibling nodes $a < b$ in T satisfy $\varphi(a) < \varphi(b)$ in H .

3.2 A Generating Polynomial for Branching Walks

We now define a generating polynomial for properly ordered branching walks of size k in H . Introduce a variable x_u for each vertex $u \in V(H)$ and two variables $y_{(u,v)}$ and $y_{(v,u)}$ for each edge $\{u, v\} \in E(H)$.

Let $W = (T, \varphi)$ be a properly ordered branching walk that starts from $s \in V(H)$ and has size k . Associate with W the *monomial fingerprint*

$$F(W, \mathbf{x}, \mathbf{y}) = \prod_{\substack{\{a,b\} \in E(T) \\ a < b}} y_{(\varphi(a), \varphi(b))} x_{\varphi(b)},$$

where the product is taken over all edges $\{a, b\} \in E(T)$.

Define the generating polynomial $P_{k,s}(\mathbf{x}, \mathbf{y})$ as the sum of the monomial fingerprints of the properly ordered branching walks that start from s and have size k . Let $P_k(\mathbf{x}, \mathbf{y}) = \sum_{s \in V(H)} x_s P_{k,s}(\mathbf{x}, \mathbf{y})$. Observe that all monomial in $P_k(\mathbf{x}, \mathbf{y})$ have total degree $2k - 1$.

Lemma 7 *A monomial in $P_k(\mathbf{x}, \mathbf{y})$ is multilinear in the variables \mathbf{x} if and only if it originates from a monomial fingerprint of a simple branching walk. Moreover, such a simple branching walk can be reconstructed from its monomial fingerprint.*

Proof For the first claim it suffices to consider an arbitrary monomial of $P_k(\mathbf{x}, \mathbf{y})$ and observe that the degree of the variable x_u indicates how many times $u \in V(H)$ occurs in the image of φ . In particular, φ is injective if and only if the monomial is multilinear in the variables \mathbf{x} .

For the second claim, let $W = (T, \varphi)$ be a simple and properly ordered branching walk that starts from s . We must reconstruct W from its monomial fingerprint that has been multiplied by x_s . Since φ is injective, we can immediately reconstruct (up to labels of the vertices) the rooted tree structure of T because the degrees of the variables $y_{(u,v)}$ in the monomial (if any) reveal both the edges and the orientation of each edge in T . Since W is properly ordered, we can reconstruct (up to labels of the vertices) the ordering of T . Finally, we can reconstruct the vertex labels of T by carrying out a preorder traversal of T .

An immediate corollary of Lemma 7 is that (H, M, C, c, k) is a YES-instance of MAXIMUM GRAPH MOTIF if and only if $P_k(\mathbf{x}, \mathbf{y})$ has a monomial that is both properly colored and multilinear in the sense of Lemma 2. Indeed, a multilinear monomial corresponds to a simple branching walk, which by definition spans a connected set of vertices. Conversely, every connected set of vertices admits at least one simple branching walk. Thus, to complete the proof of Theorem 4 it remains to derive a fast way to evaluate the polynomial $P_k(\mathbf{x}, \mathbf{y})$ and then apply Lemma 2 to obtain an algorithm design.

3.3 Evaluating the Generating Polynomial

This section develops a dynamic programming recurrence to evaluate the polynomial $P_k(\mathbf{x}, \mathbf{y})$ at a given assignment of values to the variables \mathbf{x}, \mathbf{y} .

For a vertex $u \in V(H)$, denote the ordered sequence of neighbors of u in H by $u_1 < u_2 < \dots < u_{\deg_H(u)}$.

For each $u \in V(H)$, $1 \leq i \leq \deg_H(u) + 1$, and $0 \leq \ell \leq k$, denote by $\mathcal{W}(\ell, u, i)$ the set of properly ordered branching walks $W = (T, \varphi)$ such that (i) the size of W is ℓ , (ii) W starts from u , and (iii) for any child node a of 1 in T it holds that $\varphi(a) = u_j$ implies $j \geq i$. Define the associated generating polynomial over the variables \mathbf{x}, \mathbf{y} by

$$P_{\ell,u,i}(\mathbf{x}, \mathbf{y}) = \sum_{(T,\varphi) \in \mathcal{W}(u,1,\ell)} \prod_{\substack{\{a,b\} \in E(T) \\ a < b}} y_{(\varphi(a), \varphi(b))} x_{\varphi(b)}.$$

It is immediate from the definition that $P_{\ell,u}(\mathbf{x}, \mathbf{y}) = P_{\ell,u,1}(\mathbf{x}, \mathbf{y})$.

The functions $P_{\ell,u,i}(\mathbf{x}, \mathbf{y})$ admit the following recurrence. The base case occurs for $\ell = 1$ or $i = \deg_H(u) + 1$, in which case we have

$$P_{\ell,u,i}(\mathbf{x}, \mathbf{y}) = \begin{cases} 1 & \text{if } \ell = 1, \\ 0 & \text{otherwise.} \end{cases} \quad (11)$$

For $2 \leq \ell \leq k$ and $1 \leq i \leq \deg_H(u)$, we have

$$P_{\ell,u,i}(\mathbf{x}, \mathbf{y}) = P_{\ell,u,i+1}(\mathbf{x}, \mathbf{y}) + y_{(u,u_i)x_{u_i}} \sum_{\substack{\ell_1+\ell_2=\ell \\ \ell_1, \ell_2 \geq 1}} P_{\ell_1,u,i+1}(\mathbf{x}, \mathbf{y}) \cdot P_{\ell_2,u_i,1}(\mathbf{x}, \mathbf{y}). \quad (12)$$

To see that the recurrence is correct, observe that the two lines above in (12) correspond to a partitioning of the properly ordered branching walks in $\mathcal{W}(\ell, u, i)$ into two disjoint classes where either (i) there is no child node a of 1 in T such that $h(a) = u_i$ or (ii) there is a unique such child (At most one such child may exist because the branching walk is properly ordered.)

Thus, we can evaluate the polynomial $P_k(\mathbf{x}, \mathbf{y})$ via (11), (12), and

$$P_k(\mathbf{x}, \mathbf{y}) = \sum_{u \in V(H)} x_u P_{k,u,1}(\mathbf{x}, \mathbf{y}). \quad (13)$$

3.4 The Algorithm

We are now ready to describe the algorithm for Theorem 4. Assume an instance (H, M, C, c, k) of the MAXIMUM GRAPH MOTIF has been given as input.

Let $b = \lceil \log_2 6k \rceil$ and consider the finite field \mathbb{F}_{2^b} of order 2^b . Introduce variables $v_{i,d}$ and $w_{d,j}$ as in the setup of Lemma 2. Assign a value from \mathbb{F}_{2^b} uniformly and independently at random to each of these variables. Similarly, as in the setup of Sect. 3.2, introduce two variables $y_{(r,s)}$ and $y_{(s,r)}$ to each edge $\{r, s\} \in E(H)$ and assign a value to each variable uniformly and independently at random from \mathbb{F}_{2^b} . We thus have three vectors of values in \mathbb{F}_{2^b} , namely \mathbf{v} , \mathbf{w} , and \mathbf{y} .

Using the recurrence given by (11), (12), and (13) for each $A \subseteq L$ in turn, compute the value

$$Q(\mathbf{v}, \mathbf{w}, \mathbf{y}) = \sum_{A \subseteq L} P_k(\mathbf{u}^A(\mathbf{v}, \mathbf{w}), \mathbf{y}), \quad (14)$$

where the values $\mathbf{u}^A(\mathbf{v}, \mathbf{w}) = (u_1^A(\mathbf{v}, \mathbf{w}), u_2^A(\mathbf{v}, \mathbf{w}), \dots, u_n^A(\mathbf{v}, \mathbf{w}))$ are determined from the set A and the values \mathbf{v} and \mathbf{w} as in Lemma 2. If $Q(\mathbf{v}, \mathbf{w}, \mathbf{y})$ is nonzero in \mathbb{F}_{2^b} , output YES; otherwise output NO. This completes the description of the algorithm.

3.5 Running Time

To analyse the running time of the algorithm, observe that we can assume that $m(q) \leq k$. Thus, computing the values $\mathbf{u}^A(\mathbf{v}, \mathbf{w})$ for a fixed $A \subseteq L$ takes $O(k^2n)$ arithmetic

operations in \mathbb{F}_{2^b} , and each such operation can be implemented to run in time $\mu = O(b \log b \log \log b)$ [6]. Furthermore, each evaluation of (11), (12), and (13) for a fixed A takes $O(k^2 e)$ arithmetic operations in \mathbb{F}_{2^b} . Hence, recalling that $e \geq n - 1$, the total running time of the algorithm is $O(2^k k^2 e \mu)$.

3.6 Correctness

To establish the desired properties of the algorithm, observe that from Sect. 3.2 and Lemma 2 it follows that (14) —viewed as a polynomial in the variables \mathbf{v} , \mathbf{w} , and \mathbf{y} — is not identically zero if and only if (H, M, C, c, k) is a YES-instance of MAXIMUM GRAPH MOTIF. Thus, if (H, M, C, c, k) is a NO-instance, then (14) evaluates to zero and the algorithm gives a NO output. Furthermore, if (H, M, C, c, k) is a YES-instance, then (14) is an evaluation of a nonzero multivariate polynomial of total degree $3k - 1$ at a point $(\mathbf{v}, \mathbf{w}, \mathbf{y})$ selected uniformly at random. Recalling that $2^b \geq 6k$, the following lemma thus implies that the value $Q(\mathbf{v}, \mathbf{w}, \mathbf{y})$ is nonzero (and hence the algorithm outputs YES) with probability at least $1/2$.

Lemma 8 ([8, 20, 22]) *A nonzero polynomial $P(z_1, z_2, \dots, z_\ell)$ of total degree d with coefficients in the finite field \mathbb{F}_q has at most $dq^{\ell-1}$ roots in \mathbb{F}_q^ℓ .*

This completes the proof of Theorem 4. □

3.7 Minor Variants and Extensions

The basic framework presented above immediately allows for some minor variants and extensions, such as seeking an exact match instead of the maximum match by setting $|M| = k$. Similarly, one may extend from a fixed coloring $c : V(H) \rightarrow C$ into a *list coloring* version where each vertex $i \in V(H)$ gets associated a list $C(i) \subseteq C$ of valid colors instead of a single color $c(i)$, and the motif M may match against any one of the colors in the list. This variant can be implemented by simply changing the inner sum in Lemma 2 to $u_{i,j} = \sum_{d \in \cup_{q \in C(i)} S_q} v_{i,d} w_{d,j}$. That is, we sum over the shades of all the colors q in $C(i)$.

4 An Algorithm for the Closest Graph Motif Problem

This section gives a proof of Theorem 5 using Lemma 3 and the generating function developed in Sect. 3.2.

Consider an instance $(H, M, C_0, c, \sigma_S, \sigma_I, \sigma_D, \tau, k)$ of CLOSEST GRAPH MOTIF with $V(H) = \{1, 2, \dots, n\}$. Let us again write $m(q)$ for the number of occurrences of color $q \in C_0$ in the multiset M . We may assume that $m(q) \leq k$. Furthermore, since H is connected, the number of vertices n and the number of edges e satisfy $e \geq n - 1$.

The key step in arriving at Theorem 5 is to transport weighted edit distance into the setting of Lemma 3.

4.1 Optimum Edit Sequences

It will be convenient to have available the following lemma that characterizes the structure of a sequence of operations that realizes the minimum cost to transform a multiset M to the multiset N , where both multisets are over C_0 .

Let $k = |N|$. Consider an arbitrary sequence of basic operations that transforms M to N . As the sequence is executed, each original element of M gets assigned into one of three classes. First, there are k_U elements in M that remain untouched (and hence in N) when the execution terminates. Second, there are k_S elements in M that undergo at least one substitution—which we may view as “recoloring” of the element—and remain in N when the execution terminates. Third, the remaining $|M| - k_U - k_S$ elements of M get deleted during execution. Thus, at least $k - k_U - k_S$ insertions must occur in the sequence. Let us call the values k_U and k_S the *parameters* of the sequence.

Lemma 9 *Let there exist at least one sequence with parameters k_U and k_S that transforms M into N . Then, the cost of this sequence is at least*

$$\sigma_S k_S + \sigma_D (|M| - k_U - k_S) + \sigma_I (k - k_U - k_S), \quad (15)$$

with equality for at least one sequence that transforms M into N .

Proof The inequality is immediate from the preceding analysis; the sequence that meets equality (i) does nothing for the k_U untouched original elements, (ii) substitutes the correct final color with one substitution for each of the k_S originals, (iii) deletes each of the $|M| - k_U - k_S$ remaining originals, and (iv) finally inserts $k - k_U - k_S$ new elements to match with N .

Lemma 9 reveals a useful symmetry between insertions and deletions in an optimum sequence; that is, if we let $k_{ID} = k - k_U - k_S$, then (15) is equal to

$$\sigma_S k_S + (\sigma_I + \sigma_D) k_{ID} + \sigma_D (|M| - k). \quad (16)$$

Thus it suffices to optimize over k -multisets of colors while tracking the parameters k_S and k_{ID} to arrive at the optimum. This strategy will be employed in our algorithm.

4.2 The Algorithm

Assume an instance $(H, M, C_0, c, \sigma_S, \sigma_I, \sigma_D, \tau, k)$ of CLOSEST GRAPH MOTIF has been given as input.

Let us first set up the application of Lemma 3. Introduce a new color “*” and let $C = C_0 \cup \{*\}$ with $m(*) = k$. As already highlighted in the remarks to Lemma 3, instead of one indeterminate η , we will work with two indeterminates η_S and η_{ID} in Lemma 3 to simultaneously track the S-cost and the ID-cost. For $i \in [n]$ and $q \in C$, define the cost functions

$$\kappa_i^S(q) = \begin{cases} 0 & \text{if } q = c(i); \\ 1 & \text{if } q \neq c(i) \text{ and } q \in C_0; \\ 0 & \text{if } q = * \end{cases} \quad (17)$$

and

$$\kappa_i^{\text{ID}}(q) = \begin{cases} 0 & \text{if } q = c(i); \\ 0 & \text{if } q \neq c(i) \text{ and } q \in C_0; \\ 1 & \text{if } q = *. \end{cases} \quad (18)$$

The intuition underlying (17) and (18) is as follows. Coloring a vertex i with color $q \notin \{c(i), *\}$ corresponds to substitution of a copy of q in M by a copy of $c(i)$. Coloring i with color “*” corresponds to inserting a copy of $c(i)$ to M .

The algorithm now proceeds as follows. Let $b = \lceil \log_2 6k \rceil$ and consider the finite field \mathbb{F}_{2^b} of order 2^b . Introduce variables $v_{i,d}$ and $w_{d,j}$ as in the setup of Lemma 3. Assign a value from \mathbb{F}_{2^b} uniformly and independently at random to each of these variables. Similarly, as in the setup of Sect. 3.2, introduce two variables $y_{(r,s)}$ and $y_{(s,r)}$ to each edge $\{r, s\} \in E(H)$ and assign a value to each variable uniformly and independently at random from \mathbb{F}_{2^b} . We thus have three vectors of values in \mathbb{F}_{2^b} , namely \mathbf{v} , \mathbf{w} , and \mathbf{y} .

The main part of the algorithm consists of two outer loops that cycle through $k + 1$ distinct values in \mathbb{F}_{2^b} to each of the variables η_S and η_{ID} . For each pair of values $(\eta_S, \eta_{\text{ID}})$ in \mathbb{F}_{2^b} , we use the recurrence given by (11), (12), and (13) for each $A \subseteq L$ in turn, and compute the value

$$\mathcal{Q}(\mathbf{v}, \mathbf{w}, \mathbf{y}, \eta_S, \eta_{\text{ID}}) = \sum_{A \subseteq L} P_k(\mathbf{u}^A(\mathbf{v}, \mathbf{w}, \eta_S, \eta_{\text{ID}}), \mathbf{y}), \quad (19)$$

where the values

$$\mathbf{u}^A(\mathbf{v}, \mathbf{w}, \eta_S, \eta_{\text{ID}}) = (u_1^A(\mathbf{v}, \mathbf{w}, \eta_S, \eta_{\text{ID}}), u_2^A(\mathbf{v}, \mathbf{w}, \eta_S, \eta_{\text{ID}}), \dots, u_n^A(\mathbf{v}, \mathbf{w}, \eta_S, \eta_{\text{ID}}))$$

are determined from the set A and the values \mathbf{v} and \mathbf{w} as in Lemma 3, but with (4) replaced by

$$u_{i,j} = \sum_{q \in C} \eta_S^{\kappa_i^S(q)} \eta_{\text{ID}}^{\kappa_i^{\text{ID}}(q)} \sum_{d \in S_q} v_{i,d} w_{d,j} \quad (20)$$

for all $i \in [n]$, $j \in L$, and $A \subseteq L$. When the main part terminates, we have available $(k + 1)^2$ evaluations of (19) at points $(\eta_S, \eta_{\text{ID}})$.

By Lagrange interpolation, we recover (19) as a bivariate polynomial of total degree at most k in the indeterminates η_S and η_{ID} . If this bivariate polynomial has at least one monomial $\eta_S^{k_S} \eta_{\text{ID}}^{k_{\text{ID}}}$ such that the degrees k_S and k_{ID} satisfy

$$\sigma_S k_S + (\sigma_1 + \sigma_D) k_{\text{ID}} + \sigma_D(|M| - k) \leq \tau, \quad (21)$$

then the algorithm outputs YES; otherwise the algorithm outputs NO. This completes the description of the algorithm.

4.3 Running Time

The analysis is essentially similar to Sect. 3.5, with two differences. First, the outer loop in the main part introduces a multiplicative factor k^2 compared with Sect. 3.5. Second, the implementation of (20) requires us to sum over all the shades originating from M and the k shades of the color “*”. This can be done efficiently by precomputing the inner sums $\sum_{d \in S_q} v_{i,d} w_{d,j}$ for each color $q \in C$, index $i \in [n]$, and label $j \in L$, which takes $O((|M| + k)kn\mu)$ time outside the main loops. In the outer loop of the main part it thus suffices to compute only the outer sum in (20) for each choice of (η_S, η_{ID}) , which leads to $O(|C_0|kn\mu)$ time for each iteration of the outer loop. In the inner loop over $A \subseteq L$, it takes $O(kn\mu)$ time to prepare the vector $\mathbf{u}^A(\mathbf{v}, \mathbf{w}, \eta_S, \eta_{ID})$. Compared with Sect. 3.5, this gives a further contributing factor of $|C_0|k$ outside the inner loop (The running time cost of the final interpolation step and the checking of the at most k^2 monomials of the bivariate polynomial $Q(\mathbf{v}, \mathbf{w}, \mathbf{y}, \eta_S, \eta_{ID})$ with respect to (21) is assumed to be subsumed by the running time bound.)

4.4 Correctness

We start by observing that (17) and (18) imply that (19) has total degree at most k in the variables η_S and η_{ID} , thus implying that Lagrange interpolation will correctly recover the polynomial in η_S and η_{ID} from the $(k + 1)^2$ evaluations computed in the main loop.

Let us say that (19)—viewed as a polynomial in all the variables $\mathbf{v}, \mathbf{w}, \mathbf{y}, \eta_S, \eta_{ID}$ —is *witnessing* if there exists at least one monomial whose degrees k_S and k_{ID} satisfy (21).

Lemma 10 *The polynomial (19) is witnessing if and only if the given input is a YES-instance of CLOSEST GRAPH MOTIF.*

Proof In the “only if” direction, consider a monomial of (19) whose degrees k_S and k_{ID} satisfy (21). From Lemma 3 we have that the polynomial $P_k(\mathbf{x}, \mathbf{y})$ has at least one monomial that is both multilinear in \mathbf{x} and admits a proper coloring with S-cost k_S and ID-cost k_{ID} . From Sect. 3.2 it follows that this monomial of $P_k(\mathbf{x}, \mathbf{y})$ corresponds to a simple branching walk in H and thus identifies a connected set $K \subseteq V(H)$ of vertices in H . Furthermore, the existence of a proper coloring of the monomial implies by (17), (18), and Lemma 9 that there exists a sequence of basic operations that transforms the multiset M to the multiset $c(K)$ with total cost (16). In particular, since k_S and k_{ID} satisfy (21), we have that $(H, M, C_0, c, \sigma_S, \sigma_I, \sigma_D, \tau, k)$ is a YES-instance of CLOSEST GRAPH MOTIF.

In the “if” direction, let $(H, M, C_0, c, \sigma_S, \sigma_I, \sigma_D, \tau, k)$ be a YES-instance of CLOSEST GRAPH MOTIF. Let $K \subseteq V(H)$ be a solution set and consider an associated sequence Δ of basic operations that transforms M to $c(K)$ with cost at most τ . We may without loss of generality assume that the cost of the sequence Δ satisfies equality in Lemma 9. In particular, from (16) we thus observe that the parameters k_S and k_{ID} of the sequence Δ thus satisfy (21). Consider a simple branching walk of size k in H that spans the vertices in K . From Sect. 3.2 we observe that there is a corresponding

multilinear monomial in $P_k(\mathbf{x}, \mathbf{y})$. Next observe that we can properly color this monomial in the sense of Lemma 3 by (i) assigning the color $*$ to each of the k_{ID} values $i \in K$ that correspond to elements inserted in Δ , (ii) assigning the substituted color to each of the k_S values $i \in K$ that correspond to elements of M receiving substitutions in Δ , and (iii) assigning the color $c(i)$ to each of the remaining $k - k_S - k_{\text{ID}}$ values $i \in K$ that correspond to elements of M that are not touched by Δ . Furthermore, by (17) and (18), this proper coloring has S-cost k_S and ID-cost k_{ID} . From Lemma 3 we thus have that (19)—viewed as a polynomial in the variables $\mathbf{v}, \mathbf{w}, \mathbf{y}, \eta_S, \eta_{\text{ID}}$ —has at least one monomial whose degrees k_S and k_{ID} satisfy (21).

Let us now study the operation of the algorithm in more detail. We have that the given input is a NO-instance if and only if (19) is not witnessing. Thus, given a NO-instance as input, the algorithm always gives a NO output.

So suppose that the given input is a YES-instance. Since (19) is witnessing, there exist degrees k_S and k_{ID} that are present in a monomial of (19) such that (21) holds. In particular, coefficient of the monomial $\eta_S^{k_S} \eta_{\text{ID}}^{k_{\text{ID}}}$ computed by the algorithm is an evaluation of a nonzero multivariate polynomial of total degree $3k - 1$ at a point $(\mathbf{v}, \mathbf{w}, \mathbf{y})$ selected uniformly at random. Recalling that $2^b \geq 6k$, Lemma 8 thus implies that the coefficient is nonzero (and hence the algorithm outputs YES) with probability at least $1/2$. This completes the proof of Theorem 5. \square

5 A Lower Bound Reduction from Set Cover

We base our proof of Theorem 6 on the following theorem, which can be extracted from the proof of Theorem 4.4 in a recent paper by Cygan et al. [7].

Theorem 11 ([7]) *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 $\epsilon' > 0$.*

5.1 Proof of Theorem 6

Let (\mathcal{S}, t) be an instance of SET COVER. We are going to show a polynomial-time reduction to MAXIMUM GRAPH MOTIF so that in the resulting instance (H, C, m, c, k) we have $\sum_{q \in C} m(q) = k = n + t + 1$. Combined with Theorem 11, this reduction will immediately establish our claim.

The graph H is defined as follows. The vertex set consists of the universe U , t copies of the family \mathcal{S} , and a special vertex r , that is, $V(H) = U \cup \{s_i^j : i = 1, 2, \dots, m, j = 1, 2, \dots, t\} \cup \{r\}$. The edge set is $E(H) = \{\{a, s_i^j\} : a \in S_i\} \cup \{\{r, s_i^j\} : i = 1, 2, \dots, m, j = 1, 2, \dots, t\}$. Let $k = n + t + 1$.

To establish part (1), let $C = \{1, 2, \dots, n + t + 1\}$ with $m(q) = 1$ for each $q \in C$. Furthermore, assign the colors to vertices so that $c(s_i^j) = j$ for every $i = 1, 2, \dots, m, j = 1, 2, \dots, t$ and $c(r) = t + 1$. Finally, assign the n colors $t + 2, t + 3, \dots, n + t + 1$ bijectively to the vertices in U .

We show that (\mathcal{S}, t) is a YES-instance if and only if (H, C, m, c, k) is a YES-instance. To establish the “only if” direction, suppose that $S_{i_1}, S_{i_2}, \dots, S_{i_t}$ is a solution

of (S, t) . Then let $K = \{r\} \cup U \cup \{s_{ij}^j : j = 1, 2, \dots, t\}$. It is clear that $c(K) = C$ and that $H[\{r\} \cup \{s_{ij}^j : j = 1, 2, \dots, t\}]$ is connected. Since for every $a \in U$ there is $j = 1, 2, \dots, t$ such that $a \in S_{ij}$, so $\{a, s_{ij}^j\} \in E(G[K])$. It follows that $G[K]$ is connected, and hence K is a solution of (H, C, m, c, K) . To establish the “if” direction, suppose that K is a solution of (H, C, m, c, k) . Then for every $j = 1, 2, \dots, t$ there is exactly one $i_j \in \{1, 2, \dots, m\}$ such that $s_{i_j j}^j \in K$, since $c(K) = C$. Moreover, since $G[K]$ is connected we observe that for every $a \in U$ there is a $j = 1, 2, \dots, t$ such that $\{a, s_{i_j j}^j\} \in E(G[K])$. But then $a \in S_{i_j}$ and it follows that $S_{i_1}, S_{i_2}, \dots, S_{i_t}$ is a solution of (S, t) .

To establish part (2), let $C = \{1, 2\}$ with $m(1) = n + 1$ and $m(2) = t$. Set $c(r) = 1$ and $c(a) = 1$ for every $a \in U$. All the remaining vertices are colored with 2. The proof of equivalence is similar to part (1) and is left to the reader. \square

Acknowledgments A preliminary conference abstract of this work has appeared as [5]. This research was supported in part by the Swedish Research Council, Grant VR 2012-4730 (A. B.), the Academy of Finland, Grants 252083 and 256287 (P. K.), and by the National Science Centre of Poland, Grants N206 567140 and UMO-2013/09/B/ST6/03136 (Ł. K.). The third author thanks Sylwia Antoniuk, Marek Cygan, Michał Debski, and Matthias Mnich for helpful discussions on related topics.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Betzler, N., Fellows, M.R., Komusiewicz, C., Niedermeier, R.: Parameterized algorithms and hardness results for some graph motif problems. In: Proceedings of CPM'08. LNCS, vol. 5029, pp. 31–43 (2008)
2. Björklund, A.: Determinant sums for undirected hamiltonicity. In: Proceedings of the FOCS'10, pp. 173–182 (2010)
3. Björklund, A.: Counting perfect matchings as fast as Ryser. In: Proceedings of the SODA'12, pp. 914–921 (2012)
4. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Narrow sieves for parameterized paths and packings. CoRR. [abs/1007.1161](https://arxiv.org/abs/1007.1161) (2010)
5. Björklund, A., Kaski, P., Kowalik, L.: Probably optimal graph motifs. In: Portier, N., Wilke, T. (eds.) STACS. LIPIcs, vol. 20, pp. 20–31. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Wadern (2013)
6. Bürgisser, P., Clausen, M., Shokrollahi, M.A.: Algebraic Complexity Theory, Grundlehren der mathematischen Wissenschaften, vol. 315. Springer, New York (1997)
7. Cygan, M., Dell, H., Lokshtanov, D., Marx, D., Nederlof, J., Okamoto, Y., Paturi, R., Saurabh, S., Wahlström, M.: On problems as hard as CNF-SAT. In: IEEE Conference on Computational Complexity, pp. 74–84 (2012)
8. DeMillo, R.A., Lipton, R.J.: A probabilistic remark on algebraic program testing. Inf. Process. Lett. 7, 193–195 (1978)
9. Dondi, R., Fertin, G., Viallette, S.: Maximum motif problem in vertex-colored graphs. In: Proceedings of the CPM'09. LNCS, vol. 5577, pp. 221–235 (2009)
10. Dondi, R., Fertin, G., Viallette, S.: Finding approximate and constrained motifs in graphs. In: Proceedings of the CPM'11. LNCS, vol. 6661, pp. 388–401 (2011)
11. Fellows, M.R., Fertin, G., Hermelin, D., Viallette, S.: Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In: Proceedings of the ICALP'07. LNCS, vol. 4596, pp. 340–351 (2007)

12. Fellows, M.R., Fertin, G., Hermelin, D., Vialette, S.: Upper and lower bounds for finding connected motifs in vertex-colored graphs. *J. Comput. Syst. Sci.* **77**(4), 799–811 (2011)
13. Guillemot, S., Sikora, F.: Finding and counting vertex-colored subtrees. In: *Proceedings of the MFCS'10*. LNCS, vol. 6281, pp. 405–416 (2010)
14. Koutis, I.: Faster algebraic algorithms for path and packing problems. In: *Proceedings of the ICALP'08*. LNCS, vol. 5125, pp. 575–586 (2008)
15. Koutis, I.: The power of group algebras for constrained multilinear monomial detection. In: *Dagstuhl meeting 10441* (2010)
16. Koutis, I.: Constrained multilinear detection for faster functional motif discovery. *Inf. Process. Lett.* **112**(22), 889–892 (2012)
17. Koutis, I., Williams, R.: Limits and applications of group algebras for parameterized problems. In: *ICALP (1)*. LNCS, vol. 5555, pp. 653–664 (2009)
18. Lacroix, V., Fernandes, C.G., Sagot, M.F.: Motif search in graphs: application to metabolic networks. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **3**(4), 360–368 (2006)
19. Nederlof, J.: Fast polynomial-space algorithms using Möbius inversion: improving on Steiner tree and related problems. In: *Proceedings of the ICALP'09*. LNCS, vol. 5555, pp. 713–725 (2009)
20. Schwartz, J.T.: Fast probabilistic algorithms for verification of polynomial identities. *J. ACM* **27**(4), 701–717 (1980)
21. Williams, R.: Finding paths of length k in $O^*(2^k)$ time. *Inf. Process. Lett.* **109**(6), 315–318 (2009)
22. Zippel, R.: Probabilistic algorithms for sparse polynomials. In: *Proceedings of the International Symposium on Symbolic and Algebraic Computation*. LNCS, vol. 72, pp. 216–226 (1979)