# Deterministic Parameterized Algorithms for the Graph Motif Problem

Ron Y. Pinter, Hadas Shachnai, and Meirav Zehavi

Department of Computer Science, Technion, Haifa 32000, Israel
{pinter,hadas,meizeh}@cs.technion.ac.il

**Abstract.** We study the classic Graph Motif problem: given a graph $G = (V, E)$ with a set of colors for each node, and a multiset $M$ of colors, we seek a subtree $T \subseteq G$, and a coloring of the nodes in $T$, such that $T$ carries exactly (also with respect to multiplicity) the colors in $M$. Graph Motif plays a central role in the study of pattern matching problems, primarily motivated from the analysis of complex biological networks.

Previous algorithms for Graph Motif and its variants either rely on techniques for developing randomized algorithms that, if derandomized, render them inefficient, or the algebraic narrow sieves technique for which there is no known derandomization. In this paper, we present fast *deterministic* parameterized algorithms for Graph Motif and its variants. Specifically, we give such an algorithm for the more general Graph Motif with Deletions problem, followed by faster algorithms for Graph Motif and other well-studied special cases. Our algorithms make non-trivial use of *representative families*, and a novel tool that we call *guiding trees*, together enabling the efficient construction of the output tree.

## 1 Introduction

With the advent of network biology and complex network analysis in general, the study of pattern matching problems in graphs has become of major importance [12,16]. Indeed, the term "graph motif" plays a central role in this context, with different node colors used to model different functionalities of the network (see, e.g., [17,7]). Due to the generic nature of the Graph Motif (GM) problem (also known as the Topology-Free Network Query problem), the so called *motif analysis* approach has become useful also in the study of social networks (see, e.g., [23] and the references therein).

The GM problem is a natural variant of classic pattern matching problems, where the topology of the pattern $M$ is unknown or of lesser importance. Given a graph $G = (V, E)$ with a set of colors for each node, and a multiset $M$ of colors, we seek a subtree $T \subseteq G$, and a coloring of the nodes in $T$, such that $T$ carries exactly (also with respect to multiplicity) the colors in $M$. We call $T$ an *occurrence* of $M$ in $G$. To allow more flexibility in the definition of an occurrence, and since biological network data often contains noise, a generalized version of GM allows *deleting* colors from $M$.

Parameterized algorithms solve NP-hard problems by confining the combinatorial explosion to a parameter $k$. More precisely, a problem is *fixed-parameter*

E. Csuhaj-Varjú et al. (Eds.): MFCS 2014, Part II, LNCS 8635, pp. 589–600, 2014.

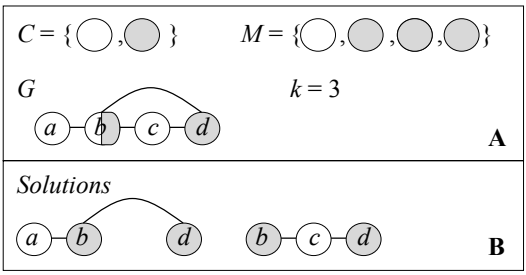


**Fig. 1.** An input for $\mathrm{GM_D}$ (A), and two possible solutions (B)

*tractable (FPT)* with respect to a parameter $k$ if it can be solved in time $O^*(f(k))$ for some function $f$, where $O^*$ hides factors polynomial in the input size. Since GM is NP-complete [17], there is a growing body of literature studying its parameterized complexity (see the excellent survey in [26]). In this paper, we present fast *deterministic* parameterized algorithms for GM and its variants.

### 1.1 Problem Statement

The most general variant considered in this paper is Graph Motif with Deletions ($\mathrm{GM_D}$): the input is a set of colors $C$, a multiset $M$ of colors from $C$, and an undirected graph $G = (V, E)$. The nodes in $V$ are associated with colors via a (set-)coloring $Col : V \to 2^C$. We are also given a parameter $k \leq |M|$.

We need to decide if there exists a subtree $T = (V_T, E_T)$ of $G$ on $k$ nodes, and a coloring $col : V_T \to C$ that assigns a color from $Col(v)$ to each node $v \in V_T$, such that

$$\forall c \in C : |\{v \in V_T : col(v) = c\}| \leq occ(c), \tag{1}$$

where $occ(c)$ is the number of occurrences of a color $c$ in $M$ (see Fig. 1).[1]

**Special Cases:** Restricted $\mathrm{GM_D}$ ($\mathrm{RGM_D}$) is the special case of $\mathrm{GM_D}$ where for any node $v \in V$, $|Col(v)| = 1$. Also, GM and RGM are the special cases of $\mathrm{GM_D}$ and $\mathrm{RGM_D}$, respectively, where deletions are not allowed (i.e., the inequality in (1) is replaced by equality, and $k = |M|$).

### 1.2 Known Results and Our Contribution

$\mathrm{GM_D}$ has received considerable attention since it was introduced by Lacroix et al. [17]. The paper [17] also shows that RGM is NP-hard when $M$ is a set and $G$ is a tree. Even seemingly simpler cases of RGM are known to be NP-hard (see [11,2,8]). Moreover, a natural optimization version of $\mathrm{RGM_D}$, minimizing the number of deletions from $M$, is hard to approximate within factor $|V|^{\frac{1}{3}-\epsilon}$ [24].

[1] Some papers define $\mathrm{GM_D}$ as a problem where one seeks a connected subgraph $S$ of $G$, which is equivalent to our definition (simply consider some spanning tree $T$ of $S$).

On the positive side, using techniques for developing randomized parameterized algorithms, many such algorithms have been obtained for $\mathrm{GM_D}$ and its variants [3,4,6,7,9,10,14,15,21,22]. Some of these algorithms can be derandomized, resulting, however, in inefficient algorithms. In particular, Fellows et al. [10] gave a deterministic algorithm for RGM that runs in time $O^*(87^k)$, based on a derandomization of the color coding technique [1]. Currently, the best randomized algorithm for $\mathrm{GM_D}$ runs in time $O^*(2^k)$, due to Björklund et al. [6]. This algorithm is based on the narrow sieves technique [5], for which there is no known derandomization. Thus, previous studies left open the existence of a *fast* deterministic parameterized algorithm for $\mathrm{GM_D}$.

In this paper, we present fast deterministic parameterized algorithms for $\mathrm{GM_D}$ and its variants. In particular, we develop an $O^*(6.86^k)$ time algorithm for $\mathrm{GM_D}$, an $O^*(5.22^k)$ time algorithm for GM, and an $O^*(5.18^k)$ time algorithm for $\mathrm{RGM_D}$.

Due to space constraints, some of the proofs are omitted. The detailed results appear in [20].

### 1.3 Techniques

Our algorithms make non-trivial use of *representative families*, and a novel tool that we call *guiding trees*, together enabling the efficient construction of the output tree. Informally, a guiding tree is a constant-size rooted tree which provides some structural information about the solution tree. To efficiently compute a family $\mathcal{S}$ of partial solutions, we first construct a polynomial number of *suitable* guiding trees. We then use these trees to generate $\mathcal{S}$, by combining previously computed families of partial solutions. Thus, we avoid iterating over all $O^*(2^k)$ possible topologies for the solution tree.

The efficiency of our algorithms is further improved via replacement of each family of partial solutions, $\mathcal{S}$, by a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$, which represents $\mathcal{S}$. Each representative family $\widehat{\mathcal{S}}$ contains enough sets from $\mathcal{S}$, thus, we preserve the correctness of the algorithm while improving its running time.

Building on the powerful technique of Fomin et al. [13], for efficient construction of representative families, we tailor the definitions of these sets to the problem at hand. This also leads to replacing *uniform* matroid (often used for fast computation of representative families) by *partition* matroid, which captures more closely the restricted variants of GM.

## 2 Preliminaries

Given a graph $H$, let $V_H$ and $E_H$ denote its node-set and edge-set, respectively.

**Matroids:** In deriving our results, we use two types of matroids.[2] Given a constant $k$, the first is defined by a pair $M = (E, \mathcal{I})$, where $E$ is an $n$-element set, and $\mathcal{I} = \{S \subseteq E : |S| \leq k\}$. Such a pair is called a *uniform matroid*, denoted by $U_{n,k}$.

[2] For a broader overview of matroids, see, e.g., [19].

Given some constants $\ell$ and $k_1, k_2, \ldots, k_\ell$, the second is defined by a pair $(E, \mathcal{I})$, where $E$ is an $n$-element set partitioned into disjoint sets $E_1, E_2, \ldots, E_\ell$, and $\mathcal{I} = \{S \subseteq E : |S \cap E_1| \leq k_1, |S \cap E_2| \leq k_2, \ldots, |S \cap E_\ell| \leq k_\ell\}$. Such a pair is called a *partition matroid*. Note that, when $\ell = 1$, the definitions for the two types of matroids coincide.

**Representative Families:** Given a family $\mathcal{S}$ of sets that are partial solutions, we would like to replace $\mathcal{S}$ by a smaller subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$. If there is a partial solution in $\mathcal{S}$ that can be extended to a solution, it is clearly necessary that there would also be a partial solution in $\widehat{\mathcal{S}}$ that can be extended to a solution. The following definition captures such a family $\widehat{\mathcal{S}}$.

**Definition 1.** *Given a matroid $M = (E, \mathcal{I})$, and a family $\mathcal{S}$ of subsets of size $p$ of $E$, we say that a subfamily $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ $q$-*represents *$\mathcal{S}$ if for every pair of sets $X \in \mathcal{S}$, and $Y \subseteq E \setminus X$ such that $|Y| \leq q$ and $X \cup Y \in \mathcal{I}$, there is a set $\widehat{X} \in \widehat{\mathcal{S}}$ disjoint from $Y$ such that $\widehat{X} \cup Y \in \mathcal{I}$.*

The next two results enable the efficient construction of small representative families.

**Theorem 1 ([13,25]).** *Given a parameter $c \geq 1$, a uniform matroid $U_{n,k} = (E, \mathcal{I})$, and a family $\mathcal{S}$ of subsets of size $p$ of $E$, a family $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ of size at most $\frac{(ck)^k}{p^p(ck-p)^{k-p}} 2^{o(k)} \log n$ that $(k-p)$-represents $\mathcal{S}$ can be found in time $O(|\mathcal{S}|(ck/(ck-p))^{k-p} 2^{o(k)} \log n)$.*

**Theorem 2 ([13,18]).** *Given constants $\ell, k_1, k_2, \ldots, k_\ell$ and $k \leq \sum_{i=1}^{\ell} k_i$, a corresponding partition matroid $M = (E, \mathcal{I})$, and a family $\mathcal{S}$ of subsets of size $p$ of $E$, a family $\widehat{\mathcal{S}} \subseteq \mathcal{S}$ of size at most $\binom{k}{p} n^{O(1)}$ that $(k-p)$-represents $\mathcal{S}$ can be found in time $O(|\mathcal{S}| \binom{k}{p}^{\widetilde{w}-1} n^{O(1)})$, where $\widetilde{w} < 2.3727$ is the matrix multiplication exponent [27].*

Let $\mathsf{UniRep}(c, U_{n,k}, \mathcal{S})$ and $\mathsf{ParRep}(k, M, \mathcal{S})$ be the algorithms implied by Theorems 1 and 2, respectively.

**Guiding Trees:** Recall that $G = (V, E)$ is the input graph, and let $2 \leq d \leq k/2$ be a constant (to be determined).[3] Given a rooted tree $T$ and a node $v \in V_T$ that is not the root of $T$, let $f_T(v)$ be the father of $v$ in $T$. Given nodes $v, u \in V$, we say that a tree $T$ rooted at $v$ is a $(v,u)$-*tree* if $u \in V_T$. Furthermore, a $(v,u)$-tree $R$ is a $(v,u)$-*guide* if $3 \leq |V_R| \leq 2d$ and $V_R \subseteq V$ ($E_R$ may not be contained in $E$). Let $\mathcal{G}_{v,u}$ be the set of $(v,u)$-guides. Finally, let $\mathcal{T}_{v,u,\ell}$ be the set of $(v,u)$-trees on $\ell$ nodes, that, when unrooted, are subtrees of $G$.

We now define which subtrees of $G$ listen to the instructions of a given guide (see Fig. 2).

[3] The choice of $d$ concerns the analysis of the running times of our algorithms.

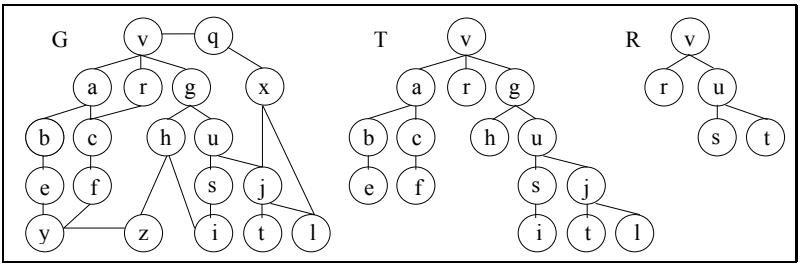


**Fig. 2.** A $(v, u)$-tree $T$, and a $(v, u)$-guide $R$, where $d = 3$, $k = 12$, and $T$ listens to $R$

**Definition 2.** *Given $v, u \in V$ and $\ell \leq k$, we say that $T \in \mathcal{T}_{v,u,\ell}$* listens to *$R \in \mathcal{G}_{v,u}$ if the following two conditions are fulfilled.*

1. *$\forall v', u' \in V_R : v'$ is an ancestor of $u'$ in $R$ iff $v'$ is an ancestor of $u'$ in $T$.*
2. *For each tree $X$ in the forest obtained by removing $V_R$ from $T$, let $N_X = \{v' \in V_R : \{v', u'\} \in E_T$ for some $u' \in V_X\}$.*
   *Then, $|N_X| \leq 2$, and $[N_X \neq \{v\} \rightarrow (|V_X \cup N_X| \leq k/d)]$.*

The next lemma, which asserts that none of the subtrees of $G$ relevant to solving $\mathrm{GM_D}$ is completely undisciplined, is implicit in [13].

**Lemma 3.** *For any rooted tree $T \in \mathcal{T}_{v,u,\ell}$, where $v, u \in V$ and $3 \leq \ell \leq k$, there exists $R \in \mathcal{G}_{v,u}$ to whom $T$ listens.*

**Feasible Colorings:** Given $U \subseteq V$, we say that a coloring $col : U \rightarrow C$ is *feasible* if $[\forall v \in U : col(v) \in Col(v)]$ and $[\forall c \in C : |\{v \in U : col(v) = c\}| \leq occ(c)]$. Denote by $\mathrm{ima}(col)$ the image of $col$.

## 3 An Algorithm for $\mathrm{GM_D}$

In this section we solve $\mathrm{GM_D}$ in time $O^*(6.86^k)$. Since in $\mathrm{GM_D}$ each node is assigned a set of colors whose size can be greater than 1, we may assume w.l.o.g that $M$ is a set equal to $C$ (a formal proof is given, e.g., in [22]).

The main idea of the algorithm is to iterate over all pairs of nodes $v, u \in V$, and all values $1 \leq \ell \leq k$. When we reach such $v, u$ and $\ell$, we have already computed, for all $v', u' \in V$ and $1 \leq \ell' < \ell$, representative families for families of corresponding "partial solutions". Each such partial solution is a union of a set $A$ containing exactly $\ell'$ nodes, and a set $B$ containing exactly $\ell'$ colors. The sets $A$ and $B$ correspond to a pair of a rooted tree $T \in \mathcal{T}_{v',u',\ell'}$ satisfying $A = V_T$, and a feasible coloring $col : A \rightarrow B$.

To compute a family of partial solutions corresponding to $v, u$ and $\ell$, we iterate over all $(v, u)$-guides in $\mathcal{G}_{v,u}$. We follow the instructions of the current guide $R$ by using another, internal dynamic programming-based computation. At each stage of this computation, we have a family of partial solutions listening to a certain

subtree of $R$. We unite these partial solutions with other *small* partial solutions, according to the instructions of $R$, thus *efficiently* constructing a family of partial solutions listening to a greater subtree of $R$. For this family, we compute a smaller representative family, so that the following stage can be executed efficiently. After iterating over all relevant guides, we find a family representing the union of the families returned by the internal dynamic programming-based computations. This family includes enough, but not too many, partial solutions corresponding to $v, u$ and $\ell$, which ensures the correctness of the algorithm.

### 3.1 The Algorithm

We now describe $\mathsf{GM_D\text{-}Alg}$, our algorithm for $\mathrm{GM_D}$ (see the pseudocode below). $\mathsf{GM_D\text{-}Alg}$ first generates a matrix M, where each entry $[v, u, c_v, c_u, \ell]$ holds a family that represents $Sol_{v,u,c_v,c_u,\ell}$, the family of every set $(X \cup Y)$ satisfying $|X| = |Y| = \ell$, for which there exist $T \in \mathcal{T}_{v,u,\ell}$ such that $X = V_T$, and a feasible $col : X \to Y$ satisfying $col(v) = c_v$ and $col(u) = c_u$.

**Algorithm 1.** $\mathsf{GM_D\text{-}Alg}(C, G = (V, E), Col, k)$

1. let M be a matrix that has an entry $[v, u, c_v, c_u, \ell]$ for all $v, u \in V$, $c_v \in Col(v)$, $c_u \in Col(u)$, and $1 \leq \ell \leq k$, initialized to $\emptyset$.
2. $\mathrm{M}[v,v,c,c,1] \Leftarrow \{\{v,c\}\}$ for all $v \in V$ and $c \in Col(v)$.
3. $\mathrm{M}[v,v,c,c,2] \Leftarrow \{\{v,u,c,c'\} : \{v,u\} \in E, c' \in Col(u) \setminus \{c\}\}$ for all $v \in V$ and $c \in Col(v)$.
4. $\mathrm{M}[v,u,c,c',2] \Leftarrow \{\{v,u,c,c'\}\}$ for all $\{v,u\} \in E$, $c \in Col(v)$ and $c' \in Col(u) \setminus \{c\}$.
5. **for all** $v, u \in V$, $c_v \in Col(v)$, $c_u \in Col(u)$, and $\ell = 3, \ldots, k$ **do**
6. let N be a matrix that has an entry $[R, col_R]$ for all $R \in \mathcal{G}_{v,u}$, and feasible $col_R : V_R \to C$ satisfying $col_R(v) = c_v$ and $col_R(u) = c_u$, initialized to $\emptyset$.
7. **for all** $[R, col_R] \in \mathrm{N}$ **do**
8. let $w_1, \ldots, w_{|V_R|}$ be a preorder on $V_R$, where $w_1 = v$.
9. let L be a matrix that has an entry $[i, \ell']$ for all $1 \leq i \leq |V_R|$ and $1 \leq \ell' \leq \ell$, initialized to $\emptyset$.
10. $\mathrm{L}[1, \ell'] \Leftarrow \mathrm{M}[v, v, c_v, c_v, \ell']$ for all $1 \leq \ell' < \ell$.
11. **for** $i = 2, \ldots, |V_R|$, and $\ell' = 2, \ldots, \ell$ **do**
12. let $\mathcal{A}$ include all sets $(U \cup W)$ for which there exists $2 \leq \ell'' \leq \min\{\ell', \ell - 1, k/d\}$ satisfying (1) or (2):
    (1) $U \cap W = \{f_R(w_i), col_R(f_R(w_i))\}$,
    $U \in \mathrm{M}[f_R(w_i), w_i, col_R(f_R(w_i)), col_R(w_i), \ell'']$ and $W \in \mathrm{L}[i-1, \ell' - \ell'' + 1]$.
    (2) $U \cap W = \{w_i, col_R(w_i)\}$,
    $U \in \mathrm{M}[w_i, w_i, col_R(w_i), col_R(w_i), \ell'']$ and $W \in \mathrm{L}[i, \ell' - \ell'' + 1]$.
13. $\mathrm{L}[i, \ell'] \Leftarrow \mathsf{UniRep}(1.447, U_{(|V|+|C|),2k}, \mathcal{A})$.
14. **end for**
15. $\mathrm{N}[R, col_R] \Leftarrow \mathrm{L}[|V_R|, \ell]$.
16. **end for**
17. $\mathrm{M}[v, u, c_v, c_u, \ell] \Leftarrow \mathsf{UniRep}(1.447, U_{(|V|+|C|),2k}, \bigcup_{[R,col_R] \in \mathrm{N}} \mathrm{N}[R, col_R])$.
18. **end for**
19. accept iff $(\bigcup_{v \in V, c_v \in Col(v)} \mathrm{M}[v, v, c_v, c_v, k]) \neq \emptyset$.

Then, in Steps 2–4, $\mathsf{GM_D\text{-}Alg}$ computes all "basic" entries of M, i.e., entries of the form $[v, u, c_v, c_u, \ell]$, where $\ell \leq 2$. Next, in Step 5, $\mathsf{GM_D\text{-}Alg}$ iterates over all values $v, u, c_v, c_u$ and $\ell$ that define an entry of M that is not basic, in an order that guarantees that when we reach an entry [\$] of M, we have already computed entries of M that are relevant to [\$]. Now, consider a specific iteration of Step 5, and note that the goal of this iteration is to compute $\mathrm{M}[v, u, c_v, c_u, \ell]$.

$\mathsf{GM_D\text{-}Alg}$, in Step 6, generates a matrix N. Each entry $[R, col_R]$ holds a family that represents a subfamily of $Sol_{v,u,c_v,c_u,\ell}$. A set $(X \cup Y) \in Sol_{v,u,c_v,c_u,\ell}$ belongs to this subfamily if its corresponding $(v, u)$-tree $T \in \mathcal{T}_{v,u,\ell}$ and feasible coloring $col$ also satisfy the requirements that $T$ listens to $R$, and $col$ colors the nodes in $V_R$ exactly as $col_R$ colors them. Now, consider a specific iteration of Step 7, and note that the goal of this iteration is to compute $\mathrm{N}[R, col_R]$. To this end, $\mathsf{GM_D\text{-}Alg}$ executes an internal dynamic programming-based computation, which takes place in Steps 9–14.

First, in Step 9, $\mathsf{GM_D\text{-}Alg}$ generates a matrix L. Almost every entry $[i, \ell']$ holds a family that represents $Sol_{i,\ell'}$,[4] the family including every set $(X \cup Y)$ satisfying $|X| = |Y| = \ell'$, for which there exist a $(v, w_i)$-tree $T \in \mathcal{T}_{v,w_i,\ell'}$ and a feasible coloring $col : X \to Y$, satisfying the following conditions. The subtree $T$ listens to the subtree of $R$ induced by $\{w_1, \ldots, w_i\}$, $X = V_T$, and $col$ colors the nodes in $\{w_1, \ldots, w_i\}$ exactly as $col_R$ colors them. Note that the subgraph of $R$ induced by $\{w_1, \ldots, w_i\}$ is a tree because of the preorder defined in Step 8. Then, in Step 10, $\mathsf{GM_D\text{-}Alg}$ computes all "basic" entries of L, i.e., entries of the form $[1, \ell']$. Next, in Step 11, $\mathsf{GM_D\text{-}Alg}$ iterates over all values $i$ and $\ell'$ that define an entry of L that is not basic, in an order that guarantees that when we reach an entry [\$] of L, we have already computed entries of L that are relevant to [\$]. Now, consider a specific iteration of Step 11, and note that the goal of this iteration is to compute $\mathrm{L}[i, \ell']$.

$\mathsf{GM_D\text{-}Alg}$, in Step 12, computes a family $\mathcal{A}$ that represents $Sol_{i,\ell'}$. The computation involves uniting sets $U$, found in previous stages of the external dynamic programming-based computation (i.e., $U$ belongs to an entry of M), with sets $W$, found in previous stages of the internal dynamic programming-based computation (i.e., $W$ belongs to an entry of L). It is easy to verify that the restrictions posed on the choices of $U$ and $W$ gaurantee that their union indeed belongs to $Sol_{i,\ell'}$, noting the following observations. The restriction $\ell'' \leq k/d$ concerns Condition 2 in Definition 2, whose relevance follows from the requirement of existence of a $(v, w_i)$-tree $T$ as defined above. The first line in each of the options (1) and (2) ensures that we do not use any node or color more than once. The other line of option (1) ensure that $U \in Sol_{f_R(w_i),w_i,col_R(f_R(w_i)),col_R(w_i),\ell''}$ and $W \in Sol_{i-1,\ell'-\ell''+1}$, and the other line of option (2) ensures that $U \in Sol_{w_i,w_i,col_R(w_i),col_R(w_i),\ell''}$ and $W \in Sol_{i,\ell'-\ell''+1}$.

After computing $\mathcal{A}$, $\mathsf{GM_D\text{-}Alg}$ computes $\mathrm{L}[i, \ell']$ (in Step 13) by finding a smaller family that represents $\mathcal{A}$. Upon completing the computation of L, since $V_R = \{w_1, \ldots, w_{|V_R|}\}$, $\mathsf{GM_D\text{-}Alg}$ can compute $\mathrm{N}[R, col_R]$ (in Step 15) by a simple assignment. Then, the union of the families stored in N is a family that represents

[4] More precisely, here we refer to all entries $[i, \ell']$ such that $(\ell' = \ell \to i = |V_R|)$.

$Sol_{v,u,c_v,c_u,\ell}$, a claim supported by Lemma 3. Therefore, in Step 19,$\mathsf{GM_D\text{-}Alg}$ can compute $\mathrm{M}[v,u,c_v,c_u,\ell]$ by simply finding a family that represents this union.

Finally, $\mathsf{GM_D\text{-}Alg}$ accepts iff $\bigcup_{v\in V,c_v\in Col(v)} \mathrm{M}[v,v,c_v,c_v,k] \neq \emptyset$. Indeed, note that the input is a yes-instance iff $\bigcup_{v\in V,c_v\in Col(v)} Sol_{v,v,c_v,c_v,k} \neq \emptyset$.

### 3.2 Correctness

Recall that $Sol_{v,u,c_v,c_u,\ell}$ is the family of every set $(X\cup Y)$ satisfying $|X| = |Y| = \ell$, for which there exist $T \in \mathcal{T}_{v,u,\ell}$ such that $X = V_T$, and a feasible $col : X \to Y$ satisfying $col(v) = c_v$ and $col(u) = c_u$.

The correctness of the algorithm follows directly from the next lemma.

**Lemma 4.** *Every entry $M[v,u,c_v,c_u,\ell]$ $(2k-2\ell)$-represents $Sol_{v,u,c_v,c_u,\ell}$.*

*Proof (Lemma 4).* By Steps 1–4, the lemma holds for any entry $[v,u,c_v,c_u,\ell]$ in M such that $\ell \leq 2$. Now, consider some $v,u \in V$, $c_v \in Col(v)$, $c_u \in Col(u)$ and $3 \leq \ell \leq k$, and assume that the lemma holds for all $v',u' \in V$, $c'_v \in Col(v')$, $c'_u \in Col(u')$ and $1 \leq \ell' < \ell$.

For an entry $\mathrm{N}[R,col_R]$, let $Sol(R,col_R)_{v,u,c_v,c_u,\ell}$ include every set $(X\cup Y) \in Sol_{v,u,c_v,c_u,\ell}$ whose corresponding $(v,u)$-tree $T \in \mathcal{T}_{v,u,\ell}$ and feasible coloring $col$ also satisfy the requirements that $T$ listens to $R$, and $col$ colors the nodes in $V_R$ exactly as $col_R$ colors them.

Towards proving the main inductive claim, we need the following claim.

**Claim 1.** *Every entry $N[R,col_R]$ $(2k-2\ell)$-represents $Sol(R,col_R)_{v,u,c_v,c_u,\ell}$.*

We first show that Claim 1 implies the correctness of the main inductive claim. Since representation is a transitive relation, it is enough to prove that $\mathcal{B} = \bigcup_{[R,col_R]\in \mathrm{N}} \mathrm{N}[R,col_R]$ $(2k-2\ell)$-represents $Sol_{v,u,c_v,c_u,\ell}$. By Claim 1, $\mathcal{B} \subseteq \bigcup_{[R,col_R]\in \mathrm{N}} Sol(R,col_R)_{v,u,c_v,c_u,\ell} \subseteq Sol_{v,u,c_v,c_u,\ell}$.

Consider some sets $A \in Sol_{v,u,c_v,c_u,\ell}$, and $B \subseteq (V\cup C)\setminus A$ such that $|B| \leq 2k-2\ell$. Since $A \in Sol_{v,u,c_v,c_u,\ell}$, we have that $A$ is of the form $(X_A \cup Y_A)$, where $|X_A| = |Y_A| = \ell$, for which there exist $T \in \mathcal{T}_{v,u,\ell}$ such that $X_A = V_T$, and a feasible $col : X_A \to Y_A$ satisfying $col(v) = c_v$ and $col(u) = c_u$. By Lemma 3, there exists $R \in \mathcal{G}_{v,u}$ such that $T$ listens to $R$. Let $col_R$ be defined as $col$ when restricted to the domain $V_R$. We get that $A \in Sol(R,col_R)_{v,u,c_v,c_u,\ell}$. By Claim 1, there is $\widehat{A} \in \mathrm{N}[R,col_R] \subseteq \mathcal{B}$ such that $\widehat{A}\cap B = \emptyset$. Thus, $\mathcal{B}$ $(2k-2\ell)$-represents $Sol_{v,u,c_v,c_u,\ell}$. □

We now turn to prove Claim 1.

*Proof (Claim 1).* Consider an iteration of Step 7, corresponding to an entry $\mathrm{N}[R,col_R]$. For an entry $\mathrm{L}[i,\ell']$, let $R(i)$ be the subtree of $R$ induced by $\{w_1,\ldots,w_i\}$. Moreover, let $Sol_{i,\ell'}$ be the family including every set $(X\cup Y)$ satisfying $|X| = |Y| = \ell'$, for which there exist a $(v,w_i)$-tree $T \in \mathcal{T}_{v,w_i,\ell'}$ and a feasible coloring $col : X \to Y$, satisfying the following conditions. The subtree $T$ listens to $R(i)$, $X = V_T$, and $col$ colors the nodes in $\{w_1,\ldots,w_i\}$ exactly as $col_R$ colors them.

Towards proving Claim 1, we need the following claim.

**Claim 2.** *Every entry L[i,ℓ′], where* $(\ell' = \ell \rightarrow i = |V_R|)$, $(2k–2\ell')$*-represents* $Sol_{i,\ell'}$.

Since $\mathrm{N}[R, col_R] = \mathrm{L}[|V_R|, \ell]$ and $Sol(R, col_R)_{v,u,c_v,c_u,\ell} = Sol_{|V_R|,\ell}$, Claim 2 implies the correctness of Claim 1. □

Finally, we turn to prove Claim 2, concluding the correctness of the algorithm.

*Proof (Claim 2).* By Steps 9 and 10, and the induction hypothesis concerning the matrix M, the claim holds for ($i = 1$ and all $1 \leq \ell' < \ell$) and (all $1 \leq i \leq |V_R|$ and $\ell' = 1$). Now, consider some $2 \leq i \leq |V_R|$ and $2 \leq \ell' \leq \ell$, and assume that the claim holds for all $1 \leq i' \leq i$ and $1 \leq \ell'' < \ell'$. Since representation is a transitive relation, it is enough to prove that $\mathcal{A}$ $(2k - 2\ell')$-represents $Sol_{i,\ell'}$.

By definition, a set $A$ belongs to $Sol_{i,\ell'}$ iff there are sets $U$ and $W$ whose union is $A$, for which there exists $2 \leq \ell'' \leq \min\{\ell', \ell–1, k/d\}$ satisfying (1) or (2):

1. $U \cap W = \{f_R(w_i), col_R(f_R(w_i))\}$,
   $U \in Sol_{f_R(w_i),w_i,col_R(f_R(w_i)),col_R(w_i),\ell''}$ and $W \in Sol_{i-1,\ell'-\ell''+1}$.
2. $U \cap W = \{w_i, col_R(w_i)\}$,
   $U \in Sol_{w_i,w_i,col_R(w_i),col_R(w_i),\ell''}$ and $W \in Sol_{i,\ell'-\ell''+1}$.

Thus, by Step 12 and the inductive hypotheses for the matrices M and L, $\mathcal{A} \subseteq Sol_{i,\ell'}$. Now, consider some $A \in Sol_{i,\ell'}$, and $B \subseteq (V \cup C) \setminus A$ such that $|B| \leq 2k - 2\ell'$. Since $A \in Sol_{i,\ell'}$, there are $U$, $W$, and $\ell''$ as mentioned above.

First, suppose that $U$, $W$, and $\ell''$ correspond to the first option. Note that $|(W \setminus \{f_R(w_i), col_R(f_R(w_i))\}) \cup B| = |W| - 2 + |B| \leq 2(\ell' - \ell'' + 1) - 2 + (2k - 2\ell') = 2k - 2\ell''$. Therefore, by the inductive hypothesis concerning M, there is a set $\widehat{U} \in \mathrm{M}[f_R(w_i), w_i, col_R(f_R(w_i)), col_R(w_i), \ell'']$ such that $\widehat{U} \cap ((W \setminus \{f_R(w_i), col_R(f_R(w_i))\}) \cup B) = \emptyset$. Moreover, $|(\widehat{U} \setminus \{f_R(w_i), col_R(f_R(w_i))\}) \cup B| = |\widehat{U}| - 2 + |B| \leq (2\ell'') - 2 + (2k - 2\ell') = 2k - 2(\ell' - \ell'' + 1)$. Therefore, by the inductive hypothesis concerning L, there is a set $\widehat{W} \in \mathrm{L}[i - 1, \ell' - \ell'' + 1]$ such that $\widehat{W} \cap ((\widehat{U} \setminus \{f_R(w_i), col_R(f_R(w_i))\}) \cup B) = \emptyset$.

Now, suppose that $U$, $W$, and $\ell''$ correspond to the second option. Note that $|(W \setminus \{w_i, col_R(w_i)\}) \cup B| = |W| - 2 + |B| \leq 2(\ell' - \ell'' + 1) - 2 + (2k - 2\ell') = 2k - 2\ell''$. Therefore, by the inductive hypothesis concerning M, there is a set $\widehat{U} \in \mathrm{M}[w_i, w_i, col_R(w_i), col_R(w_i), \ell'']$ such that $\widehat{U} \cap ((W \setminus \{w_i, col_R(w_i)\}) \cup B) = \emptyset$. Moreover, $|(\widehat{U} \setminus \{w_i, col_R(w_i)\}) \cup B| = |\widehat{U}| - 2 + |B| \leq (2\ell'') - 2 + (2k - 2\ell') = 2k - 2(\ell' - \ell'' + 1)$. Therefore, by the inductive hypothesis concerning L, there is a set $\widehat{W} \in \mathrm{L}[i, \ell' - \ell'' + 1]$ such that $\widehat{W} \cap ((\widehat{U} \setminus \{w_i, col_R(w_i)\}) \cup B) = \emptyset$. □

### 3.3 Running Time

Let $0 < \epsilon < 1$ be some constant, $c = 1.447$, and $q = 2k$. Choose a constant $d \geq 2$ satisfying, for any integer $n$, $\binom{cn}{n/d} = O(2^{\epsilon n})$ and $1/d \leq \epsilon$.

For any $0 \leq r^* \leq q$ and call $\mathsf{UniRep}(c, U_{|V|+|C|,q}, \mathcal{S})$ executed by $\mathsf{GM_D\text{-}Alg}$, where $\mathcal{S}$ is a family of subsets of size $r^*$ of $V \cup C$, there exists $0 \leq r' \leq \min\{r^*, q/d\}$ such that

$$|\mathcal{S}| \leq 2^{o(q)}|V|^{O(d)}\left(\frac{(cq)^q}{(r^*-r')^{r^*-r'}(cq-(r^*-r'))^{q-(r^*-r')}}\right)\left(\frac{(cq)^q}{r'^{r'}(cq-r')^{q-r'}}\right).$$

We get that $\mathsf{GM_D}$-$\mathsf{Alg}$ runs in time

$$O\Big(2^{o(q)}|V|^{O(d)}\max_{r=0}^{q}\max_{r'=0}^{\min\{q-r,q/d\}}\left\{\left(\frac{(cq)^q}{r^r(cq-r)^{q-r}}\right)\left(\frac{(cq)^q}{r'^{r'}(cq-r')^{q-r'}}\right)\left(\frac{cq}{cq-(r+r')}\right)^{q-(r+r')}\right\}\Big)$$

$$=O\Big(2^{o(q)}|V|^{O(1)}\max_{r=0}^{q}\max_{r'=0}^{\min\{q-r,q/d\}}\left\{\left(\frac{(cq)^q}{r^r(cq-r)^{q-r}}\right)\binom{cq}{r'}\left(\frac{cq}{cq-(r+q/d)}\right)^{q-r}\right\}\Big)$$

$$=O\Big(2^{o(q)}|V|^{O(1)}\max_{r=0}^{q}\left\{\left(\frac{(cq)^q}{r^r(cq-r)^{q-r}}\right)\binom{cq}{q/d}\left(\frac{cq}{cq-r-(1/d)q}\right)^{q-r}\right\}\Big)$$

$$=O\Big(2^{\epsilon q+o(q)}|V|^{O(1)}\max_{r=0}^{q}\left\{\left(\frac{(cq)^q}{r^r(cq-r)^{q-r}}\right)\left(\frac{cq}{cq-r-\epsilon q}\right)^{q-r}\right\}\Big).$$

By choosing a small enough $\epsilon > 0$, the maximum is obtained at $r = \alpha q$, where $\alpha \cong 0.55277$. Thus, $\mathsf{GM_D}$-$\mathsf{Alg}$ runs in time $O(6.85414^k|V|^{O(1)})$.

## 4 An Algorithm for GM

In this section we develop algorithm $\mathsf{GM}$-$\mathsf{Alg}$, proving the following result.

**Theorem 5.** $\mathsf{GM}$-$\mathsf{Alg}$ *solves* GM *in time* $O^*(5.21914^k)$.

Algorithm $\mathsf{GM}$-$\mathsf{Alg}$ computes families of "partial solutions" that contain only nodes, and handles colors by adding a parameter to the matrices holding these families. More precisely, given a pair of nodes $v, u \in V$, and a subset of colors $D \subseteq C$, we compute families of partial solutions of the following form. A partial solution is a subset $U \subseteq V$ of $|D|$ nodes, for which there exist a $(v,u)$-tree $T \in \mathcal{T}_{v,u,|D|}$ satisfying $U = V_T$, and a feasible coloring $col : U \to D$. Having a family of such partial solutions, we compute a family that represents it, calling algorithm $\mathsf{UniRep}$. Such computations of representative families are embedded in a dynamic programming-based framework, whose progress is governed by guiding trees. Note that, since we iterate over every subset $D \subseteq C$, the running time of $\mathsf{GM}$-$\mathsf{Alg}$ crucially relies on the fact that deletions are not allowed in GM.

## 5 An Algorithm for $\mathrm{RGM_D}$

In this section we develop algorithm $\mathsf{RGM_D}$-$\mathsf{Alg}$, proving the following result.

**Theorem 6.** $\mathsf{RGM_D}$-$\mathsf{Alg}$ *solves* $\mathrm{RGM_D}$ *in time* $O^*(5.1791^k)$.

To efficiently compute representative families, we define a partition matroid $P = P(C, M, G, Col) = (E, \mathcal{I})$ as follows. Denote $C = \{c_1, \ldots, c_{|C|}\}$. Now, let $E = V$ be partitioned into sets $E_1, \ldots, E_{|C|}$, where $E_i = \{v \in V : c_i \in Col(v)\}$,

for all $1 \leq i \leq |C|$. The sets $E_1, \ldots, E_{|C|}$ are disjoint because $|Col(v)| = 1$, for all $v \in V$. Now, let $k_i = occ(c_i)$ for all $1 \leq i \leq |C|$ (recall that $occ(c)$ is the number of occurences of a color $c$ in $M$). Accordingly, define $\mathcal{I} = \mathcal{I}(C, M, G, Col) = \{S \subseteq E : |S \cap E_1| \leq k_1, \ldots, |S \cap E_{|C|}| \leq k_{|C|}\}$.

Intuitively, this definition ensures that $U \in \mathcal{I}$ iff $U$ can be colored without using any color "too many" times, i.e., there exists a feasible coloring $col : U \to C$.

Algorithm $\mathsf{RGM_D\text{-}Alg}$ computes families of "partial solutions" that contain only nodes, and handles colors by computing representative families with respect to the partition matroid $P$. More precisely, when we now consider a pair of nodes $v, u \in V$, and a value $1 \leq \ell \leq k$, we compute families of partial solutions of the following form. A partial solution is a set of nodes $U \in \mathcal{I}$, for which there exists a $(v, u)$-tree $T \in \mathcal{T}_{v,u,\ell}$ satisfying $U = V_T$. Having a family of such partial solutions, we compute a family that represents it with respect to the matroid $P$, calling algorithm $\mathsf{ParRep}$. Such computations of representative families are embedded in a dynamic programming-based framework, whose progress is governed by guiding trees.

## References


1. Alon, N., Yuster, R., Zwick, U.: Color coding. J. Assoc. Comput. Mach. 42(4), 844–856 (1995)
2. Ambalath, A.M., Balasundaram, R., Rao H., C., Koppula, V., Misra, N., Philip, G., Ramanujan, M.S.: On the kernelization complexity of colorful motifs. In: Raman, V., Saurabh, S. (eds.) IPEC 2010. LNCS, vol. 6478, pp. 14–25. Springer, Heidelberg (2010)
3. Betzler, N., Bevern, R., Fellows, M.R., Komusiewicz, C., Niedermeier, R.: Parameterized algorithmics for finding connected motifs in biological networks. IEEE/ACM Trans. Comput. Biol. Bioinf. 8(5), 1296–1308 (2011)
4. Betzler, N., Fellows, M.R., Komusiewicz, C., Niedermeier, R.: Parameterized algorithms and hardness results for some graph motif problems. In: Ferragina, P., Landau, G.M. (eds.) CPM 2008. LNCS, vol. 5029, pp. 31–43. Springer, Heidelberg (2008)
5. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Narrow sieves for parameterized paths and packings. CoRR abs/1007.1161 (2010)
6. Björklund, A., Kaski, P., Kowalik, L.: Probably optimal graph motifs. In: STACS, pp. 20–31 (2013)
7. Bruckner, S., Hüffner, F., Karp, R.M., Shamir, R., Sharan, R.: Topology-free querying of protein interaction networks. J. Comput. Biol. 17(3), 237–252 (2010)
8. Dondi, R., Fertin, G., Vialette, S.: Finding approximate and constrained motifs in graphs. In: Giancarlo, R., Manzini, G. (eds.) CPM 2011. LNCS, vol. 6661, pp. 388–401. Springer, Heidelberg (2011)
9. Dondi, R., Fertin, G., Vialette, S.: Maximum motif problem in vertex-colored graphs. In: Kucherov, G., Ukkonen, E. (eds.) CPM 2009. LNCS, vol. 5577, pp. 221–235. Springer, Heidelberg (2009)
10. Fellows, M.R., Fertin, G., Hermelin, D., Vialette, S.: Sharp tractability borderlines for finding connected motifs in vertex-colored graphs. In: Arge, L., Cachin, C., Jurdziński, T., Tarlecki, A. (eds.) ICALP 2007. LNCS, vol. 4596, pp. 340–351. Springer, Heidelberg (2007)

11. 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)
12. Fionda, V., Palopoli, L.: Biological network querying techniques: Analysis and comparison. J. Comput. Biol. 18(4), 595–625 (2011)
13. Fomin, F.V., Lokshtanov, D., Saurabh, S.: Efficient computation of representative sets with applications in parameterized and exact agorithms. In: SODA (see also: CoRR abs/1304.4626), pp. 142–151 (2014)
14. Guillemot, S., Sikora, F.: Finding and counting vertex-colored subtrees. In: Hliněný, P., Kučera, A. (eds.) MFCS 2010. LNCS, vol. 6281, pp. 405–416. Springer, Heidelberg (2010)
15. Koutis, I.: Constrained multilinear detection for faster functional motif discovery. Inf. Process. Lett. 112(22), 889–892 (2012)
16. Koyutürk, M.: Algorithmic and analytical methods in network biology. Wiley Interdiscip. Rev. Syst. Biol. Med. 2(3), 277–292 (2010)
17. Lacroix, V., Fernandes, C.G., Sagot, M.F.: Motif search in graphs: Application to metabolic networks. IEEE/ACM Trans. Comput. Biol. Bioinf. 3(4), 360–368 (2006)
18. Lokshtanov, D., Misra, P., Panolan, F., Saurabh, S.: Deterministic truncation of linear matroids. CoRR abs/1404.4506 (2014)
19. Oxley, J.G.: Matroid theory. Oxford University Press (2006)
20. Pinter, R.Y., Shachnai, S., Zehavi, M.: Deterministic parameterized algorithms for the graph motif problem, `http://www.cs.technion.ac.il/~hadas/PUB/Graph_Motif_full.pdf`
21. Pinter, R.Y., Zehavi, M.: Partial information network queries. In: Lecroq, T., Mouchard, L. (eds.) IWOCA 2013. LNCS, vol. 8288, pp. 362–375. Springer, Heidelberg (2013)
22. Pinter, R.Y., Zehavi, M.: Algorithms for topology-free and alignment network queries. J. Discrete Algorithms (to appear, 2014)
23. Pinter-Wollman, N., Hobson, E.A., Smith, J.E., Edelman, A.J., Shizuka, D., de Silva, S., Waters, J.S., Prager, S.D., Sasaki, T., Wittemyer, G., Fewell, J., McDonald, D.B.: The dynamics of animal social networks: analytical, conceptual, and theoretical advances. Behavioral Ecology 25(2), 242–255 (2014)
24. Rizzi, R., Sikora, F.: Some results on more flexible versions of graph motif. In: Hirsch, E.A., Karhumäki, J., Lepistö, A., Prilutskii, M. (eds.) CSR 2012. LNCS, vol. 7353, pp. 278–289. Springer, Heidelberg (2012)
25. Shachnai, H., Zehavi, M.: Faster computation of representative families for uniform matroids with applications. CoRR abs/1402.3547 (2014)
26. Sikora, F.: An (almost complete) state of the art around the graph motif problem. Université Paris-Est Technical reports (2012)
27. Williams, V.V.: Multiplying matrices faster than Coppersmith-Winograd. In: STOC, pp. 887–898 (2012)