The Maximum Colorful Arborescence problem: How (computationally) hard can it be?

Guillaume Fertin, Julien Fradin, Géraldine Jean

Version of Record: https://www.sciencedirect.com/science/article/pii/S0304397520306617
Manuscript_a34298a03458e98b2257ea696517e280

The Maximum Colorful Arborescence Problem: How (Computationally) Hard can it be ? ☆

Guillaume Fertin, Julien Fradin, Géraldine Jean*

Université de Nantes, CNRS, LS2N, F-44000, Nantes

Abstract

Given a vertex-colored arc-weighted directed acyclic graph G, the Maximum Colorful Subtree problem (or MCS) aims at finding an arborescence of maximum weight in G, in which no color appears more than once. The problem was originally introduced in [1] in the context of *de novo* identification of metabolites by tandem mass spectrometry. However, a thorough analysis of the initial motivation shows that the formal definition of MCS should be amended, since the input graph G actually possesses extra properties, which have been unexploited so far. This leads us to describe in this paper a more precise model that we call Maximum Colorful Arborescence (MCA), which we extensively study in terms of algorithmic complexity. In particular, we show that exploiting the implied Color Hierarchy Graph of the input graph G can lead to exact polynomial algorithms and approximation algorithms. We also develop Fixed-Parameter Tractable (FPT) algorithms for the problem parameterized by the "dual parameter" $\ell_{\mathcal{C}}$, defined as the minimum number of vertices of G which are *not* kept in the solution.

Keywords: Complexity, FPT algorithms, approximation algorithms, Maximum colorful arborescence, tandem mass spectometry

1. Introduction

Metabolites are small molecules that are involved in cellular reactions, most of them remaining unknown to this date [2]. Consequently, identifying molecular structures of metabolites is a key problem in biology [3, 4], and in particular in drug design [5, 6]. Tandem mass spectrometry is one of the most commonly used technologies to achieve

☆A preliminary version of this work appeared in Proceedings of the 14th Annual Conference on Theory and Applications of Models of Computation (TAMC '17), volume 10185 of LNCS, pages 216–230, under the title "Algorithmic Aspects of the Maximum Colorful Arborescence Problem". This work was partially supported by the PHC PROCOPE project "Efficient Algorithms for Hard Computational Problems in Mass Spectrometry" (37748TL).

*Corresponding author

Email addresses: guillaume.fertin@univ-nantes.fr (Guillaume Fertin), julien.fradin@univ-nantes.fr (Julien Fradin), geraldine.jean@univ-nantes.fr (Géraldine Jean)

this goal. In a tandem mass spectrometry experiment, a metabolite is fragmented into smaller molecules. The mass spectrometer then outputs a fragmentation spectrum that consists of a series of peaks where, ideally, each peak corresponds to the mass of one of the generated fragments. If we are able to "explain" the spectrum by finding the molecule which corresponds to each peak it contains, then the input metabolite can in turn be infered. Such an identification can be achieved by comparison with some reference database(s) [7]; however, the databases at hand are largely incomplete [2, 8, 9]. This is why *de novo* interpretation of the fragments, directly from the spectra, is a promising alternative.

Determining the chemical formula of metabolites is generally used as a first step in the identification of their molecular structure. In [1], Böcker *et al.* initiated such study, where the problem of *de novo* identifying the chemical formulas of metabolites from tandem mass spectrometry spectra was formally modeled by the Maximum Colorful Subtree (or MCS) problem. Let μ be an unknown metabolite and s_μ a tandem mass spectrum of μ. Intuitively, a solution of MCS represents the best possible "fragmentation scenario", called a fragmentation tree, of μ. Böcker *et al.* showed that computing the fragmentation tree of μ allows to determine the chemical formula of the studied metabolite [1]. Further studies then showed how to use these fragmentation trees in order to determine the molecular structures of these metabolites [10, 11, 12, 13, 14].

In the following, we describe the main ideas behind MCS. First, for each peak p in s_μ representing a mass, a set of chemical formulas is generated such that the mass of any molecule having one of those chemical formulas lies in the same range as p. A directed acyclic graph (DAG) $G = (V, A)$ is then created as follows: every node $v \in V$ represents a chemical formula; two nodes u and v are linked by an arc (u, v) if one molecule, whose chemical formula is represented by vertex v, is possibly the result of the fragmentation of another molecule whose chemical formula is represented by vertex u; each vertex possesses a color corresponding to its mass (or better said, its mass range). A weight function $w : A \to \mathbb{R}$ is also assigned to the arcs of G. Informally, weights correspond to some degree of confidence concerning the fragmentation of a molecule into its sub-molecule; it is also important to note that in most applications, the weight function w is logarithmic, and thus arc weights in G may be negative. Note finally that in such a graph, there exists a unique vertex of indegree 0 (that can be seen as the "root" of G), whose color is also unique: this particular vertex represents one possible candidate chemical formula for metabolite μ. Now the MCS problem, introduced in [1], is defined as follows: given a DAG $G = (V, A)$, a set $\mathcal{C}$ of colors, a coloring function $\text{col} : V \to \mathcal{C}$ and a weight function $w : A \to \mathbb{R}$, find a subtree T of G such that (1) no two vertices of T carry the same color (we then say that T is *colorful*) and (2) T is of maximum weight (where the weight of T is the sum of the weights of the arcs it contains).

However, a finer-grained analysis shows that modeling the initial problem as the MCS problem does not completely reflect the precise structure of the input. First, it is easy to see that G is not *any* DAG: more precisely, as discussed above, it has a unique root r having indegree 0. Moreover, let us define $\mathcal{H}(G)$ as the directed graph that is built from G as follows: $V(\mathcal{H}(G))$ is the set $\mathcal{C}$ of colors used to color $V(G)$, and there is an arc from c to c' in $\mathcal{H}(G)$ if there is an arc in G from a vertex of color c to a

vertex of color c'. Since vertices are colored according to the masses of the molecules they represent and since an arc (x, y) represents the fragmentation of a molecule of chemical formula x into a molecule of chemical formula y, there necessarily exists a partial order among colors, which implies that $\mathcal{H}(G)$ is necessarily a DAG too; therefore $\mathcal{H}(G)$ is called the *Color Hierarchy Graph* of G. Finally, by the nature of the initial problem, the output tree T must necessarily contain the root r. Thus, T is formally an arborescence, i.e., a directed graph $T = (V_T, A_T)$ with a designated root r such that there exists only one path from r to any node $v \in V_T$. This leads us to reformulate the MCS problem into the following Maximum Colorful Arborescence (or MCA) problem, which better reflects the initial motivation.

Maximum Colorful Arborescence (MCA)

• **Input**: A DAG $G = (V, A)$ rooted in some vertex r, a set $\mathcal{C}$ of colors, a coloring function $\text{col} : V \to \mathcal{C}$ s.t. $\mathcal{H}(G)$ is a DAG and an arc weight function $w : A \to \mathbb{R}$.

• **Output**: A colorful arborescence $T = (V_T, A_T)$ rooted in r and of maximum weight $w(T) = \sum_{a \in A_T} w(a)$.

We say that an optimization problem Q is FPT (for Fixed-Parameter Tractable) with respect to a given parameter k if it can be exactly solved in time $\mathcal{O}(f(k) \cdot \text{poly}(|I|))$ for some computable function f and any instance I of Q, i.e., if the exponential part of its complexity depends only on k. If k is small in practice, designing Fixed-Parameter Tractable algorithms is of interest since they are exact and possibly run in reasonable time, even for large inputs. For more details on the theory of fixed-parameter tractability, we refer the reader to [15]. Now, because the definition of MCA is more accurate, it seems interesting to provide a detailed analysis of the computational complexity of the problem, as done in this paper. Indeed, studying MCA contributes to a better understanding of the initial biological problem. In particular, we will see that the fact that $\mathcal{H}$ is a DAG can be positively exploited in some situations. Moreover, since any instance of MCA is also an instance of MCS, any positive result (such as polynomial-time, approximation and FPT algorithms) for MCS also applies to MCA – with a time complexity and/or approximation ratio that may even be improved for MCA. Besides, a negative result for MCS does not necessarily imply the same result for MCA. In this paper, we study MCA under an algorithmic viewpoint: a first goal is to distinguish tractable instances from intractable ones; a second one is to provide new polynomial and FPT algorithms for the problem.

This paper, which is an extended version of [16], is organized as follows. In Section 2, we introduce notations that will be used throughout the paper. We then show in Section 3 that MCA remains hard even when the input instances consist in very specific arborescences. In this extended version, we also propose approximation algorithms for such instances, that essentially take advantage of the fact that $\mathcal{H}$ is a DAG. Using this property of $\mathcal{H}$, we describe in Section 4 a new range of instances that are polynomial-time solvable, and provide new FPT algorithms. Finally, in Section 5,

we present FPT algorithms for MCA parameterized by parameter $\ell_{\mathcal{C}}$, defined as the minimum number of vertices *not* present in the solution. In particular, we show in this extended version a new FPT algorithm whenever (1) G is a tree and (2) the arc weights are uniform.

2. Preliminaries

Notations. For any positive integer k, we use the notation $[k] = \{1, 2, \ldots, k\}$. For any vertex-colored and arc-weighted DAG $G = (V, A)$ given as input of MCA, we let $n = |V|$ and $m = |A|$. Since the input DAG of MCA instances is rooted in a single vertex, we always denote by r the root of G. For any $v \in V$, $N^+(v)$ denotes the set of outneighbors of v (thus excluding v) and, for any $V' \subseteq V$, $N^+(V') = \bigcup_{v' \in V'} N^+(v')$ is the set of outneighbors of all $v' \in V'$ (excluding V'). For any $v \in V$, $d^+(v)$ (resp. $d^-(v)$) denotes the outdegree (resp. indegree) of v. Moreover, for all $V' \subseteq V$ and $v \in V'$, $G_v[V']$ denotes the induced DAG of $G[V']$ that is rooted in v. When G is an arborescence, for any vertex $v \in V$ we let $f(v)$ be the father (i.e., the unique inneighbor) of v in G. For any subset V' of V, $\mathrm{col}(V')$ denotes the multiset of colors assigned to the vertices of V' and $\mathrm{col}^*(V')$ its underlying set. We say that V is *colorful* when $\mathrm{col}(V) = \mathrm{col}^*(V)$ and that G is *colorful* when $V(G)$ is itself colorful. For a vertex $v \in V$, we let $d(r, v)$ denote the oriented distance from r to v in (the unweighted version of) G, that is the minimum number of arcs needed to reach v from r in G. The Color Hierarchy Graph of G is denoted by $\mathcal{H}(G) = (\mathcal{C}, A_{\mathcal{C}})$ or, when clear from the context, simply $\mathcal{H}$.

The problem MCA$^+$ denotes the restriction of MCA to DAGs with positive weights, whereas UMCA denotes the restriction of MCA$^+$ to instances having unit arc weights, i.e., $w(a) = 1$ for all $a \in A$. Note that the root r of G is a trivial solution to any instance which only contains negative arcs; therefore, there is no need to define a restriction of MCA to DAGs with only negative arc weights. The problem MCA-x is the restriction of MCA in which any color $c \in \mathcal{C}$ appears at most x times in $\mathrm{col}(V)$. Finally, we can also constrain the input instances of MCA both on the weights and on the maximal number of occurrences of a color, and thus combine any the three abovementioned variants. In that case, the problem will be naturally denoted by combining the corresponding notations. For instance, the UMCA-x problem considers input DAGs with positive and uniform weights, and in which any color appears at most x times in G.

Note that although G, the solution arborescence T and the Color Hierarchy Graph $\mathcal{H}$ are by definition directed, in the rest of the paper we will often, for simplicity and when clear from the context, refer to the underlying undirected graph of some graph H (rather than H itself). For instance, when we talk about MCA "in trees", we actually mean that the underlying undirected graph of G is a tree.

Two parameters will be of importance in the algorithmic study of MCA: $\ell_{\mathcal{C}} = n - |\mathcal{C}|$ is the minimum number of vertices that are *not* part of the solution, and s is the minimum number of arcs that need to be removed from $\mathcal{H}$ in order to turn it into an arborescence.

Previous results. We summarize here known results about MCS, and also note that every result mentioned below concerning MCS also applies to MCA. Indeed, MCA being a particular case of MCS, any positive result for MCS also holds for MCA. Moreover, for all negative results below, the MCS instances that are built in the corresponding proofs turn out to be either MCA instances themselves, or can easily be transformed into such instances.

MCS is known to be NP-hard even when every arc weight is equal to 1 [1], and it can be seen that the result also applies to UMCA. MCS is also APX-hard on binary trees [17, 18], a result that also applies to UMCA-2. However, the problem is FPT parameterized by $|\mathcal{C}|$, by a dynamic programming algorithm that runs in $\mathcal{O}^*(3^{|\mathcal{C}|})$ time and uses $\mathcal{O}^*(2^{|\mathcal{C}|})$ space [19, 1].

Theorem 1 from [17], which shows hardness results for MCS, is directly applicable to MCA-1, since the MCS instance constructed in the reduction turns out to be an MCA-1 instance. Besides, it can be slighlty modified, while preserving the results, into an MCA-2 instance in which G is a tree. Thus we derive from Theorem 1 in [17] the following results: MCA-1 and MCA-2 in trees both are W[1]-hard when parameterized by the weight w of the solution ; MCA-1 is W[1]-hard when parameterized by $\ell_{\mathcal{C}} = n - |\mathcal{C}|$; unless $\mathsf{P} \neq \mathsf{NP}$, there is no polynomial-time approximation algorithm achieving a ratio of $\mathcal{O}(n^{\frac{1}{2}-\epsilon})$ with $\epsilon > 0$ for both MCA-1 and MCA-2 in trees.[1]

MCA has also been studied in [20], under a parameterized complexity point of view. It has been proved that MCA is FPT in the number of vertices of indegree at least 2 in $\mathcal{H}$ and W[2]-hard parameterized by the treewitdh $t_{\mathcal{H}}$ of $\mathcal{H}$. An FPT algorithm in $\ell_{\mathcal{C}} + t_{\mathcal{H}}$ is also given.

Our results. The main results obtained in this paper are summarized in Tables 1 and 2. They will be developed in the following sections.

3. MCA in Trees

In this section, we focus on MCA in the case where the input graph G is a tree, aiming at determining which tree structures lead to (in)tractable (resp. (in)approximable) results. We start with the following theorem, that applies to the general case of trees.

Theorem 1. *For any $\delta < 1$,* UMCA *in trees cannot be approximated within $2^{\log^{\delta} n}$ in polynomial time, unless* $\mathsf{NP} \subseteq \mathsf{DTIME}[2^{\text{poly} \log n}]$.

Proof. Dondi *et al.* introduced the Maximum Level Motif (or MLM) problem [18], a maximization variant of the Graph Motif problem [21] dealing with colorful motifs on trees. Besides, MLM incorporates the notion of a *leveled coloring function* $\text{col}' : V \to \mathcal{C}$ for which two vertices can have the same color only if they are at the same distance from the root. The formal definition of MLM is given below.

[1] As a side note, we point out an error in the inapproximation ratio given in Theorem 1 in [17], which should be $\mathcal{O}(n^{\frac{1}{2}-\epsilon})$ instead of $\mathcal{O}(n^{1-\epsilon})$.

Restriction		Constraint	Variant	Result
	on G	tree	UMCA	no $2^{log^{\delta} n}$ approx., $\delta < 1$ (Th. 1) $\mathcal{O}(\frac{n}{\log n})$ approx. in binary trees (Prop. 2)
		superstar	UMCA-2	APX-hard (Th. 4) 2-approx. (Prop. 5)
		comb-graph	UMCA-2	no $\mathcal{O}(n^{\frac{1}{3}-\epsilon})$ approx. (Th. 8)
			UMCA	$\mathcal{O}(n^{\frac{1}{2}-\epsilon})$ approx. (Prop. 11)
		caterpillar	MCA	belongs to P (Prop. 12)
	on $\mathcal{H}$	tree	MCA	belongs to P (Th. 15)

Table 1: Overview of the approximation and exact results presented in this paper for the MCA problem and its variants. Here, n is the number of vertices in G. Positive results are obtained based on the fact that $\mathcal{H}$ is a DAG.

Variant	Restriction on G	Result
MCA	-	$\mathcal{O}^*(2^{\ell_c + m^-})$ (Prop. 19)
MCA$^+$	-	$\mathcal{O}^*(2^{\ell_c})$ (Cor. 20)
MCA	G is a tree	$\mathcal{O}^*(2^{\ell_c})$ (Prop 21)
MCA$^+$	G is a tree	$\mathcal{O}^*(1.62^{\ell_c})$ (Prop. 22)
UMCA	G is a tree	$\mathcal{O}^*(1.33^{\ell_c})$ (Prop. 23)
UMCA-2	-	no $\mathcal{O}^*((2-\epsilon)^{\ell_c})$ algorithm (Thm. 24)

Table 2: Overview of the parameterized results presented in this paper for the MCA problem and its variants. Here, n is the number of vertices in G, $\mathcal{C}$ is the color set of G, $\ell_{\mathcal{C}} = n - |\mathcal{C}|$ and m^- is the number of arcs with negative weight in G.

Maximum Level Motif (MLM)

• **Input**: A rooted tree $H = (V, E)$, a color set $\mathcal{C}$, a leveled coloring function $\text{col}' : V \to \mathcal{C}$.

• **Output**: A maximum cardinality subset $V' \subseteq V$ such that the induced subgraph $H[V']$ is connected and colorful.

Let I be any instance of MLM. We construct an instance I' of MCA as follows:

graph G is built on V, and each edge in H is changed into an arc in G, between the same vertices, such that each arc is oriented from parent to child; clearly, G is a tree. We let $w(a) = 1$ for any arc, and we also apply the same coloring function col', given as input of MLM, to color the vertices of G. Since col' is a leveled coloring function, the colors in $\mathcal{C}$ are partially ordered and therefore $\mathcal{H}$ is a DAG. Thus, I' is a correct UMCA instance. We now show that there exists a solution V' of cardinality at least k for MLM iff there exists a colorful arborescence $T = (V_T, A_T)$ such that $w(T) \geq k-1$ in G.

($\Rightarrow$) Suppose there exists a solution V' of MLM such that $|V'| \geq k$. Let T be the spanning arborescence of V' in G, with $V_T = V'$. Trivially, T is colorful and of weight at least $k-1$. If $r \notin V_T$, we search for a vertex $x \in V_T$ such that $d(r,x) = \min\{d(r,u) : u \in V_T\}$. Let $V_{r,x}$ (resp. $A_{r,x}$) be the set of vertices (resp. arcs) in the path from r to x in G. We construct a new arborescence $T' = (V_{T'}, A_{T'})$, with $V_{T'} = V_T \cup V_{r,x}$ and $A_{T'} = A_T \cup A_{r,x}$. According to col', $V_{r,x}$ is colorful and each vertex in $V_{r,x}$ has a different color from any of the vertices in V_T. Thus, $V_{T'}$ is colorful.

($\Leftarrow$) Suppose there exists a colorful arborescence $T = (V_T, A_T)$ of weight at least $k-1$ in G. Then, we choose $V' = V_T$. Trivially, V' is colorful and $|V'| \geq k$.

Dondi et al. proved that, under the condition that $\mathsf{NP} \subseteq \mathsf{DTIME}[2^{\mathrm{poly}\log n}]$, MLM cannot be approximated within $2^{\log^{\delta} n}$, $\delta < 1$, in polynomial time [18]. By linearity of the above reduction, we reach the same conclusion for UMCA in trees. □

We just showed that UMCA is highly inapproximable in trees. However, an approximation algorithm for UMCA can be obtained in binary trees. Indeed, note that a path from the root to any other vertex is always colorful as $\mathcal{H}$ is a DAG. Since the eccentricity of the root in any such tree is at least $\lceil \log n \rceil$, and since any solution for UMCA in trees has weight less then or equal to $n-1$, we obtain the following result.

Proposition 2. UMCA *in binary trees can be approximated within ratio* $\mathcal{O}(\frac{n}{\log n})$.

From Theorem 1, it seems natural to further restrict the structure of the input tree in order to draw the line between tractable and intractable cases. We begin by the particular case where G is a star, i.e., a tree with one internal vertex, say z, that is connected to all other vertices. In case the internal vertex of G is the root r, an optimal solution to MCA is obtained as follows: for every color $c \in \mathcal{C}$, consider all arcs from r to a vertex of color c, and keep the one with maximum weight if it is positively weighted, or discard it otherwise. In case $z \neq r$, z is the unique outneighbor of r. Since r must belong to the solution, we proceed in two steps. First, consider the star $G[V \setminus \{r\}]$ and apply the same strategy as before with z playing the role of r. We obtain a partial solution of positive total weight containing at least z. Second, we try to add the arc (r,z) to the partial solution. Note that the arc (r,z) can be positively or negatively weighted. Therefore, the final solution is obtained by adding the arc (r,z) to the partial solution if the total weight is still positive. Otherwise, it means that the arc (r,z) is too negatively weighted and the final solution is restricted to the vertex r. We then get the following easy result, which will prove useful in some of our FPT algorithms in Section 4.

Proposition 3. MCA *in stars is polynomial-solvable.*

Superstars, defined as rooted trees of height 2, are a natural extension of stars. However, in that case the MCA problem turns out to be hard, as shown by the following result.

Theorem 4. UMCA-2 *is* APX*-hard, even if G is a superstar.*

Proof. The proof is by reduction from Max-2-SAT(3), which is known to be APX-hard [22]. It can be seen as an extension of proof of Theorem 1 in [1].

Max-2-SAT(3)

• **Input**: A set $X = \{x_1, x_2 \dots x_p\}$ of variables, a CNF-formula ϕ on a set of size-2 clauses $C_1, C_2 \dots C_q$ built from X, such that each variable occurs in at most 3 clauses.

• **Output**: A boolean assignment $\beta : X \to \{\texttt{true}, \texttt{false}\}$ that maximizes the number of satisfied clauses from ϕ.

Recall that $f(v)$ denotes the unique inneighbor of any $v \in V$ as G is a tree. For every $j \in [q]$, let $l_{j,1}$ and $l_{j,2}$ be the two literals that appear in clause C_j. The reduction is as follows: for any instance of Max-2-SAT(3), we create a directed superstar $G = (V, A)$ that we can see as a three-leveled graph (see Figure 1). The root r is at level 1, two vertices v_i and $\overline{v_i}$ are created for every $i \in [p]$ at level 2, and two vertices $C_{j,1}, C_{j,2}$ are created for every $j \in [q]$ at level 3. There exists an arc from r to every level-2 vertex. For all $i \in [p]$ and $j \in [q]$, there exists an arc from v_i (resp. $\overline{v_i}$) to $C_{j,1}$ if $l_{j,1} = x_i$ (resp. $l_{j,1} = \overline{x_i}$) or from v_i (resp. $\overline{v_i}$) to $C_{j,2}$ if $l_{j,2} = x_i$ (resp. $l_{j,2} = \overline{x_i}$). The intuition is that an arc $(v_i, C_{j,\_})$ (resp. $(\overline{v_i}, C_{j,\_})$) appearing in an arborescence represents the situation where setting $x_i = \texttt{true}$ (resp. $x_i = \texttt{false}$), modeled by vertex v_i (resp. $\overline{v_i}$), satisfies clause C_j. The coloring function on $V(G)$ is defined as follows: the root r is assigned a unique color; for all $i \in [p]$, vertices v_i and $\overline{v_i}$ are assigned the same color $c(v_i)$; for all $j \in [q]$, vertices $C_{j,1}$ and $C_{j,2}$ are assigned the same color $c(C_j)$. Clearly, each color occurs at most twice in G, and the coloring function is partially ordered because any two vertices having the same color lie at the same level. Finally, every arc in G is assigned a weight of 1.

We now show that there exists a boolean assignment β for X that satisfies at least k clauses in ϕ iff there exists a colorful arborescence $T = (V_T, A_T)$ of weight $w(T) \geq p+k$ in G.

($\Rightarrow$) Suppose there exists an assignment β for X that satisfies at least k clauses of ϕ. We define:

$$S_T = \{v_i : i \in [p] \ s.t. \ x_i = \texttt{true} \ in \ \beta\}$$

and

$$S_F = \{\overline{v_i} : i \in [p] \ s.t. \ x_i = \texttt{false} \ in \ \beta\}$$

Then, we define:

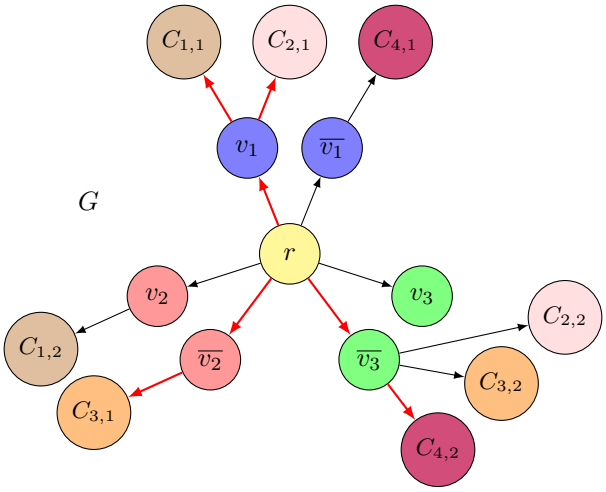

Figure 1: Construction of an instance of UMCA-2 from the following MAX-2-SAT(3) instance: $\phi = (x_1 \vee x_2) \wedge (x_1 \vee \overline{x_3}) \wedge (\overline{x_2} \vee \overline{x_3}) \wedge (\overline{x_1} \vee \overline{x_3})$. By definition, all arc weights are equal to 1, and are not represented for clarity. The assignment $x_1 = \texttt{true}$, $x_2 = \texttt{false}$ and $x_3 = \texttt{false}$ satisfies ϕ and corresponds to the UMCA-2 solution T in G with bold red arcs.

$$\begin{aligned} V_T = &\{r\} \cup S_T \cup S_F \\ &\cup \{C_{j,1} : j \in [q] \ s.t. \ f(C_{j,1}) \in (S_T \cup S_F)\} \\ &\cup \{C_{j,2} : j \in [q] \ s.t. \ f(C_{j,2}) \in (S_T \cup S_F) \ and \ f(C_{j,1}) \notin (S_T \cup S_F)\} \end{aligned}$$

We denote by T the spanning arborescence of V_T. By construction, there cannot exist $j \in [q], h \in \{1,2\}$ such that $C_{j,h} \in V_T$ and $f(C_{j,h}) \notin V_T$. Thus, T is connected. Moreover, since β satisfies at least k clauses, there exists at least k distinct vertices of type $C_{j,h}$ that belong to T, in addition to the p vertices in $S_T \cup S_F$ and the root r. Besides, T is necessarily colorful and $w(T) \geq p + k$.

($\Leftarrow$) Suppose there exists a colorful arborescence $T' = (V_{T'}, A_{T'})$ of weight $w(T') \geq p + k$ in G. If $V_{T'}$ does not contain p vertices from level 2, then it is always possible to extend it to a set V_T such that $V_{T'} \subseteq V_T$, V_T is colorful and contains p vertices from level 2. Let T be the spanning arborescence of V_T. Note that since T is colorful, for any $1 \leq i \leq p$, either v_i or $\overline{v_i}$ is in V_T but not both. We now construct a truth assignment β that satisfies at least k clauses from ϕ: for every $i \in [p]$, if $v_i \in V_T$ (resp. $\overline{v_i} \in V_T$) then we let $x_i = \texttt{true}$ (resp. $x_i = \texttt{false}$) in β. We now claim that β satisfies at least k clauses from ϕ. Indeed, if a vertex $C_{j,h}$, $j \in [q]$ and $h \in \{1,2\}$ is in V_T, then necessarily $f(C_{j,h}) \in V_T$ and, by construction, C_j is satisfied by β. Moreover, $C_{j,1}$ and $C_{j,2}$ cannot both belong to V_T because T is colorful. Since T has weight $w(T) \geq p+k$, V_T must contain at least k vertices from level 3, which means that β satisfies at least k clauses.

To conclude the proof, recall that $k \leq q$ since no more than q clauses can be

satisfied. Note also that $2q \leq 3p$ as every variable appears at most three times in ϕ, while every clause is of size 2. This gives us $p \geq \frac{2q}{3} \geq \frac{2k}{3}$ and $p + k \geq \frac{2k}{3} + k \geq \frac{5k}{3}$. Thus, there exists an assignment β that satisfies at least k clauses of ϕ iff there exists a colorful arborescence $T = (V_T, A_T)$ of weight $w(T) \geq \frac{5k}{3}$ in G. The linearity of the reduction combined with the APX-hardness of Max-2-SAT(3) shows APX-hardness of UMCA-2, even on superstars. □

Although Theorem 4 suggests that no PTAS exists for UMCA-2 in superstars, we are able to provide in that case a constant ratio approximation algorithm, as described below.

Proposition 5. UMCA-2 *is 2-approximable in superstars.*

Proof. In the following, for any $v \in V$ (resp. any $V' \subseteq V$), let $\mathrm{col}^+(v)$ (resp. $\mathrm{col}^+(V')$) be the underlying set of $\mathrm{col}(N^+(v))$ (resp. $\mathrm{col}(N^+(V'))$). We first assume that the root r of the superstar G is also the *center* of G, i.e., that there does not exist any vertex $v \in V$ such that $d(r, v) > 2$ – the case where r is not the center will be discussed later. If G is a superstar, we can see it as a three-leveled graph where the root r belongs to the first level, $V_2 = N^+(r)$ belongs to the second level, and $V_3 = N^+(V_2)$ belongs to the third level. For any color $c \in \mathcal{C}$, let $V_c = \{v \in V : \mathrm{col}(v) = c\}$ be the set of all vertices of color c in G. Finally, recall that any arc in G is of weight 1 in a UMCA-2 instance. As a consequence, if there exist two vertices $v_2 \in V_2$ and $v_3 \in V_3$ such that v_2 and v_3 share the same color and such that v_3 belongs to a solution $T = (V_T, A_T)$ of UMCA-2 in G, then we can replace v_3 by v_2 in V_T without decreasing the weight of T. In the following, we thus assume without loss of generality that $\mathrm{col}(V_2) \cap \mathrm{col}(V_3) = \emptyset$.

For any $1 \leq |\mathrm{col}^+(r)| \leq |\mathcal{C}| - 1$, we prove by induction that there exists a colorful arborescence $T = (V_T, A_T)$ of weight $w(T) \geq \left\lceil \frac{|\mathcal{C}|}{2} \right\rceil$ in G. To begin, if $|\mathrm{col}^+(r)| = 1$, then V_2 contains either a single vertex or two vertices which share the same color, since no color can appear more than twice in a UMCA-2 instance. In the first case, the single vertex in V_2 has an outgoing arc towards all the vertices in V_3. In the second case, at least one of the vertices in V_2 has an outgoing arc towards at least $\left\lceil \frac{|\mathcal{C}|}{2} \right\rceil - 1$ vertices of distinct colors in V_3. Therefore, in both cases, there exists a trivial colorful arborescence T of weight $w(T) \geq \left\lceil \frac{|\mathcal{C}|}{2} \right\rceil$.

For any $k \in [|\mathcal{C}| - 2]$, if $|\mathrm{col}^+(r)| = k$, then we suppose by induction hypothesis that there exists a colorful arborescence $T = (V_T, A_T)$ of weight $w(T) \geq \left\lceil \frac{|\mathcal{C}|}{2} \right\rceil$ in G. Suppose now that $|\mathrm{col}^+(r)| = k+1$. Let c be an arbitrary color in $\mathrm{col}^+(r)$, from which we define $V^+ = \{r\} \cup V_c \cup N^+(V_c)$ and $V^- = \{r\} \cup (V_2 \setminus V_c) \cup (V_3 \setminus \{v \in V_3 : \mathrm{col}(v) \in \mathrm{col}^+(V_c)\})$, with $\mathcal{C}^+ = \mathrm{col}^*(V^+)$ and $\mathcal{C}^- = \mathrm{col}^*(V^-)$. By induction, notice that $G[V^+]$ (resp. $G[V^-]$) contains a UMCA-2 solution T^+ (resp. T^-) of weight $w(T^+) \geq \left\lceil \frac{|\mathcal{C}^+|}{2} \right\rceil$ (resp. $w(T^-) \geq \left\lceil \frac{|\mathcal{C}^-|}{2} \right\rceil$). We now show that $T = T^+ \cup T^-$ is a UMCA-2 solution in G. First, recall that r necessarily is the root of T^+ and T^-. Since r is the only vertex

which belongs both to V^+ and V^-, T is an arborescence in G. Second, by construction, observe that $\mathcal{C} = \mathcal{C}^+ \oplus (\mathcal{C}^- \setminus \{\text{col}(r)\})$, where operator $\oplus$ represents the disjoint union between two sets. In fact, except for the root r that belongs to both sets, all vertices from the second level are kept either in V^+ or V^- in such a way that vertices of the same color belong to the same set. Concerning the third level, the only vertices that are not considered are outneighbors of level-2 vertices from V^- whose color is already present in V^+. As a consequence, T is colorful and $w(T) = w(T^+) + w(T^-)$, which leads to $w(T) \geq \left\lceil \frac{|\mathcal{C}^+|}{2} \right\rceil + \left\lceil \frac{|\mathcal{C}^-|}{2} \right\rceil$ and thus to $w(T) \geq \left\lceil \frac{|\mathcal{C}|}{2} \right\rceil$.

Finally, if the center of G, say z, is such that $z \neq r$, then we consider two cases: (a) either r and z are neighbors, or (b) they lie at distance 2. In case (a), we have $r \in V_2$ meaning that all outneighbors of r except z belong to the third level of G and consequently, all its outneighbors except z are leaves. Let us study the graph G by ignoring r and its neighbors from level three. To do so, we define $V^r = \{r\} \cup (N^+(r) \setminus \{z\})$ and $V' = V \setminus V^r$, with $\mathcal{C}^r = \text{col}^*(V^r)$ and $\mathcal{C}' = \text{col}^*(V')$. Notice that $G[V']$ is still a superstar centered in z and, according to the proof above, it contains a UMCA-2 solution T' rooted in z of weight $w(T') \geq \left\lceil \frac{|\mathcal{C}'|}{2} \right\rceil$. Now, we define $\mathcal{C}^u = \mathcal{C}^r \setminus \mathcal{C}'$ as the set of colors that are *uniquely* present in V^r, i.e., that are present in V^r but not in V' and we build the colorful set V^u as follows: for each color $c \in C^u$, pick exactly one vertex $v \in V^r$ such that $\text{col}(v) = c$ and add it to V^u. Recall that r belongs to V^r and its color is unique, thus $r \in V^u$. Consequently, $T^u = G[V^u]$ is a colorful arborescence rooted in r and of weight $w(T^u) = |\mathcal{C}^u| - 1$. We construct the final arborescence T by connecting the two colorful ones T^u and T' with the arc (r, z) between their corresponding roots. Since $V' \cap V^u = \emptyset$ and $\mathcal{C}' \cap \mathcal{C}^u = \emptyset$, T is a colorful arborescence rooted in r and $w(T) = w(T') + w(T^u) + 1$ which leads to $w(T) \geq \left\lceil \frac{|\mathcal{C}'|}{2} \right\rceil + |\mathcal{C}^u|$. Since $\mathcal{C} = \mathcal{C}' \oplus \mathcal{C}^u$, we can conclude $w(T) \geq \left\lceil \frac{|\mathcal{C}|}{2} \right\rceil$. In case (b), $r \in V_3$, it means that there exists a unique vertex $z' \in V_2$ that lies on the path from r to z. Moreover, since G is a superstar centered in z and r is on level three, then r has a unique outneighbor that is z'. Consequently, the color of z' is unique. Therefore, we can apply case (a) by considering z' playing the role of r. Once again, since r is connected to z' and r has a unique color, r is part of the solution and this concludes the proof. □

Theorem 4 shows that MCA remains APX-hard even in trees of height 2. Hence, another option, if one wants to find tractable instances, consists in constraining the maximum degree of the input tree, which motivates the study of comb-graphs. If $d(v)$ denotes the degree of a vertex v in a graph, a comb-graph is defined as a tree for which $d(v) \leq 3$ for any $v \in V$, and where all vertices of degree 3 lie on a path, called the *spine*. Unfortunately, we show in Theorem 8 that UMCA-2 remains APX-hard (with a large inapproximability ratio) even when the input tree is a comb-graph.

We obtain this result by reduction from Maximum Independent Set (or MIS), a proof somewhat similar to proof of Proposition 1 in [23]. In the following, we first explain our reduction from MIS to UMCA-2. Then, we will prove two intermediate lemmas, namely Lemmas 6 and 7, to show how to obtain a solution of MCA (resp. MIS) from a solution of MIS (resp. MCA) in the constructed instances.

We recall that MIS takes a graph $H = (U, E)$ as input, and asks for a maximum cardinality set $U' \subseteq U$ such that no two vertices in U' are connected by an edge in H. In the following, let $U = \{u_1, u_2 \dots u_{n'}\}$ and $E = \{e_1, e_2 \dots e_{m'}\}$. For all $i \in [n']$, let $L_i = (j_1, \dots, j_d)$ such that $j_1 < \dots < j_d$, and e_{j_h} is incident to u_i for all $h \in [d]$ be the *ordered* list of indices of edges that are incident to u_i, and note that $|L_i| = d(u_i)$ is the degree of vertex u_i. For all $k \in [d(u_i)]$, we denote by $L_i(k)$ the k-th element in L_i.

We create a directed comb-graph $G = (V, A)$, with:

$$\begin{aligned} V = &\{r_i : i \in [n']\} \\ &\cup \{x_i^{L_i(k)} : i \in [n'], k \in [d(u_i)]\} \\ &\cup \{z_i^p : i \in [n'], p \in [n'^2]\} \end{aligned}$$

and

$$\begin{aligned} A = &\{(r_i, r_{i+1}) : i \in [n' - 1]\} \\ &\cup \{(r_i, x_i^{L_i(1)}) : i \in [n']\} \\ &\cup \{(x_i^{L_i(k)}, x_i^{L_i(k+1)}) : i \in [n'], k \in [d(u_i) - 1]\} \\ &\cup \{(x_i^{L_i(d(u_i))}, z_i^1) : i \in [n']\} \\ &\cup \{(z_i^p, z_i^{p+1}) : i \in [n'], p \in [n'^2 - 1]\} \end{aligned}$$

For an illustration, see Figure 2. Informally, the vertices of type r_i represent the vertices of U and constitute the spine of G, which is rooted in r_1, and that we can visualize as a horizontal path. Then, to every r_i we attach a vertical path composed first of vertices of type x_i^h, that represent the edges incident to vertex u_i in U (ordered by their index), followed by n'^2 vertices of type z_i^p.

Since G is rooted in r_1, the orientation of the arcs directly follows and G is clearly a comb-graph. Now let us describe the colors assigned to the vertices of G: each of the vertices of type r_i and z_i^j has its own color, which is thus unique. For all $h \in [m']$, we assign the same color $c(h)$ to the two vertices $x_{i_1}^h, x_{i_2}^h \in V$. Since $x_{i_1}^h$ and $x_{i_2}^h$ represent the two extremities u_{i_1} and u_{i_2} from E of the edge e_h in U of H, this means that $c(h)$, for any $h \in [m']$, appears exactly twice.

We can easily see that the coloring is partially ordered, since the vertices of type x_i^h (which are the only ones that can have repeated colors) are ordered according to the edge number they correspond to. Finally, each arc $a \in A(G)$ is assigned a weight $w(a) = 1$, which altogether ensures that G is a valid instance of UMCA-2. In the following, for a given $i \in [n']$, we will denote by Z_i the path induced by vertices of type z_i^j.

We now prove Lemmas 6 and 7.

Lemma 6. *If there exists an independent set I of size k in H, then there exists a colorful arborescence $T = (V_T, A_T)$ of weight $W \geq k \cdot n'^2$ in G.*

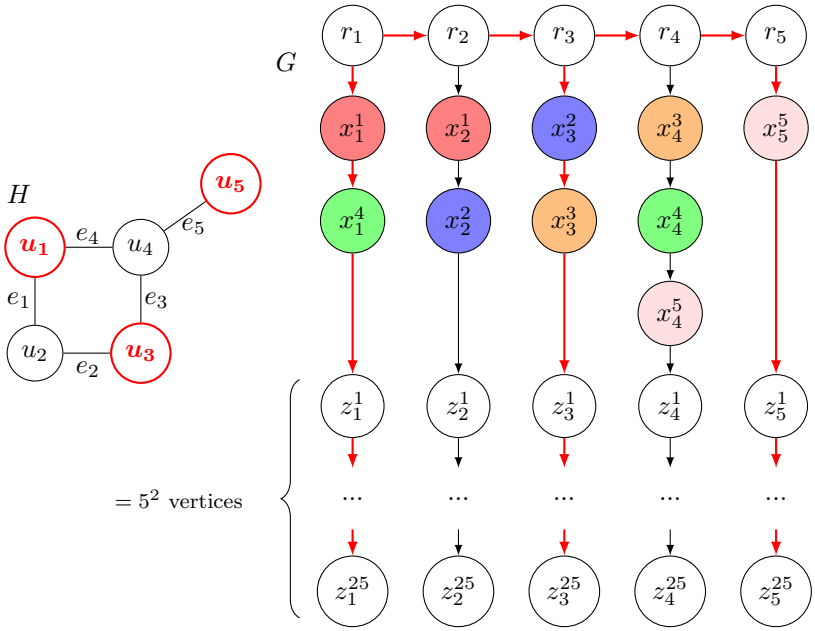

Figure 2: Construction of a UMCA-2 instance (on the right) from an MIS instance (on the left). The edges of H are called $e_1, e_2, \ldots, e_5$. Each non-colored vertex of G in the above picture has a unique color. For clarity, we do not represent the weights of the arcs – which are all equal to 1 by definition. The subset $I = \{u_1, u_3, u_5\}$ (in bold red) is an MIS solution in H and corresponds to the UMCA-2 solution T in G with bold red arcs.

PROOF. We let $V_T = \{r_i : i \in [n']\} \cup \{x_i^{L_i(j)} : i \text{ s.t. } u_i \in I, j \in [d(u_i)]\} \cup \{z_i^p : i \text{ s.t. } u_i \in I, p \in [n'^2]\}$, and we define T as the spanning arborescence of V_T. In that case T is colorful, otherwise there would exist $x_{i_1}^h, x_{i_2}^h \in V_T$, which means by construction that vertices u_{i_1} and u_{i_2} from I are connected by edge e_h, a contradiction. Moreover, T contains k paths Z_i, each of size n'^2. Consequently, $W \geq k \cdot n'^2$. □

Lemma 7. *If there exists a colorful arborescence $T = (V_T, A_T)$ of weight W in G, then there exists an independent set I of size $k \geq \left\lceil \frac{W+1-n'-m'}{n'^2} \right\rceil$ in H.*

PROOF. We build I according to the following rule: for all $i \in [n']$, if there exists $p \in [n'^2]$ such that $z_i^p \in V_T$ then $u_i \in I$. In other words, I is composed of every $i \in [n']$ for which there exists at least one vertex of type z_i^p in V_T. We first prove that I is an independent set. Indeed, suppose by contradiction there exists $u_{i_1}, u_{i_2} \in I$ such that u_{i_1} and u_{i_2} are adjacent. In that case, there would exist $h \in [m']$ such that $\text{col}(x_{i_1}^h) = \text{col}(x_{i_2}^h)$ while $x_{i_1}^h$ and $x_{i_2}^h$ belong to V_T, which contradicts the fact that T is colorful.

We now prove that $k \geq \left\lceil \frac{W+1-n'-m'}{n'^2} \right\rceil$. First, note that we can always extend T to a colorful arborescence $T' = (V_{T'}, A_{T'})$ in such a way that the following conditions hold: (i) $V_T \subseteq V_{T'}$, (ii) for every $i \in [n']$, $r_i \in V_{T'}$ and (iii) if a vertex of type z_i^p belongs

to V_T, then the whole path Z_i belongs to T'. Let W' denote the weight of T'. Since $V_{T'}$ contains n' vertices of type r_i, $k \cdot n'^2$ vertices of type z_i^j and at most m' vertices of type x_i^h, and since all arcs have weight 1, we have that $W' \leq n' + m' - 1 + k \cdot n'^2$. Now it suffices to note that $W \leq W'$ to conclude. □

We are now able to prove Theorem 8.

Theorem 8. UMCA-2 *cannot be approximated within* $\mathcal{O}(n^{\frac{1}{3}-\epsilon})$*, for any* $\epsilon > 0$*, even if* G *is a comb-graph.*

Proof. Suppose UMCA-2 is approximable in polynomial time within some ratio ρ. Then, in particular one can find in polynomial time an approximate solution of weight W for the instance we built, such that $W \geq \frac{W^*}{\rho}$, where W^* denotes the weight of an optimal solution. According to Lemma 6, from an optimal solution of MIS of size k^*, an optimal solution of weight $W' \geq k^* \cdot n'^2$ exists for UMCA-2, which implies that $W^* \geq W' \geq k^* \cdot n'^2$. By substitution, we obtain $W \geq \frac{k^* \cdot n'^2}{\rho}$. According to Lemma 7, if there exists a solution of weight W for UMCA-2, then there exists a solution of size $k \geq \frac{W+1-n'-m'}{n'^2}$ for MIS. By substitution again, we obtain $k \geq \frac{\frac{k^* \cdot n'^2}{\rho}+1-n'-m'}{n'^2}$. Note that $\frac{m'+n'-1}{n'^2} \leq 1$ as $m' \leq \frac{n'(n'-1)}{2}$. Thus, we obtain $k \geq \frac{k^*}{\rho} - 1$. As a consequence, any approximation algorithm of ratio ρ for UMCA-2 would imply an approximation algorithm of ratio ρ for MIS. We recall that MIS cannot be approximated in polynomial time within $\mathcal{O}(n'^{1-\epsilon})$ for any $\epsilon > 0$, unless $\mathsf{P} = \mathsf{NP}$ [24] and that $|V| = \mathcal{O}(n'^3)$, which concludes our theorem. □

Note that in the above argument, if we replace each vertical path of n'^2 vertices of type z_i^p by a single vertex of type z_i and if we set all weights to 0 except for the incoming arcs of all vertices of type z_i, Lemmas 6 and 7 easily show that there exists a independent set I of size k in the instance H of MIS if and only if there exists a colorful arborescence $T = (V_T, A_T)$ of weight $w(T) = k$ in the MCA$^+$-2 instance G built from H. This observation directly implies the following result.

Corollary 9. MCA$^+$-2 *cannot be approximated within* $\mathcal{O}(n^{\frac{1}{2}-\epsilon})$*,* $\epsilon > 0$*, even if* G *is a comb-graph.*

Similarly to Theorem 1, Theorem 8 and Corollary 9 provide large inapproximability ratios, i.e. ratios which are function of n, the number of vertices of the input graph G. However, we provide in Proposition 11 an approximation algorithm for UMCA in comb-graphs which, as in the proof of Proposition 2, relies on the fact that $\mathcal{H}$ is a DAG. We first need to prove the following lemma.

Lemma 10. *Let* $G = (V, A)$ *be a comb-graph rooted in* $r \in V$*, and let* p *be an arbitrary positive integer. If* $d(r, v) \leq p$ *for any vertex* $v \in V$*, then* G *contains at most* $(p+1)^2$ *vertices.*

Proof. By definition, any comb-graph $G = (V, A)$ contains a spine having vertex set $S \subseteq V$, and for any vertex $v \in S$, G may contain a *hanging* path which is attached to v, i.e. a path such that only one of the two extremities of the path belongs to S.

We first assume that we want to build a comb-graph G having a maximum number of vertices n, and such that $d(r, v) \leq p$ for any vertex $v \in V$. To begin with, we create two vertices $v_a \in S$ and $v_b \notin S$ such that r belongs to the hanging path from v_a to v_b. Let $a \leq p$ (resp. $b \leq p$) be the distance from r to v_a (resp. v_b) in G. Because of the distance constraint, we know that for any vertex $v \in S$, $d(v_a, v) \leq p - a$. As a consequence, for any $v \in S$ such that $d(v_a, v) = d'$, with $d' \in \{1, \ldots, p - a\}$, the hanging path from v is of length at most $p - a - d'$ (see Figure 3 for an illustration).

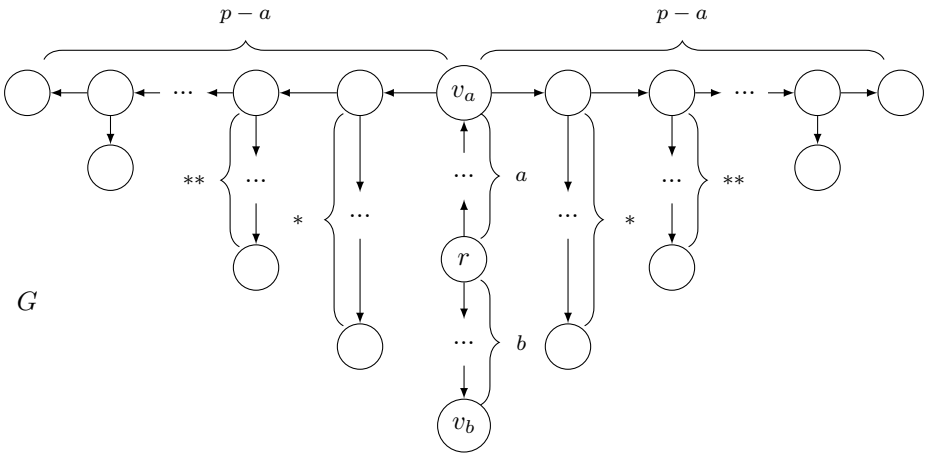

Figure 3: Construction of a comb-graph G maximizing $|V(G)|$ and such that $d(r, v) \leq p$ for any $v \in V$, where p is a given positive integer). The braces represent a maximum distance between two vertices: the symbol $*$ means "$p - a - 1$" and $**$ means "$p - a - 2$".

Now let us turn to computing the maximum number of vertices n in such a comb-graph G. Note that $n \leq 1 + a + b + 2 \cdot \sum_{d'=1}^{p-a} d'$, which leads to $n \leq 1 + a + b + 2 \cdot \frac{(p-a)(p-a+1)}{2}$ and $n \leq (p - a)^2 + p + b + 1$. Now, observe that n is maximized by setting a to 0 (which in turn implies $r \in S$), and $b = p$. As a consequence, we have that $n \leq (p + 1)^2$, which concludes the proof. □

Lemma 10 now allows us to prove the next result.

Proposition 11. UMCA *in comb-graphs can be approximated in polynomial time within a ratio* $\mathcal{O}(n^{\frac{1}{2}})$.

Proof. In the following, recall that G contains n vertices. According to Lemma 10, if $d(r, v) \leq \sqrt{n} - 2$ for any $v \in V$, then G contains at most $(\sqrt{n} - 2 + 1)^2$ vertices, which contradicts the fact that G contains n vertices. As a consequence, in any MCA instance such that G is a comb-graph, there exists at least one vertex $v \in V$ such that $d(r, v) \geq \sqrt{n} - 1$. Now, recall that any solution of UMCA in G is of weight at most $n - 1$. Moreover, recall that any path from the root in G is colorful as $\mathcal{H}$ is a DAG. Therefore, the longest path from the root is a solution to UMCA, of weight at least $\sqrt{n} - 1$. Since no solution of weight more than $n - 1$ exists, this shows UMCA admits an approximation algorithm of ratio $\mathcal{O}(n^{\frac{1}{2}})$. □

Recall that Theorem 8 does not forbid any $\mathcal{O}(n^{\frac{1}{3}})$-approximation algorithm for UMCA in comb-graphs. Therefore, determining if there exists such an approximation algorithm remains an open question. Although the approximation algorithm from Proposition 11 does not apply to comb-graphs with non-uniform weights, there exists a trivial $|\mathcal{C}|$-approximation algorithm for MCA$^+$ in such graphs. If T is the optimal solution of an instance G and W^* is the largest weight of G, then $w(T) \leq W^* \cdot (|\mathcal{C}| - 1)$ since T cannot contain more than $|\mathcal{C}|$ vertices. Moreover, recall that any path from r to a vertex $v \in V$ is colorful as $\mathcal{H}$ is a DAG. Hence, the $|\mathcal{C}|$-approximation algorithm consists in taking any path from r which includes the largest weighted arc of G.

We have seen that MCA remains hard even in superstars (Theorem 4) and comb-graphs (Theorem 8 and Corollary 9). Another way of restricting the input tree structure is to consider trees that are "close to paths". When G is a path, it can be easily seen that MCA is in P. The next step is to study caterpillars, which are trees that become paths after removal of their leaves. Note that a superstar becomes a star, i.e., a special case of caterpillar, after removal of its leaves. Moreover, MCA is APX-hard in superstars (see Theorem 4), and MCA is in P in stars (see Proposition 3). As shown below, more generally, MCA in caterpillars is in P. Thus, the following theorem allows us to draw a more precise line between intractable and tractable instances for MCA in trees.

Proposition 12. MCA *in caterpillars is in* P.

Proof. The main purpose of the algorithm we present in this proof is to show polynomiality of MCA when G is a caterpillar; no particular effort is made here on optimizing the running time. Let $S = \{r\} \cup \{v \in V : d^+(v) \neq 0\}$ contain the root r and all vertices of G which are not leaves. Clearly, $G[S]$ is connected by definition of a caterpillar. The proposed algorithm works as follows. First, we generate all colorful subsets $S' \subseteq S$ such that $r \in S'$ and $G[S']$ is connected. Second, for each such S', we denote by $N^{++}(S') = \{v \in N^+(u) : u \in S' \text{ and } v \notin S\}$, we create a set $S'' = S'$ and we proceed as follows: for all colors $c \in \text{col}^*(N^{++}(S')) \setminus \text{col}(S')$, take $x \in N^{++}(S')$ of color c with the maximum weighted incoming arc a_x and add x to S'' only if $w(a_x) > 0$. From this newly built set S'', we compute the spanning arborescence T'', and finally output the arborescence that reaches the maximum weight among all the computed arborescences.

Clearly, the algorithm is correct because we generated all possible connected and colorful subsets S'. Moreover, for any such subset S', computing a maximum spanning arborescence in a caterpillar which contains S' can be achieved greedily. Since there are $\mathcal{O}(n^2)$ subsets of vertices S' such that $r \in S'$ and $G[S']$ is connected, and since each S' is treated in polynomial time, the whole algorithm is thus polynomial. □

4. A closer look at the Color Hierarchy Graph $\mathcal{H}$

One major difference between MCS and MCA lies in the fact that $\mathcal{H}$ is imposed to be a DAG in any MCA instance. In this section, we exploit this particularity

and prove that MCA belongs to P whenever $\mathcal{H}$ is a tree. We first give the following reduction rule, and prove its correctness.

Reduction Rule 13. *Let $I = (G, \mathcal{C}, \text{col}, w, r)$ be an* MCA *instance. If there exists a color $c \in \mathcal{C}$ such that (i) c does not have any outneighbor in $\mathcal{H}$ and (ii) c has a unique inneighbor $c^- \neq \text{col}(r)$ in $\mathcal{H}$, then we do the following:*

- *for any vertex $v^- \in V(G)$ of color c^-, add $\max\{0, \max_{v \in N^+(v^-) \mid \text{col}(v)=c}\{w(v^-, v)\}\}$ to the weight of each incoming arc of v^-;*
- *remove all vertices of color c in $V(G)$, alongside to all arcs which are incident to a vertex of color c in $A(G)$.*

Lemma 14. *Reduction rule 13 is correct.*

Proof. Let $I' = (G', \mathcal{C}', \text{col}', w', r')$ be the MCA instance which is obtained after applying Reduction rule 13 on I. We show that there exists a colorful arborescence $T = (V_T, A_T)$ of weight at least W in I if and only if there exists a colorful arborescence $T' = (V_{T'}, A_{T'})$ of weight at least W in I'. We only show the first direction of the equivalence; the other can obtained by using a symmetric reasoning. First, if T does not contain any vertex of color c, then we can trivially set $T' = T$. Otherwise, if T contains a vertex v of color c and an arc (u, v), then we set $V_{T'} = V_T \setminus \{v\}$ and $A_{T'} = A_T \setminus \{(u, v)\}$. Clearly, T' is a colorful arborescence in G'. Moreover, if u^- is the inneighbor of u in T (and thus in T'), then $w(T') = w(T)$ since $w'(u^-, u) = w(u^-, u) + w(u, v)$. □

Figure 4 shows an illustration of the application of Reduction rule 13. Now, we show how to use this rule in order to prove that MCA belongs to P when $\mathcal{H}$ is a tree.

Theorem 15. MCA *belongs to* P *when $\mathcal{H}$ is a tree.*

Proof. First, recall that Reduction rule 13 removes at most $n - 1$ vertices from G and modifies in polynomial time the weight of at most $m - 1$ arcs of G. Hence, this reduction rule can be executed in polynomial time. Moreover, it can be executed at most $|\mathcal{C}| - 2$ times from an initial instance of MCA. Finally, let $I = (G, \mathcal{C}, \text{col}, w, r)$ be an initial instance of MCA such that $\mathcal{H}$ is a tree. If we apply Reduction rule 13 on I until it can no longer be applied, then G is clearly a star in the final instance which is obtained from I (see Figure 5 for an illustration). As MCA in stars is in P (see Proposition 3), we conclude that MCA is in P when $\mathcal{H}$ is a tree. □

We recall that s is defined as the number of arcs that need to be removed from $\mathcal{H} = (\mathcal{C}, A_{\mathcal{C}})$ in order for $\mathcal{H}$ to become a tree. In the following, let $\mathcal{X} = \{c \in \mathcal{C} \ : \ d^-(c) > 1\}$ be the set of *difficult* colors in $\mathcal{H}$, i.e., colors that have indegree strictly more than one; clearly, $\mathcal{H}$ is not a tree whenever $\mathcal{X}$ is not empty. For any $\mathcal{X} \neq \emptyset$, we let $p = \min\{d^-(c) : c \in \mathcal{X}\}$. We now show how to use Theorem 15 in order to obtain an FPT algorithm for MCA parameterized by $s + p$.

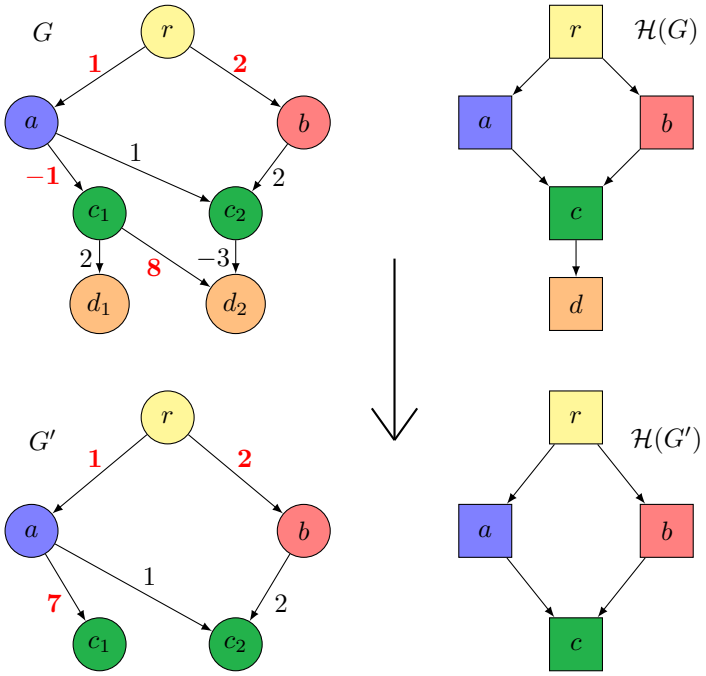

Figure 4: Example of application of Reduction rule 13 on the orange color (vertices marked d_1 and d_2) of an initial MCA instance, whose initial graph G is at the top left and whose initial Color Hierarchy Graph is at the top right. We create a new MCA instance whose graph G' is at the bottom left and whose Color Hierarchy Graph is at the bottom right. We initialize $G' = G$, then we remove all orange vertices (d_1 and d_2) in G' (together with their incident arcs). Finally, we set $w'(a, c_1) = -1 + \max\{0, \max\{2, 8\}\} = 7$, $w'(a, c_2) = 1 + \max\{0, -3\} = 1$ and $w'(b, c_2) = 2 + \max\{0, -3\} = 2$ in G'. In both graphs G and G', we represent the arcs of an MCA solution in bold red.

Proposition 16. MCA *can be solved in time* $\mathcal{O}^*(p^{\frac{s}{p-1}})$.

Proof. We design a branching algorithm that recursively computes every spanning arborescence of $\mathcal{H}$. To do so, we create a set Z, where initially $Z = A(\mathcal{H})$, and we consider a graph $\mathcal{H}' = (\mathcal{C}, Z)$. For every difficult color $c \in \mathcal{X}$, we recursively branch on the $d^-(c)$ different cases where only one incoming arc of c is not removed from Z.

At the end of these branching steps, each color $c \in \mathcal{C}$ has indegree 1 in $\mathcal{H}'$ and thus the connected component of $\mathcal{H}'$ which contains $\mathrm{col}(r)$ – where r is the root of G – is a tree. We then create a graph $G' = (V, A_Z)$ with $A_Z = A \setminus \{(x, y) \in A : (\mathrm{col}(x), \mathrm{col}(y)) \notin Z\}$ and we consider the connected component G'' of G' which contains the root of G'. Informally, G'' is a subgraph of G, which is built from Z and such that $\mathcal{H}(G'')$ is a tree. Hence, by Theorem 15, we know that computing a colorful arborescence of maximum weight in G'' is polynomial-time solvable. Now, for any solution T of MCA in G, recall that the color hierarchy graph of T is necessarily a tree. As a consequence, computing a solution of MCA in every such subgraph $G'' \subseteq G$ ensures that the above described algorithm is correct.

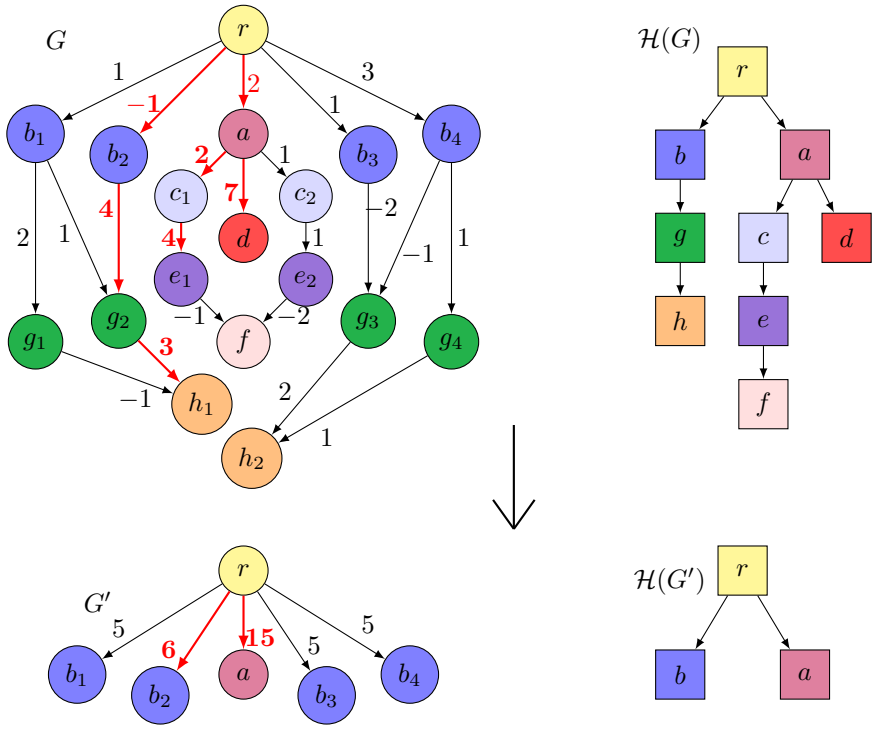

Figure 5: Example of iterative applications of Reduction Rule 13 on an initial MCA instance, whose initial graph G is at the top left and whose initial Color Hierarchy Graph is at the top right. As $\mathcal{H}(G)$ is a tree, we iteratively apply Reduction Rule 13, in this order, on the orange (h_1 and h_2), green (g_1 to g_4), pink (f), dark purple (e_1 and e_2), light blue (c_1 and c_2) and finally on the red color (d). The new MCA instance is displayed in graphs G' at the bottom left, and $\mathcal{H}(G')$ at the bottom right. In both graphs G and G', we represent the arcs of an optimal solution to MCA in bold red.

We now discuss the complexity of the above algorithm. To do so, let $\mathcal{T}$ be the search tree of the above algorithm and observe that each step that contributes to constructing $\mathcal{T}$ is achieved in polynomial time. The complexity of the above algorithm is thus only exponential in the number of nodes $|V(\mathcal{T})| = \prod_{c \in \mathcal{X}} d^-(c)$ of $\mathcal{T}$. We now show that $|V(\mathcal{T})| \leq p^{\frac{s}{p-1}}$ which in turn shows that MCA can be solved in $\mathcal{O}^*(p^{\frac{s}{p-1}})$ time. Assuming $\mathcal{X}$ is not empty, we look for the smallest real number α such that the inequality (1) $d^-(c) \leq \alpha^{d^-(c)-1}$ holds for all colors $c \in \mathcal{X}$. From (1), we have $\log(d^-(c)) \leq (d^-(c)-1) \cdot \log(\alpha)$, thus $\alpha \geq e^{\frac{\log(d^-(c))}{d^-(c)-1}}$, which gives us $\alpha \geq d^-(c)^{\frac{1}{d^-(c)-1}}$. The corresponding function $f(x) = x^{\frac{1}{x-1}}$ is monotonously decreasing for all $x \in [2; +\infty[$. By definition of p, this implies that we can set α to $p^{\frac{1}{p-1}}$ in order to ensure that $\alpha \geq d^-(c)^{\frac{1}{d^-(c)-1}}$ for any $c \in \mathcal{X}$. As $|V(\mathcal{T})| \leq \prod_{c \in \mathcal{X}} \alpha^{d^-(c)-1}$ and as $s = \sum_{c \in \mathcal{X}} d^-(c) - 1$, we obtain $|V(\mathcal{T})| \leq p^{\frac{s}{p-1}}$ and the time complexity of the above algorithm is $\mathcal{O}^*(p^{\frac{s}{p-1}})$. □

For instance, if $p = 3$, then by Proposition 16 we obtain a running time of $\mathcal{O}^*(1.733^s)$ for solving MCA. In general, we always have $p \geq 2$ for graphs G for which $\mathcal{H}$ is not a tree. Thus, by setting p to 2, it leads to an FPT algorithm of the problem parameterized by s alone and we obtain the following "universal" corollary.

Corollary 17. MCA *can be solved in time* $\mathcal{O}^*(2^s)$.

5. FPT results with respect to parameter $\ell_{\mathcal{C}}$

In [25], some statistics concerning more than 600,000 biological instances of MCS (provided by Sebastian Böcker's group from Friedrich-Schiller-Universität Jena, Germany), were provided. In particular, it can be seen that after applying the reduction rules from [26], the average value for parameter $\ell_{\mathcal{C}} = n - |\mathcal{C}|$ is below 16, whenever the weight of the studied metabolite is below 500 Daltons. Therefore, the study of parameter $\ell_{\mathcal{C}}$ is of particular interest for the MCA problem. Unfortunately, as noted in Section 2, MCA-1 parameterized by $\ell_{\mathcal{C}}$ is W[1]-hard. However, we show in the following that MCA is FPT parameterized by $\ell_{\mathcal{C}} + m^-$, where m^- denotes the number of negative arcs in the input instance. Although in practice $\ell_{\mathcal{C}} + m^-$ may appear too large – for instance, in the abovementioned experimental data, $\ell_{\mathcal{C}} + m^-$ reaches (on average) almost 221 for metabolites of weight at most 500 Daltons –, this result remains of interest as it allows us to draw a border between W[1]-hardness (for $\ell_{\mathcal{C}}$ alone) and fixed-parameter tractability (for $\ell_{\mathcal{C}} + m^-$). Moreover, arcs of negative weight represent arcs for which the degree of confidence (into the actual fragmentation it corresponds to) is low: the initial data can thus be modified so as to delete the less relevant arcs (by means of a threshold), which consequently decreases the value of m^-.

In order to show that MCA is FPT parameterized by $\ell_{\mathcal{C}} + m^-$, we first introduce the following notion: a *fully-colorful* subgraph of a graph G is a subgraph of G that contains exactly one occurrence of each color from $\mathcal{C}$. We recall the following lemma from [20].

Lemma 18 ([20]). *Given any graph G with $|\mathcal{C}|$ colors, there exist at most $2^{\ell_{\mathcal{C}}}$ fully-colorful subgraphs of G.*

We are now ready to prove the next proposition.

Proposition 19. MCA *can be solved in* $\mathcal{O}^*(2^{\ell_{\mathcal{C}}+m^-})$ *time.*

Proof. In the following, let $G' = (V', A')$ be an arbitrary fully-colorful subgraph of G. We say that a subset X of arcs of negative weights is *correct* if no vertex from V' has two or more incoming arcs from X. For any correct subset $X \subseteq A'$, we will first describe how to build a subgraph $G'_X = (V'_X, A'_X)$ of G' such that any spanning arborescence T'_X of G'_X necessarily contains all the arcs from $X \cap A'_X$. Then, we will prove the following claim: if T is a solution of MCA in G, then there exists a fully-colorful subgraph $G' = (V', A')$ of G, a correct subset of arcs $X \subseteq A'$ and a solution T'_X of MCA in G'_X such that $w(T'_X) = w(T)$.

We show how to build a subgraph $G'_X = (V'_X, A'_X)$ of G' from a fully-colorful subgraph $G' = (V', A')$ and a correct subset $X \subseteq A'$. First, we initialize $G'_X = G'$. Second, for any vertex $v \in V'$, if v has an incoming arc which belongs to X, then we remove from A'_X all the other incoming arcs of v. Third, we remove from G'_X all the vertices $v \in V'_X$ (and their incident arcs) such that there does not exist any path from r to v in G'_X. Now, notice that any spanning arborescence of G'_X necessarily contains every arc of X which belongs to G'_X, as any vertex which is incident to an arc from X is necessarily of indegree 1.

We can now prove our claim. Let $T = (V_T, A_T)$ be a solution of MCA in G and let $G' = (V', A')$ be a fully-colorful subgraph of G such that $T \subseteq G'$. As G' is fully-colorful, there does not exist any pair of vertices $u \in V_T$ and $v \in (V' \setminus V_T)$ such that $w(u, v) > 0$, otherwise T would not be a solution of MCA in G' – and thus in G. As a consequence, we set $X = \{a \in A_T \mid w(a) < 0\}$ and build G'_X as described above. Clearly, T is also an arborescence of maximum weight in G'_X. Therefore, to ensure that an optimal solution of MCA in G is found, we generate all fully-colorful subgraphs G' of G and any correct subset of arcs X in G', build all corresponding graphs G'_X, and for each find a maximum weight spanning arborescence.

Now, recall that computing such a spanning arborescence takes polynomial time [27, 28]. Besides, observe that G contains at most 2^{m^-} correct subsets of arcs and recall by Lemma 18 that G contains at most 2^{ℓ_c} fully-colorful subgraphs. Hence, altogether, our algorithm has a running time of $\mathcal{O}^*(2^{\ell_c + m^-})$. □

By setting $m^- = 0$, the above theorem implies the following corollary.

Corollary 20. MCA$^+$ *can be solved in* $\mathcal{O}^*(2^{\ell_c})$ *time.*

In the following, we will see that constraining the input instances allows us to derive several other positive results – for instance, if we constrain the input graph of an MCA instance, instead of constraining the weight function as in Corollary 20.

Proposition 21. MCA *in trees can be solved in* $\mathcal{O}^*(2^{\ell_c})$ *time.*

Proof. We design a recursive branching algorithm based on the colors of the input graph G. We first let $S = V$. If S is not colorful, we consider $u, v \in S$ such that $\mathrm{col}(u) = \mathrm{col}(v)$ and recursively branch on two cases: either $V(G_u[S])$ or $V(G_v[S])$ is removed from S. Recall that $G_u[S]$ (resp. $G_v[S]$) is the induced DAG of $G[S]$ that is rooted in u (resp. v). Clearly, for each set S we finally obtain, $G[S]$ is a colorful tree. As a consequence, $\mathcal{H}(G[S])$ is itself a tree, and by Theorem 15, we can compute a maximum weighted arborescence in $G[S]$ in polynomial time. Clearly, the above described algorithm is correct, and its running time is exponential only in the number of nodes of the search tree. Since this search tree is binary and of height $\ell_{\mathcal{C}} = n - |\mathcal{C}|$, our algorithm runs in $\mathcal{O}^*(2^{\ell_c})$. □

We now show that Proposition 21 can be improved when all arcs $a \in A$ have positive weights, based on the fact that, in that case, finding a maximum weighted colorful arborescence when repeated colors appear as leaves in the input tree is polynomial.

Proposition 22. MCA$^+$ *in trees can be solved in* $\mathcal{O}^*(1.62^{\ell_c})$ *time.*

PROOF. Recall that $f(v)$ denotes the unique inneighbor of any $v \in V \setminus \{r\}$ as G is a tree. We improve the branching algorithm discussed in proof of Proposition 21 above, by using a different branching procedure. Let $S = V$ and let us apply the following branching rule: if there exists $u, v \in S$ such that (i) $\text{col}(u) = \text{col}(v)$ and (ii) $|N^+(u)| > 0$ or $|N^+(v)| > 0$ (where $N^+(u)$ and $N^+(v)$ apply here in $G[S]$), then we branch on two cases: either $V(G_u[S])$ or $V(G_v[S])$ is removed from S. We repeat this branching procedure until it cannot longer be applied on S.

For any S which corresponds to a leaf of the search tree $\mathcal{T}$, we now show how to compute a solution of MCA$^+$ in $G[S]$. For any such S, let U_S be the set of vertices having a unique color in S. Note that because of condition (ii), two vertices $u, v \in S$ can have the same color only if they are both leaves of $G[S]$, and thus $G[U_S]$ is connected. Besides, recall that G is a tree and that for any arc $a \in A$, its weight $w(a)$ is positive. Thus, $G[U_S]$ is necessarily contained in a maximum colorful arborescence $T = (V_T, A_T)$ built from $G[S]$. We now need to compute T from S: we start by taking in T all vertices from U_S. Then, for every color $c \in \text{col}(S) \setminus \text{col}(U_S)$, we add to V_T the vertex $v \in S$ of color c such that $w((f(v), v))$ is maximum – note that $f(v)$ necessarily belongs to U_S. Finally, T is defined as the tree which is induced by V_T in $G[S]$ (see Figure 6). It can be easily seen that T is connected, colorful and of maximum weight in $G[S]$. This ensures the correctness of our algorithm as any solution of MCA is necessarily contained in the graph $G[S]$ which is produced from at least one leaf of $\mathcal{T}$. The computational complexity of our algorithm derives from the fact that, at each step, if $|N^+(u)| = 0$ (resp. $|N^+(v)| = 0$) then $|N^+(v)| > 0$ (resp. $|N^+(u)| > 0$). Therefore, the branching vector is $(1, 2)$, which leads to an $\mathcal{O}^*(1.62^{\ell_c})$ algorithm (for an introduction to the analysis of branching vectors, see e.g. [29]). □

Finally, we show that Proposition 22 can also be improved when all arcs in A have uniform weights.

Proposition 23. UMCA *in trees can be solved in* $\mathcal{O}^*(1.33^{\ell_c})$ *time.*

PROOF. In the following, for any $v \in V$, if v has a unique outneighbor u in G and if u is a leaf in G, then we say that v is a *near-leaf* in G. Moreover, for any vertex $v \in V$ such that v is neither a leaf nor a near-leaf in G, we call v a *trunk-vertex* in G.

We create a set $S = V$ on which we recursively apply the following branching rule: if there exists $u, v \in S$ such that (i) $\text{col}(u) = \text{col}(v)$, (ii) u (resp. v) is a trunk-vertex in $G[S]$ and (iii) v (resp. u) is a trunk-vertex or a near-leaf in $G[S]$, then we remove either $V(G_u[S])$ or $V(G_v[S])$ from S. If we assume without loss of generality that u is a trunk-vertex, the branching vector of this rule is $(3, 2)$. Indeed, we remove either at least three vertices while removing $V(G_u[S])$ – as u is a trunk-vertex – or at least two vertices while removing $V(G_v[S])$ – as v is a trunk-vertex or a near-leaf.

For any S which corresponds to a leaf of the search tree $\mathcal{T}$, let S_1 be the set of trunk-vertices in $G[S]$, S_2 be the set of near-leaves in $G[S]$, and S_3 be the set of leaves in $G[S]$. We may assume that any color $c \in \mathcal{C}$ appears in only one set among $\text{col}(S_1)$, $\text{col}(S_2)$ and $\text{col}(S_3)$. First, if S corresponds to a leaf of $\mathcal{T}$, then $\text{col}(S_1) \cap \text{col}(S_2) = \emptyset$, otherwise

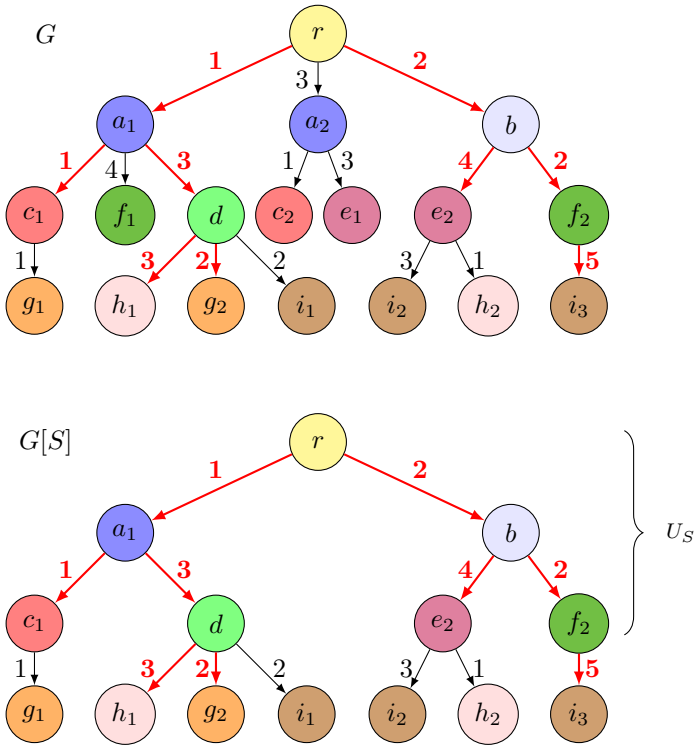

Figure 6: Example of application of the algorithm proposed in Proposition 22. Here, S corresponds to a leaf in the search tree such that we iteratively removed $V(G_{a_2}[S])$ and then $V(G_{f_1}[S])$ from $S = V$. Subset U_S contains all vertices of unique color which belong to S. As $G[U_S]$ is connected, $G[U_S]$ necessarily belongs to any solution T of MCA$^+$ in $G[S]$. For any color $c \in \text{col}(S \setminus U_S)$, it remains to add to V_T the vertex $v \in S$ of color c such that $w((f(v), v))$ is maximum in $G[S]$. In both G and $G[S]$, we represent a solution of MCA with bold red arcs.

we would have applied the above branching rule on S. Second, notice that $G[S_1]$ is colorful – as S is a leaf of $\mathcal{T}$ – and necessarily connected. As $\text{col}(S_1) \cap \text{col}(S_2) = \emptyset$, there exists at least one solution T of UMCA in $G[S]$ which contains all vertices from S_1. Indeed, if T contains a vertex $v_3 \in S_3$ such that $\text{col}(v_3) \in \text{col}(S_1)$, then we can substitute v_3 in T by a vertex of the same color which belongs to S_1. The weight of T will not be decreased as any arc weight is equal in G and as v_3 is a leaf of G. Therefore, we can remove all vertices $v_3 \in S_3$ such that $\text{col}(v_3) \in \text{col}(S_1)$ and thus assume without loss of generality that $\text{col}(S_1) \cap \text{col}(S_3) = \emptyset$. Finally, we may also assume that $\text{col}(S_2) \cap \text{col}(S_3) = \emptyset$ according to a similar reasoning.

For any S which corresponds to a leaf of $\mathcal{T}$, we now describe how to obtain a solution $T = (V_T, A_T)$ of UMCA in $G[S]$. First, recall that we may assume that any vertex from S_1 belongs to V_T. We now show how to select the vertices from S_2 and S_3 which also belong to V_T. Let M be a maximum matching in $\mathcal{H}(G[S_2 \cup S_3])$ (see Figure 7 for an illustration). For any arc $(c, c') \in M$, we add to V_T a pair of vertices

$u, v \in S$ such that $\text{col}(u) = c$, $\text{col}(v) = c'$ and $(u, v) \in A(G[S])$. Finally, for any color $c \in \text{col}(S_2)$ such that $c \notin \text{col}(V_T)$, we add an arbitrary vertex of color c from S_2 to V_T. Clearly, the spanning arborescence T of $G[V_T]$ is connected and colorful. Moreover, recall that $S_1 \in V_T$ and that $\text{col}(S_2) \subseteq \text{col}(V_T)$. As a consequence, if there exists a solution T' of UMCA such that $w(T') > w(T)$ in $G[S]$, then T' contains more vertices from S_3 than T, which contradicts the fact that M is a maximum matching in $\mathcal{H}(G[S_2 \cup S_3])$. Therefore, T is a solution of UMCA in $G[S]$. Besides, the proposed algorithm is correct as any solution of UMCA is necessarily contained in a graph $G[S]$ such that S corresponds to a leaf of $\mathcal{T}$. As the branching vector is $(3, 2)$ and as the maximum matching problem can be solved in polynomial time [28], we conclude that UMCA can be solved in $\mathcal{O}^*(1.33^{\ell_c})$ time. □

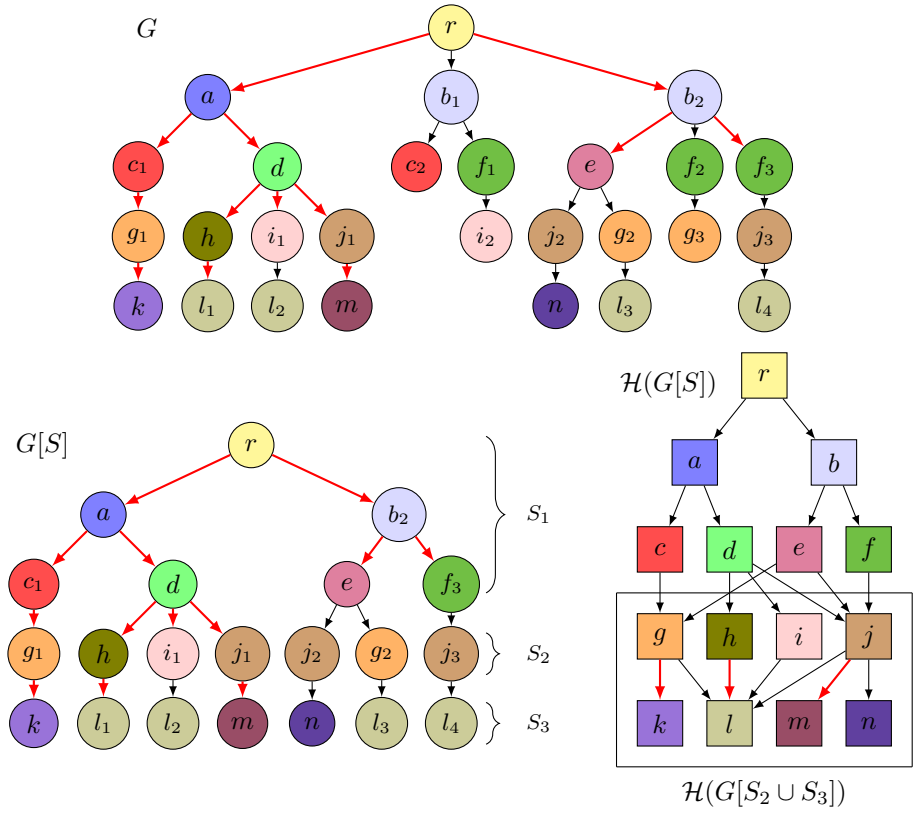

Figure 7: Example of application of the algorithm proposed in Proposition 23. Here, S corresponds to a leaf in the search tree. For clarity, we do not represent the weights of the arcs, which are all equal to 1 by definition. Observe that the spanning arborescence of $G[S_1]$ necessarily belongs to any solution of UMCA in $G[S]$ as the color of any vertex from $G[S_1]$ is unique and as $G[S_1]$ is connected. As a consequence, any solution of UMCA also contains a vertex of color c for any $c \in \text{col}(S_2)$. By solving the instance $\mathcal{H}(G[S_2 \cup S_3])$ of Maximum Matching (framed), we then compute the subset of vertices from S_3 which belongs to a solution of UMCA.

We now turn to proving a lower bound on the computational complexity of MCA with respect to $\ell_{\mathcal{C}}$. In particular, our result proves that the FPT algorithm given in Proposition 19 is essentially optimal for MCA$^+$.

Theorem 24. *The* UMCA-2 *problem cannot be solved in time* $\mathcal{O}^*((2-\epsilon)^{\ell_{\mathcal{C}}})$ *unless the Strong Exponential-Time Hypothesis fails.*

Proof. First note that the Strong Exponential-Time Hypothesis (SETH) states that the CNF-SAT problem defined on p variables cannot be solved in time $\mathcal{O}^*((2-\epsilon)^p)$ for any $\epsilon > 0$ [30]. The reduction from CNF-SAT we present here is adapted from proof of Theorem 1 in [31]. We first formally define CNF-SAT.

CNF-SAT

- **Input**: A set $X = \{x_1, x_2 \dots x_p\}$ of variables, a CNF-formula ϕ on a set $C = \{C_1, C_2 \dots C_q\}$ of clauses built from X.
- **Output**: An assignment $\beta : X \to \{\texttt{true}, \texttt{false}\}$ that satisfies ϕ.

Starting from any instance ϕ of CNF-SAT, we build an instance of UMCA-2 in the form of a three-level graph G (see Figure 8). First, level 1 only consists of the root r. For each variable $x_i \in X$, $1 \leq i \leq p$, we create two vertices v_i and $\overline{v_i}$ at level 2. For each clause $C_j \in C$, $1 \leq j \leq q$, we create a vertex z_j at level 3. We then add an arc from r to v_i and to $\overline{v_i}$ for all $i \in [p]$. There is also an arc from v_i (resp. $\overline{v_i}$) to z_j iff literal x_i (respectively $\overline{x_i}$) appears in clause C_j, for all $i \in [p]$ and for all $j \in [q]$. The root r (level 1) and every level 3 vertex is assigned a unique color. At level 2, for all $i \in [p]$, v_i and $\overline{v_i}$ share the same color c_i. Thus, every color $c \in \mathcal{C}$ can appear at most twice, and these colors can easily be partially ordered (and thus totally ordered) based on their level in the graph. Finally, the weight of every arc is 1, and it can be seen that G is indeed an instance of UMCA-2.

We now show that there exists an assignment β that satisfies ϕ iff there exists a colorful arborescence of weight $p+q$ (and thus of order $p+q+1$) in G.

($\Rightarrow$) Suppose there exists an assignment `true`/`false` of each $x_i \in X$, say β, that satisfies ϕ. Let I_T (resp. I_F) be the set of indices $i \in [p]$ such that x_i is set to `true` (resp. `false`) by β. Let $S = \{r\} \cup \{v_i \text{ for all } i \in I_T\} \cup \{\overline{v_i} \text{ for all } i \in I_F\} \cup \{z_j \text{ for all } j \in [q]\}$. Necessarily, $G[S]$ is connected: first, r is connected to every level-2 vertex; second, a vertex z_j corresponds to a clause satisfied by some x_i (resp. $\overline{x_i}$), and by definition $G[S]$ contains v_i (resp. $\overline{v_i}$), which is connected to z_j. Now, let $T = (V_T, A_T)$ be a spanning arborescence of $G[S]$. Clearly, T is colorful and of total weight $p+q$.

($\Leftarrow$) Suppose there exists a colorful arborescence $T = (V_T, A_T)$ of weight $p+q$ in G, thus of order $p+q+1$. Note that T contains at most p vertices from level 2, and thus at least q vertices from level 3. However, level 3 contains *exactly* q vertices. Thus, V_T must be composed of the root, exactly p vertices at level 2 and exacty q vertices at level 3. Since level 2 is composed of $2p$ vertices where each color appears twice, and since T is colorful, for all $i \in [p]$, either v_i or $\overline{v_i}$ is in V_T. The assignment β is thus the following: if $v_i \in V_T$ (resp. $\overline{v_i} \in V_T$) then x_i is set to `true` (resp. `false`). Since

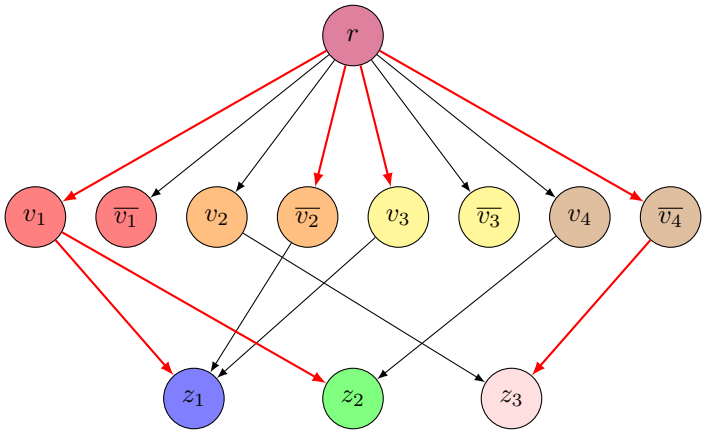

Figure 8: Construction of a UMCA-2 instance from $\phi = (x_1 \vee \overline{x_2} \vee x_3) \wedge (x_1 \vee x_4) \wedge (x_2 \vee \overline{x_4})$ of SAT. For clarity, arc weights (which are all equal to 1 by definition) are not represented. The arcs of a solution T of UMCA-2 are colored in bold red. From this solution, the assignment β consisting of setting $x_1 = \texttt{true}$, $x_2 = \texttt{false}$, $x_3 = \texttt{true}$ and $x_4 = \texttt{false}$ satisfies ϕ.

T is connected, then for any z_j with $j \in [q]$, there exists $f(z_j) \in T$, which means that every clause in ϕ is satisfied by at least one literal in β.

Hence, since $n = 2p+q+1$ and $|\mathcal{C}| = p+q+1$, we have that $\ell_{\mathcal{C}} = n - |\mathcal{C}| = p$. As a consequence, every algorithm running in time $\mathcal{O}^*((2-\epsilon)^{\ell_{\mathcal{C}}})$ for UMCA-2 would imply an algorithm running in time $\mathcal{O}^*((2-\epsilon)^p)$ for CNF-SAT, which would contradict SETH. □

6. Conclusion

In this paper, we introduced the MCA problem, a constrained version of the MCS problem, where the input graph G and its Color Hierarchy Graph $\mathcal{H}$ must be two rooted DAGs. MCA is designed to better represent the initial motivation of *de novo* inference of metabolites from tandem mass spectra, and leads to better-shaped algorithms. Although we showed that MCA remains APX-hard even for constrained inputs, we also showed that it is possible to take advantage of the fact that $\mathcal{H}$ is a DAG to describe new polynomial-time and FPT algorithms, alongside to new approximation algorithms. It remains an open problem whether other polynomial-time algorithms based on the structure of $\mathcal{H}$ can be designed.

We also provided algorithmic results concerning parameter $\ell_{\mathcal{C}} = n - |\mathcal{C}|$. Although MCA is W[1]-hard when parameterized by $\ell_{\mathcal{C}}$, we were able to provide several FPT results parameterized by $\ell_{\mathcal{C}}$ for some variants of MCA. Some of these may still be improved, even though other results taking $\ell_{\mathcal{C}}$ as a parameter were recently obtained in [20]. Many problems remain open for the MCA problem, and notably (in)approximability gaps remain to be filled. A more general question, based on the fact that, in experimental data, $\ell_{\mathcal{C}}$ remains high when the weight of a metabolite is

greater than 500 Daltons, is whether it is possible to obtain new data reduction rules which would make our FPT algorithms parameterized by $\ell_{\mathcal{C}}$ even more efficient.

Acknowledgments

We would like to thank the referees for their numerous valuable comments, which helped improve the quality of the paper.

References

[1] S. Böcker, F. Rasche, Towards *de novo* identification of metabolites by analyzing tandem mass spectra, in: 7th European Conference on Computational Biology (ECCB'08), Vol. 24(16), Bioinformatics, 2008, pp. i49–i55.

[2] A. R. Fernie, R. N. Trethewey, A. J. Krotzky, L. Willmitzer, Metabolite profiling: from diagnostics to systems biology, Nature reviews molecular cell biology 5 (9) (2004) 763.

[3] R. L. Last, A. D. Jones, Y. Shachar-Hill, Innovations: Towards the plant metabolome and beyond, Nature Reviews Molecular Cell Biology 8 (2) (2007) 167.

[4] S. Neumann, S. Böcker, Computational mass spectrometry for metabolomics: identification of metabolites and small molecules, Analytical and bioanalytical chemistry 398 (7-8) (2010) 2779–2788.

[5] J. W.-H. Li, J. C. Vederas, Drug discovery and natural products: end of an era or an endless frontier?, Science 325 (5937) (2009) 161–165.

[6] B. M. Schmidt, D. M. Ribnicky, P. E. Lipsky, I. Raskin, Revisiting the ancient concept of botanical therapeutics, Nature chemical biology 3 (7) (2007) 360.

[7] H. Oberacher, M. Pavlic, K. Libiseller, B. Schubert, M. Sulyok, R. Schuhmacher, E. Csaszar, H. C. Köfeler, On the inter-instrument and inter-laboratory transferability of a tandem mass spectral reference library: 1. Results of an Austrian multicenter study, Journal of mass spectrometry 44 (4) (2009) 485–493.

[8] R. Tautenhahn, K. Cho, W. Uritboonthai, Z. Zhu, G. J. Patti, G. Siuzdak, An accelerated workflow for untargeted metabolomics using the METLIN database, Nature biotechnology 30 (9) (2012) 826.

[9] D. S. Wishart, C. Knox, A. C. Guo, R. Eisner, N. Young, B. Gautam, D. D. Hau, N. Psychogios, E. Dong, S. Bouatra, R. Mandal, I. Sinelnikov, J. Xia, L. Jia, J. A. Cruz, E. Lim, C. A. Sobsey, S. Shrivastava, P. Huang, P. Liu, L. Fang, J. Peng, R. Fradette, D. Cheng, D. Tzur, M. Clements, A. Lewis, A. De Souza, A. Zuniga, M. Dawe, Y. Xiong, D. Clive, R. Greiner, A. Nazyrova, R. Shaykhutdinov, L. Li, H. J. Vogel, I. Forsythe, HMDB: a knowledgebase for the human metabolome, Nucleic Acids Research 37 (2009) D603–D610. doi:10.1093/nar/gkn810.

[10] F. Rasche, A. Svatos, R. K. Maddula, C. Böttcher, S. Böcker, Computing fragmentation trees from tandem mass spectrometry data, Analytical Chemistry 83 (4) (2011) 1243–1251.

[11] F. Rasche, K. Scheubert, F. Hufsky, T. Zichner, M. Kai, A. Svatos, S. Böcker, Identifying the unknowns by aligning fragmentation trees, Analytical chemistry 84 (7) (2012) 3417–3426.

[12] F. Hufsky, K. Dührkop, F. Rasche, M. Chimani, S. Böcker, Fast alignment of fragmentation trees, Bioinformatics 28 (12) (2012) 265–273.

[13] K. Dührkop, F. Hufsky, S. Böcker, Molecular formula identification using isotope pattern analysis and calculation of fragmentation trees, Mass Spectrometry 3 (Special Issue 2) (2014) S0037–S0037.

[14] S. Böcker, K. Dührkop, Fragmentation trees reloaded, J. Cheminformatics 8 (1) (2016) 5:1–5:26.

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

[16] G. Fertin, J. Fradin, G. Jean, Algorithmic aspects of the maximum colorful arborescence problem, in: T. V. Gopal, G. Jäger, S. Steila (Eds.), Theory and Applications of Models of Computation - 14th Annual Conference, TAMC 2017, Proceedings, Vol. 10185 of Lecture Notes in Computer Science, 2017, pp. 216–230. doi:10.1007/978-3-319-55911-7_16.

[17] I. Rauf, F. Rasche, F. Nicolas, S. Böcker, Finding maximum colorful subtrees in practice, Journal of Computational Biology 20 (4) (2013) 311–321. doi:10.1089/cmb.2012.0083.

[18] R. Dondi, G. Fertin, S. Vialette, Complexity issues in vertex-colored graph pattern matching, J. Discrete Algorithms 9 (1) (2011) 82–99. doi:10.1016/j.jda.2010.09.002.

[19] J. Scott, T. Ideker, R. M. Karp, R. Sharan, Efficient algorithms for detecting signaling pathways in protein interaction networks, Journal of Computational Biology 13 (2) (2006) 133–144. doi:10.1089/cmb.2006.13.133.

[20] G. Fertin, J. Fradin, C. Komusiewicz, On the Maximum Colorful Arborescence Problem and Color Hierarchy Graph Structure, in: G. Navarro, D. Sankoff, B. Zhu (Eds.), 29th Annual Symposium on Combinatorial Pattern Matching, CPM 2018, Vol. 105 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018, pp. 17:1–17:15.

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

[22] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and approximation: Combinatorial optimization problems and their approximability properties, Springer Verlag, 1999.

[23] R. Rizzi, F. Sikora, Some results on more flexible versions of graph motif, Theory of Computing Systems 56 (4) (2015) 612–629. doi:10.1007/s00224-014-9564-6.

[24] D. Zuckerman, Linear degree extractors and the inapproximability of Max Clique and Chromatic Number, Theory of Computing 3 (1) (2007) 103–128. doi:10.4086/toc.2007.v003a006.

[25] J. Fradin, Complex graphs in biology : problems, algorithms and evaluation. (graphes complexes en biologie : problèmes, algorithmes et évaluation), Ph.D. thesis, University of Nantes, France (2018).

[26] W. T. J. White, S. Beyer, K. Dührkop, M. Chimani, S. Böcker, Speedy colorful subtrees, in: D. Xu, D. Du, D. Du (Eds.), Computing and Combinatorics - 21st International Conference, COCOON, Proceedings, Vol. 9198 of Lecture Notes in Computer Science, Springer, 2015, pp. 310–322. doi:10.1007/978-3-319-21398-9.

[27] Y.-J. Chu, T.-H. Liu, On shortest arborescence of a directed graph, Scientia Sinica 14 (10) (1965) 1396–1400.

[28] J. Edmonds, Optimum branchings, Journal of Research of the National Bureau of Standards B 71 (4) (1967) 233–240.

[29] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, Springer, 2015. doi:10.1007/978-3-319-21275-3.

[30] R. Impagliazzo, R. Paturi, F. Zane, Which problems have strongly exponential complexity?, J. Comput. Syst. Sci. 63 (4) (2001) 512–530. doi:10.1006/jcss.2001.1774.

[31] G. Fertin, C. Komusiewicz, Graph Motif problems parameterized by dual, in: R. Grossi, M. Lewenstein (Eds.), 27th Annual Symposium on Combinatorial Pattern Matching, CPM 2016, Vol. 54 of LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016, pp. 7:1–7:12.