



HAL
open science

Weak pattern matching in colored graphs: Minimizing the number of connected components

Riccardo Dondi, Guillaume Fertin, Stéphane Vialette

► **To cite this version:**

Riccardo Dondi, Guillaume Fertin, Stéphane Vialette. Weak pattern matching in colored graphs: Minimizing the number of connected components. 10th Italian Conference on Theoretical Computer Science (ICTCS 2007), 2007, Rome, Italy. pp.27-38. <hal-00417910>

HAL Id: hal-00417910

<https://hal.science/hal-00417910v1>

Submitted on 17 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



HAL Authorization

Weak pattern matching in colored graphs: Minimizing the number of connected components

Riccardo Dondi

*Dipartimento di Scienze dei Linguaggi, della Comunicazione e degli Studi Culturali
Università degli Studi di Bergamo, Piazza Vecchia 8, 24129 Bergamo - Italy
EMAIL: riccardo.dondi@unibg.it*

Guillaume Fertin

*Laboratoire d'Informatique de Nantes-Atlantique (LINA), FRE CNRS 2729
Université de Nantes, 2 rue de la Houssinière, 44322 Nantes Cedex 3 - France
EMAIL: guillaume.fertin@lina.univ-nantes.fr*

Stéphane Vialette

*Laboratoire de Recherche en Informatique (LRI), UMR CNRS 8623
Université Paris-Sud 11, 91405 Orsay - France
EMAIL: stephane.vialette@lri.fr*

In the context of metabolic network analysis, Lacroix *et al.*¹¹ introduced the problem of finding occurrences of motifs in vertex-colored graphs, where a motif is a multiset of colors and an occurrence of a motif is a subset of connected vertices which are colored by all colors of the motif. We consider in this paper the above-mentioned problem in one of its natural optimization forms, referred hereafter as the MIN-CC problem: Find an occurrence of a motif in a vertex-colored graph, called the *target graph*, that induces a minimum number of connected components.

Our results can be summarized as follows. We prove the MIN-CC problem to be **APX**-hard even in the extremal case where the motif is a set and the target graph is a path. We complement this result by giving a polynomial-time algorithm in case the motif is built upon a fixed number of colors and the target graph is a path. Also, extending recent research⁸, we prove the MIN-CC problem to be fixed-parameter tractable when parameterized by the size of the motif, and we give a faster algorithm in case the target graph is a tree. Furthermore, we prove the MIN-CC problem for trees not to be approximable within ratio $c \log n$ for some constant $c > 0$, where n is the order of the target graph, and to be **W[2]**-hard when parameterized by the number of connected components in the occurrence of the motif. Finally, we give an exact efficient exponential-time algorithm for the MIN-CC problem in case the target graph is a tree.

1. Introduction

In the context of metabolic network analysis, Lacroix *et al.*¹¹ introduced the following vertex colored graph problem (referred hereafter as the GRAPH-MOTIF problem): Given a vertex-colored graph G and a multiset of colors \mathcal{M} , decide whether G has a connected subset of vertices which are exactly colored by \mathcal{M} . There, vertices correspond to chemical compounds or reactions, and each edge (v_i, v_j) corresponds to an interaction between the two compounds or reactions v_i and v_j . The vertex coloring is used to specify different chemical types or functionalities. In this scenario, connected motifs correspond to interaction-related submodules of the network which consist of a specific set of chemical compounds and reactions. A method for a rational decomposition of a metabolic network into relatively independent functional subsets is essential for a better understanding of the modularity and organization principles in the network^{5,11}. Notice that Ideker considered a related relevant work¹⁰.

Unfortunately, it turns out that the GRAPH-MOTIF problem is **NP**-complete even if the graph is a tree and the motif is actually a set^{8,11}. Moreover, the GRAPH-MOTIF problem is fixed-parameter tractable when parameterized by the size of the motif, but **W[1]**-hard when parameterized by the number of distinct colors in \mathcal{M} ⁸. Finally, Lacroix *et al.*¹¹ gave an exact algorithm dedicated to solve small instances.

For metabolic network analysis, the GRAPH-MOTIF problem appears, however, to be too stringent. Indeed, due to measurement errors, it is often not possible to find a connected component of the graph G which corresponds exactly to the motif \mathcal{M} . Hence one needs to relax the definition of an occurrence of a motif in a metabolic network. Therefore, aiming at dealing with inherent imprecise data, we consider in this paper the above-mentioned problem in one of its natural optimization form, referred hereafter as the MIN-CC problem: Find an occurrence of a motif in a vertex-colored graph, that induces a minimum number of connected components.

The paper is organized as follows. Section 2 provides basic notations and definitions that we will use in the paper. In Section 3, we prove the MIN-CC problem to be **APX**-hard even if the motif is a set and the target graph is a path. Extending recent research⁸, we prove in Section 4 that the MIN-CC problem is fixed-parameter tractable when parameterized by the size of the motif, and we give a faster algorithm in case the target graph is a tree. In Section 5 we present a polynomial-time algorithm in case the motif is built upon a fixed number of colors and the target graph is a path. Section 6 is devoted to hardness of approximation in case the target graph is a tree

and we present in Section 7 an exact efficient exponential-time algorithm for trees. Section 8 concludes our work and suggests future directions of research.

2. Preliminaries

We assume readers have basic knowledge about graph theory⁶ and we shall only recall basic notations here. Let G be a graph. We write $\mathbf{V}(G)$ for the set of vertices and $\mathbf{E}(G)$ for the set of edges. For any $V' \subseteq \mathbf{V}(G)$, we denote by $G[V']$ the subgraph of G induced by the vertices V' , that is $G[V'] = (V', E')$ and $(u, v) \in E'$ iff $u, v \in V'$ and $(u, v) \in \mathbf{E}(G)$. Let \mathcal{M} be a multiset of colors, whose colors are taken from the set $\mathcal{C} = \{c_1, c_2, \dots, c_q\}$. Let G be a connected graph, where every vertex $u \in V(G)$ is assigned a color $\lambda(u) \in \mathcal{C}$. For any subset V' of V , let $C(V')$ be the multiset of colors assigned to the vertices in V' . A subset of vertices $V' \subseteq \mathbf{V}(G)$ is said to *match* a multiset of colors \mathcal{M} if $C(V')$ is equal to \mathcal{M} . A *color-preserving injective mapping* θ of \mathcal{M} to G is an injective mapping $\theta : \mathcal{M} \rightarrow \mathbf{V}(G)$, such that $\lambda(\theta(c)) = c$ for every $c \in \mathcal{M}$. The subgraph induced by a color-preserving injective mapping $\theta : \mathcal{M} \rightarrow \mathbf{V}(G)$ is the subgraph of G induced by the images of θ in G .

We are now in position to formally define the MIN-CC problem we are interested in. Given a set of colors \mathcal{C} , a multiset (motif) \mathcal{M} of size k of colors from \mathcal{C} and a target graph G of order n together with a vertex-coloring mapping $\lambda : \mathbf{V}(G) \rightarrow \mathcal{C}$, find a color preserving injective mapping $\theta : \mathcal{M} \rightarrow \mathbf{V}(G)$, *i.e.*, $\lambda(\theta(c)) = c$ for every $c \in \mathcal{M}$ that minimizes the number of connected components in the subgraph induced by θ . In other words, the MIN-CC problem asks to find a subset $V' \subseteq \mathbf{V}(G)$ that matches \mathcal{M} , and that minimizes the number of connected components of $G[V']$. The MIN-CC problem was proved to be **NP**-complete even if the target graph is a tree and the occurrence is required to be connected (the occurrence of \mathcal{M} in G results in one connected component) but fixed-parameter tractable in this case when parameterized by the size of the given motif¹¹.

3. Hardness result for paths

In this section we show that the MIN-CC problem is **APX**-hard (not approximable within a constant) even in the simple case where the motif \mathcal{M} is a set and the target graph is a path in which each color in \mathcal{C} occurs exactly twice. Our proof consists in a reduction from a restricted version of the PAINTSHOP-FOR-WORDS problem^{2,3,15}.

First, we need some additional definitions. Define an *isogram* to be a word in which no letter is used more than once. A *pair isogram* is a word in which each letter occurs exactly twice. A *cover* of size k of a word u is an ordered collection of words $C = (v_1, v_2, \dots, v_k)$ such that $u = w_1 v_1 w_2 v_2 \dots w_k v_k w_{k+1}$ and $v = v_1 v_2 \dots v_k$ is an isogram. The cover is called *prefix* (resp. *suffix*) if w_1 (resp. w_{k+1}) is the empty word.

A *proper 2-coloring* of a pair isogram u is an assignment f of colors c_1 and c_2 to the letters of u such that every letter of u is colored with color c_1 once and colored with color c_2 once. If two adjacent letters x and y are colored with different colors we say that there is a *color change* between x and y . For the sake of brevity, we denote a pair isogram u together with a proper 2-coloring f of it as the pair (u, f) .

The 1-REGULAR-2-COLORS-PAINT-SHOP problem is defined as follows: Given a pair isogram u , find a 2-coloring f of u that minimizes the number of color changes in (u, f) . Bonsma² proved that the 1-REGULAR-2-COLORS-PAINT-SHOP problem is **APX**-hard. We show here how to reduce the 1-REGULAR-2-COLORS-PAINT-SHOP problem to the MIN-CC problem for paths. We need the following easy lemmas.

Lemma 3.1. *Let u be a pair isogram and C be a minimum cardinality cover of u . Then C cannot be both prefix and suffix.*

Lemma 3.2. *A pair isogram has a proper 2-coloring with at most k color changes iff it has a cover of size at most $\lceil \frac{k}{2} \rceil$.*

Combining Lemma 3.2 with the fact that the 1-REGULAR-2-COLORS-PAINT-SHOP problem is **APX**-hard, we state the following result.

Proposition 3.1. *The following problem is **APX**-hard : Given a pair isogram u , find a minimum cardinality cover of u .*

Corollary 3.1. *The MIN-CC problem is **APX**-hard even if \mathcal{M} is a set and P is a path in which each color appears at most twice.*

4. Fixed-parameter algorithms

Corollary 3.1 gives us a sharp hardness result for the MIN-CC problem. To complement this negative result, we first prove here that the MIN-CC problem is fixed-parameter tractable^{7,9} when parameterized by the size of the pattern \mathcal{M} . The algorithm is a straightforward extension of a recent result⁸ and is based on the *color-coding* technique¹. Next, we give a faster fixed-parameter algorithm in case the target graph is a tree.

4.1. The Min-CC problem is fixed-parameter tractable

We only sketch the fixed-parameter tractability result. Let G be a graph and k be a positive integer. Recall that a family \mathcal{F} of functions from $\mathbf{V}(G)$ to $\{1, 2, \dots, k\}$ is *perfect* if for any subset $V \subseteq \mathbf{V}(G)$ of k vertices there is a function $f \in \mathcal{F}$ which is injective on V . Let (G, \mathcal{M}) be an instance of the MIN-CC problem, where \mathcal{M} is a motif of size k . Then there is an occurrence of \mathcal{M} in G , say $V \subseteq \mathbf{V}(G)$, that results in a minimum number of connected components. Furthermore, suppose we are provided with a perfect family \mathcal{F} of functions from $\mathbf{V}(G)$ to $\{1, 2, \dots, k\}$. Since \mathcal{F} is perfect, we are guaranteed that at least one function in \mathcal{F} assigns V with k distinct labels. Let $f \in \mathcal{F}$ be such a function. We now turn to defining a dynamic programming table T indexed by vertices of G and subsets of $\{1, 2, \dots, k\}$. For any $v \in \mathbf{V}(G)$ and any $L \subseteq \{1, 2, \dots, k\}$, we define $T_L[v]$ to be the family of all motifs $\mathcal{M}' \subseteq \mathcal{M}$, $|\mathcal{M}'| = |L|$, for which there exists an exact occurrence of \mathcal{M}' in G , say V , such that $v \in V$ and the set of (unique) labels that f assigns to V is exactly L . We need the following lemma⁸.

Lemma 4.1. *For any labeling function $f : \mathbf{V}(G) \rightarrow \{1, 2, \dots, k\}$, there exists a dynamic programming algorithm that computes the table T in $\mathcal{O}(2^{5k}kn^2)$ time.*

Now, denote by \mathcal{P} the set of all pairs $(\mathcal{M}', L') \in \mathcal{M} \times 2^{\{1, 2, \dots, k\}}$ with $|\mathcal{M}'| = |L'|$ such that there exists an exact occurrence of \mathcal{M}' in G , say V' , such that $v \in V'$ and the set of (unique) labels that f assigns to V' is exactly L' . Clearly, $|\mathcal{P}| \leq 2^{2k}$. Furthermore, by resorting to any data structure for searching and inserting that guarantees logarithmic time⁴ (and observing that any two pairs (\mathcal{M}', L') and (\mathcal{M}'', L'') can be compared in $\mathcal{O}(k)$ time), one can construct the set \mathcal{P} in $\mathcal{O}(nk^22^{2k})$ time by running through the table T . Our algorithm now exhaustively considers all subsets of \mathcal{P} of size at most k to find an occurrence of \mathcal{M} in G that results in a minimum number of connected components. The rationale of this approach is that two pairs (\mathcal{M}', L') and (\mathcal{M}'', L'') with $L' \cap L'' = \emptyset$ correspond to non-overlapping occurrences in G . The total time of this latter procedure is certainly upper-bounded by $\sum_{i=1}^k k \binom{2^{2k}}{i} \leq k^2 2^{2k^2}$. Summing up and taking into account the time for computing the table T , the running time for a given $f \in \mathcal{F}$ is $\mathcal{O}(2^{5k}kn^2 + nk^22^{2k} + k^22^{2k^2})$.

According to Alon *et al.*¹, we need to use $\mathcal{O}(2^{\mathcal{O}(k)} \log n)$ functions $f : \mathbf{V}(G) \rightarrow \{1, 2, \dots, k\}$, and such a family \mathcal{F} can be computed in $\mathcal{O}(2^{\mathcal{O}(k)} n \log n)$ time. For each $f \in \mathcal{F}$ we use the above procedure to determine an occurrence of \mathcal{M} in G that results in a minimum number of

connected components. We have thus proved the following.

Proposition 4.1. *The MIN-CC problem is fixed-parameter tractable when parameterized by the size of the motif.*

4.2. A faster fixed-parameter algorithm for trees

We proved in Section 3 that the MIN-CC problem is **APX**-hard even if the target graph is a path. To complement Proposition 4.1, we give here a dynamic programming algorithm for trees that does not rely on the color-coding technique (approaches based on the color-coding technique usually suffer from bad running time performances).

Let (G, \mathcal{M}) be an instance of the MIN-CC problem for trees where both G and \mathcal{M} are built upon a set of colors \mathcal{C} . Let $k = |\mathcal{M}|$ and $q = |\mathcal{C}|$. Furthermore, for ease of exposition, write $\mathbf{V}(G) = \{1, 2, \dots, n\}$ and assume G is rooted at some arbitrary vertex $r(G)$.

Our dynamic programming algorithm is basically an exhaustive search procedure. The basic idea is to store - in a bottom-up fashion - for each vertex i of G and each submotif $\mathcal{M}' \subseteq \mathcal{M}$ that occurs in $T(i)$, *i.e.*, the subtree rooted at i , the minimum number of connected components that results in an occurrence of \mathcal{M}' in $T(i)$. More precisely, for each vertex i of G , we compute two dynamic programming tables $X[i]$ and $Y[i]$. The dynamic programming table $X[i]$ stores all pairs (\mathcal{M}', c) , where $\mathcal{M}' \subseteq \mathcal{M}$ is a submotif and c is a positive integer, such that (1) there exists an occurrence of \mathcal{M}' in $T(i)$ that matches vertex i , (2) the minimum number of connected components of an occurrence of \mathcal{M}' in $T(i)$ that matches vertex i is c . The dynamic programming table $Y[i]$ stores all pairs (\mathcal{M}', c) , where $\mathcal{M}' \subseteq \mathcal{M}$ is a submotif and c is a positive integer, such that (1') there exists an occurrence of \mathcal{M}' in $T(i)$ that *does not match* vertex i , (2') the minimum number of connected components of an occurrence of \mathcal{M}' in $T(i)$ that does not match vertex i is c .

We first claim that both $X[i]$ and $Y[i]$ contain at most k^{q+1} pairs. Indeed, the number of submotifs $\mathcal{M}' \subseteq \mathcal{M}$ is upper-bounded by k^q and any occurrence of any submotif in any subtree of G results in at most k connected components. We now describe how to compute - in a bottom-up fashion - those two dynamic programming tables X and Y .

Let i be an internal vertex of G and suppose that vertex i has s_i sons in the subtree $T(i)$ rooted at i , say $\{i_1, i_2, \dots, i_{s_i}\}$. Notice that $s_i \geq 1$ since i is an internal vertex of G . The entries $X[i]$ and $Y[i]$ are computed with the aid of two auxiliary tables W_i and V_i . Table W_i contains s_i entries, one for

each son of vertex i in the subtree rooted at i , that are defined as follows:

$$\forall 1 \leq j \leq s_i,$$

$$W_i[i_j] = \{(\mathcal{M}', c, 1) : (\mathcal{M}', c) \in X[i_j]\} \cup \{(\mathcal{M}', c, 0) : (\mathcal{M}', c) \in Y[i_j]\}.$$

In other words, we merge $X[i_j]$ and $Y[i_j]$ in $W_i[i_j]$, differentiating the origin of a pair by means of a third element (an integer that is equal to 1 for $X[i_j]$ and 0 for $Y[i_j]$). Clearly, each entry $W_i[i_j]$ contains at most $2k^{q+1}$ triples, and hence table W_i on the whole contains at most $2s_i k^{q+1} \leq 2n k^{q+1}$ triples. Table V_i also contains s_i entries, one for each son of vertex i in the subtree rooted at i , that are computed as follows: $V_i[i_1] = W_i[i_1]$ and

$$\forall 2 \leq j \leq s_i,$$

$$V_i[i_j] = W_i[i_j] \cup \{(\mathcal{M}' \cup \mathcal{M}'', c' + c'', r' + r'') \subseteq \mathcal{M} \times k \times k : \\ (\mathcal{M}', c', r') \in W_i[i_j] \text{ and } (\mathcal{M}'', c'', r'') \in V_i[i_{j-1}]\}.$$

Each entry $V_i[i_j]$ contains at most k^{q+2} triples, and hence table V_i on the whole contains at most $s_i k^{q+2} \leq n k^{q+2}$ triples. All the needed information is stored in $V_i[i_{s_i}]$, and $X[i]$ and $Y[i]$ can be now computed as follows:

$$X[i] = \{(\mathcal{M}', c - r + 1) : (\mathcal{M}', c, r) \in V_i[i_{s_i}] \text{ and } r > 0\}$$

$$Y[i] = \{(\mathcal{M}', c) : (\mathcal{M}', c, 0) \in V_i[i_{s_i}]\}.$$

The two entries $X[i]$ and $Y[i]$ are next filtered according to the following procedure: for each submotif $\mathcal{M}' \subseteq \mathcal{M}$ that occurs in at least one pair of $X[i]$ (resp. $Y[i]$), we keep in $X[i]$ (resp. $Y[i]$) the pair (\mathcal{M}', c) with the minimum c .

The base cases, *i.e.*, vertex i is a leaf, are defined as follows: $X[i] = \{(\lambda(i), 1)\}$ and $Y[i] = \emptyset$. In other words, $X[i]$ contains exactly one pair (\mathcal{M}', c) , where \mathcal{M}' consists in one occurrence of the color associated to vertex i , and $Y[i]$ does not contain any pair. The solution for the MIN-CC problem consists in finding a pair (\mathcal{M}, c) in X or Y with minimum c . If such a pair cannot be found in any entry of both X and Y , then the motif \mathcal{M} does not occur in the tree G .

Proposition 4.2. *The MIN-CC problem for trees is solvable in $\mathcal{O}(n^2 k^{(q+1)^2+1})$ time, where n is the order of the target graph, k is the size of the motif and q is the number of distinct colors.*

The above result is particularly interesting in view of the fact that the MIN-CC problem for trees parameterized by q is **W[1]**-hard⁸.

5. A polynomial-time algorithm for paths with a bounded number of colors

We complement here the results of the two preceding sections by showing that the MIN-CC problem for paths is polynomial-time solvable in case the motif is built upon a fixed number of colors. Observe, however, that each color may still have an unbounded number of occurrences in the motif.

In what follows we describe a dynamic programming algorithm for this case. The basic idea of our approach is as follows. Suppose we are left by the algorithm with the problem of finding an occurrence of a submotif $\mathcal{M}' \subseteq \mathcal{M}$ in the subpath G' of G induced by $\{i, i+1, \dots, j\}$, $1 \leq i < j \leq n$. Furthermore, suppose that any occurrence of \mathcal{M}' in G' results in at least k' connected components. This minimum number of occurrences k' can be computed as follows. Assume that we have found one leftmost connected component C_{left} of the occurrence of \mathcal{M}' in G' and let i_2 , $i \leq i_2 < j$, be the rightmost (according to the natural order of the vertices) vertex of C_{left} . Let \mathcal{M}'' be the motif obtained from \mathcal{M}' by subtracting to each color $c_\ell \in \mathcal{C}$ the number of occurrences of color c_ℓ in the leftmost connected component C_{left} . Then the occurrence of \mathcal{M}' in G' is given by C_{left} plus the occurrence of the motif \mathcal{M}'' in the subpath G'' of G' induced by $\{i_2+1, i_2+2, \dots, j\}$, which results in $k' - 1$ connected components. From an optimization point of view, the problem thus reduces to finding a subpath $\{i_1, i_1+1, \dots, i_2\}$, $i \leq i_1 \leq i_2 < j$, such that the occurrence of the motif \mathcal{M}'' modified according to the colors in $\{i_1, i_1+1, \dots, i_2\}$ in the subpath induced by $\{i_2+1, i_2+2, \dots, j\}$ results in a minimum number of connected components.

Let (G, \mathcal{M}) be an instance of the MIN-CC problem where G is a (vertex-colored) path built upon the set of colors \mathcal{C} . For ease of exposition, write $\mathbf{V}(G) = \{1, 2, \dots, n\}$ and $q = |\mathcal{C}|$. We denote by m_i the number of occurrences of color $c_i \in \mathcal{C}$ in \mathcal{M} . Clearly, $\sum_{c_i \in \mathcal{C}} m_i = |\mathcal{M}|$. We now introduce our dynamic programming table T . Define $T[i, j; p_1, p_2, \dots, p_q]$, $1 \leq i \leq j \leq n$ and $0 \leq p_\ell \leq m_\ell$ for $1 \leq \ell \leq q$, to be the minimum number of connected components in the subpath of G that starts at node i , ends at node j and that covers p_ℓ occurrences of color c_ℓ , $1 \leq \ell \leq q$. The base conditions are as follows:

- for all $1 \leq i \leq j \leq n$, $T[i, j; 0, 0, \dots, 0] = 0$ and $T[i, i; p_1, p_2, \dots, p_q] = \infty$ if $\sum_{1 \leq \ell \leq q} p_\ell > 1$,
- for all $1 \leq i \leq n$, $T[i, i; p_1, p_2, \dots, p_q] = \infty$ if $\sum_{1 \leq \ell \leq q} p_\ell = 1$ and $\lambda(i) \neq c_\ell$ and $p_\ell = 1$, and $T[i, i; p_1, p_2, \dots, p_q] = 1$ if $\sum_{1 \leq \ell \leq q} p_\ell = 1$ and $\lambda(i) = c_\ell$ and $p_\ell = 1$.

The entry $T[i, j; p_1, p_2, \dots, p_q]$ of the dynamic programming table T can be computed by the following recurrence

$$T[i, j; p_1, p_2, \dots, p_q] = \min_{i \leq i_1 \leq i_2 < j} T[i_2 + 1, j; p'_1, p'_2, \dots, p'_q] + 1 \quad (1)$$

where each $p'_\ell \geq 0$ is equal to p_ℓ minus the number of occurrences of color c_ℓ in the subpath of G induced by the vertices $\{i_1, i_1 + 1, \dots, i_2\}$. The optimal solution is clearly stored in $T[1, n; p_1, p_2, \dots, p_q]$.

We claim that our dynamic programming table T contains $\mathcal{O}(n^{q+2})$ entries. Indeed, there are q colors in \mathcal{M} , each color $c_i \in \mathcal{C}$ has at most n occurrences in G and we have $\mathcal{O}(n^2)$ subpaths in G to consider. We now turn to evaluating the time complexity for computing $T[i, j; p_1, p_2, \dots, p_q]$. Assuming each entry $T[i', j'; p'_1, p'_2, \dots, p'_q]$ with $i \leq i' \leq j' \leq j$ and $|j' - i'| < |j - i|$ has already been computed, $T[i, j; p_1, p_2, \dots, p_q]$ is obtained by taking a minimum number among $\mathcal{O}(|j - i + 1|^2) = \mathcal{O}(n^2)$ numbers, and hence is $\mathcal{O}(n^2)$ time. We have thus proved the following.

Proposition 5.1. *The MIN-CC problem for paths is solvable in $\mathcal{O}(n^{q+4})$ time, where n is the number of vertices and q is the number of colors in \mathcal{C} .*

As an immediate consequence of the above proposition, the MIN-CC problem is polynomial-time solvable in case the motif \mathcal{M} is built upon a fixed number of colors and the target graph G is a path.

6. Hardness of approximation for trees

We investigate in this section approximation issues for restricted instances of the MIN-CC problem. Unfortunately, as we shall now prove, it turns out that, even if \mathcal{M} is a set and G is a tree, the MIN-CC problem cannot be approximated within ratio $c \log n$ for some constant $c > 0$, where n is the size of the target graph G . As a side result, we prove that the MIN-CC problem is **W[2]**-hard when parameterized by the number of connected components of the occurrence of \mathcal{M} in the target graph G .

At the core of our proof is an L-reduction¹² from the SET-COVER problem. Let I be an arbitrary instance of the SET-COVER problem consisting of a universe set $X(I) = \{x_1, x_2, \dots, x_n\}$ and a collection of sets $\mathcal{S}(I) = S_1, S_2, \dots, S_m$, each over $X(I)$. For each $1 \leq i \leq m$, write $t_i = |S_i|$ and denote by $e_j(S_i)$, $1 \leq j \leq t_i$, the j -th element of S_i . For ease of exposition, we present the corresponding instance of the MIN-CC problem as a rooted tree G . We construct the tree G as follows (see Fig. 1). Define a root r and vertices S'_1, S'_2, \dots, S'_m such that each vertex S'_i is connected to the root r . For each S'_i define the subtree $G(S'_i)$ rooted at S'_i

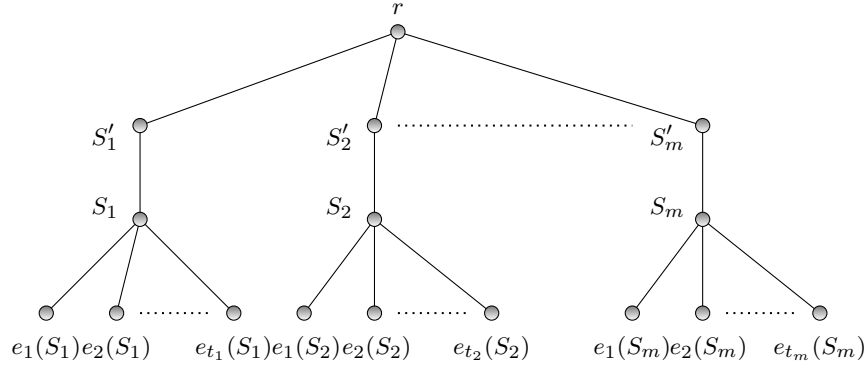


Figure 1. Construction of the corresponding instance of the MIN-CC problem.

as follows: each vertex S'_i has a unique child S_i and each vertex S_i has children $e_1(S_i), e_2(S_i), \dots, e_{t_i}(S_i)$. The set of colors \mathcal{C} is defined as follows: $\mathcal{C} = \{c(S_i) : 1 \leq i \leq m\} \cup \{c(x_j) : 1 \leq j \leq n\} \cup \{c(r)\}$. The coloring mapping $\lambda : \mathbf{V}(G) \rightarrow \mathcal{C}$ is defined by: $\lambda(S_i) = \lambda(S'_i) = c(S_i)$ for $1 \leq i \leq m$, $\lambda(x_j) = c(x_j)$ for $1 \leq j \leq n$ and $\lambda(r) = c(r)$. The motif \mathcal{M} is the set defined as follows: $\mathcal{M} = \{c(S_i) : 1 \leq i \leq m\} \cup \{c(x_i) : 1 \leq i \leq n\} \cup \{c(r)\}$.

Proposition 6.1. *For any instance I of the SET-COVER problem, there exists a solution of size h for I , i.e., a subset $\mathcal{S} \subseteq \mathcal{S}(I)$, $|\mathcal{S}| = h$, such that $\bigcup_{S_i \in \mathcal{S}} S_i = X$, if and only if then there exists an occurrence of \mathcal{M} in G that results in $h + 1$ connected components.*

It is easily seen that the above reduction is an L-reduction¹². It is known that SET-COVER cannot be approximated within ratio $c \log n$ for some constant $c > 0$ ¹⁴. Then it follows that there exists a constant $c' > 0$ such that the MIN-CC for trees cannot be approximated within performance ratio $c' \log n$, where n is the number of vertices in the target graph.

As a side result, we also observe that the above reduction is a parameterized reduction. Since the SET-COVER is **W[2]**-hard when parameterized by the size of the solution¹³, the following result holds.

Corollary 6.1. *The MIN-CC problem for trees is **W[2]**-hard when parameterized by the number of connected components of the occurrence of the motif in the graph.*

7. An exact algorithm for trees

We proved in Section 4 that the MIN-CC for trees is solvable in $\mathcal{O}(n^2 k^{(q+1)^2+1})$ time, where n is the order of the target tree, k is the size of the motif and q is the number of distinct colors. We propose here a new algorithm for this special case, which turns out not to be a fixed-parameter algorithm but has a better running time in case the motif k is not that small compared to the order n of the target graph. More precisely, we give an algorithm for solving the MIN-CC problem for trees that runs in $\mathcal{O}(n^2 2^{\frac{2n}{3}})$, where n is the order of the target tree. Due to space constraints, we skip the proof details.

Let T be the target tree. For any vertex x of T , denote by $T(x)$ the subtree of T rooted at x . The first step of our algorithm splits the target tree in a *balanced way*, so that T is rooted at a vertex r having children, r_1, r_2, \dots, r_h such that none of the trees $T(r_i)$, $1 \leq i \leq h$, has order greater than $\lceil \frac{n}{2} \rceil$. Such a vertex r can be found in $\mathcal{O}(n^2)$ time. We then construct two disjoint subsets R_1 and R_2 of r_1, \dots, r_h with the property that

$$\frac{1}{3}|T| \leq \sum_{r_i \in R_1} |T(r_i)| \leq \lceil \frac{1}{2}|T| \rceil \quad \text{and} \quad \lceil \frac{1}{2}|T| \rceil \leq \sum_{r_i \in R_2} |T(r_i)| = \frac{2}{3}|T|$$

Given V' a subset of nodes of V , we say that V' does not violate \mathcal{M} if the multiset of colors $C(V')$ is a subset of \mathcal{M} . Given a subtree T' of T , we define a *partial solution* F of MIN-CC over instance (T', \mathcal{M}) as a set of connected components of T' that does not violate the multiset \mathcal{M} .

The algorithm computes an optimal solution for MIN-CC by first computing all the partial solutions S_1 over instance (R_1, \mathcal{M}) and all the partial solutions S_2 over instance (R_2, \mathcal{M}) and then merging a partial solution F_1 of S_1 and a partial solution F_2 of S_2 into a feasible solution for the MIN-CC over instance (T, \mathcal{M}) . Since there are $2^{\frac{n}{2}}$ and $2^{\frac{2n}{3}}$ possible subsets of vertices of R_1 and R_2 respectively, it follows that the set of partial solutions over instance (R_1, \mathcal{M}) , (R_2, \mathcal{M}) can be computed in time $\mathcal{O}(2^{\frac{n}{2}})$ and $\mathcal{O}(2^{\frac{2n}{3}})$ respectively. Then set S_1 is ordered and by binary search we can find in time $\mathcal{O}(n \log 2^{\frac{n}{2}}) = \mathcal{O}(n^2)$ a solution F_1 of S_1 that, merged to a solution F_2 of S_2 , produces a feasible solution of MIN-CC over instance (T, \mathcal{M}) . Since $|S_2| = \mathcal{O}(2^{\frac{2n}{3}})$, it follows that the overall time complexity of the algorithm is $\mathcal{O}(n^2 2^{\frac{2n}{3}})$.

8. Conclusion

We mention here some possible directions for future works. First, approximation issues of the MIN-CC problem are widely unexplored. In particular,

is the MIN-CC problem for paths approximable within a constant ? Also, most parameterized complexity issues are to be discovered. Of particular importance: is the MIN-CC problem for paths $\mathbf{W}[1]$ -hard when parameterized by the number of connected components in the occurrence of the motif in the target graph ?

Bibliography

1. N. Alon, R. Yuster, and U. Zwick. Color coding. *Journal of the ACM*, 42(4):844–856, 1995.
2. P. Bonsma. Complexity results for restricted instances of a paint shop problem. Technical Report 1681, Dept of Applied Maths, Univ. of Twente, 2003.
3. P. Bonsma, T. Epping, and W. Hochstättler. Complexity results on restricted instances of a paint shop problem for words. *Discrete Applied Mathematics*, 154(9):1335–1343, 2006.
4. T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to algorithms*. McGraw Hill, New York, 2001.
5. Y. Deville, D. Gilbert, J. Van Helden, and S.J. Wodak. An overview of data models for the analysis of biochemical pathways. *Briefings in Bioinformatics*, 4(3):246–259, 2003.
6. R. Diestel. *Graph Theory*. Number 173 in Graduate texts in Mathematics. Springer-Verlag, second edition, 2000.
7. R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
8. M. Fellows, G. Fertin, D. Hermelin, and S. Vialette. Sharp tractability borders for finding connected motifs in vertex-colored graphs. In *Proc. 34th Int. Colloquium on Automata, Languages and Programming (ICALP)*, 2007. To appear.
9. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.
10. T. Ideker, R.M. Karp, J. Scott, and R. Sharan. Efficient algorithms for detecting signaling pathways in protein interaction networks. *Journal of Computational Biology*, 13(2):133–144, 2006.
11. V. Lacroix, C.G. Fernandes, and M.-F. Sagot. Motif search in graphs: application to metabolic networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 3(4):360–368, 2006.
12. C.H. Papadimitriou and M. Yannakakis. Optimization, approximation and complexity classes. *J. of Computer and System Sciences*, 43:425–440, 1991.
13. A. Paz and S. Moran. Non deterministic polynomial optimization problems and their approximations. *Theoretical Computer Science*, 15:251–277, 1981.
14. R. Raz and S. Safra. A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP. In *Proc. 29th Ann. ACM Symp. on Theory of Comp. (STOC)*, pages 475–484, 1997.
15. W. Hochstättler T. Epping and P. Oertel. Complexity results on a paint shop problem. *Discrete Applied Mathematics*, 136(2-3):217–226, 2004.