ELSEVIER

Theoretical Computer Science 244 (2000) 167–188

Theoretical Computer Science

www.elsevier.com/locate/tcs

The hardness of perfect phylogeny, feasible register assignment and other problems on thin colored graphs ☆

Hans L. Bodlaender[a,*,1], Michael R. Fellows[b,2], Michael T. Hallett[c,3], H. Todd Wareham[d,3], Tandy J. Warnow[e,4]

[a] *Utrecht University, Department of Computer Science, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands*

[b] *University of Victoria, Department of Computer Science, P.O. Box 3055, Victoria, B.C., Canada, V8W 3P6*

[c] *Department of Computer Science, ETH Zurich, Switzerland*

[d] *Department of Computer Science, McMaster University, Hamilton, ON Canada, L8S 4L7*

[e] *University of Pennsylvania, Moore School of Electrical Engineering D2, Department of Computer & Information Science, 200 South 33rd Street, Philadelphia, PA 19104-6389, USA*

Received September 1996; received in revised form June 1998

Communicated by A. Apostolico

Abstract

In this paper, we consider the complexity of a number of combinatorial problems; namely, Intervalizing Colored Graphs (DNA physical mapping), Triangulating Colored Graphs (perfect phylogeny), (Directed) (Modified) Colored Cutwidth, Feasible Register Assignment and Module Allocation for graphs of bounded pathwidth. Each of these problems has as a characteristic a uniform upper bound on the tree or path width of the graphs in "yes"-instances. For all of these problems with the exceptions of Feasible Register Assignment and Module Allocation, a vertex or edge coloring is given as part of the input. Our main results are that the parameterized variant of each of the considered problems is hard for the complexity classes $W[t]$ for all $t \in \mathbf{N}$. We also show that Intervalizing Colored Graphs,

☆ Some of the results contained in this paper were first reported in H.L. Bodlaender, M.R. Fellows, M.T. Hallett, Beyond NP-completeness for problems of bounded width: hardness for the W hierarchy, in: Proc. 26th Annu. ACM Symp. on the Theory of Computing, 1994, pp. 449–458; H.L. Bodlaender, M.R. Fellows, T.J. Warnow, Two strikes against perfect phylogeny, in: Proc. 19th Internat. Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 623, Springer, Berlin, 1992, pp. 373–383; M.R. Fellows, M.T. Hallett, H.T. Wareham, DNA physical mapping: 3 ways difficult, in: Tom Lengauer (Ed.), Proc. 1st Annu. European Symp. on Algorithms (ESA'93), Lecture Notes in Computer Science, vol. 726, Springer, Berlin, pp. 157–168.

* Corresponding author.

E-mail address: hansb@cs.uu.nl (H.L. Bodlaender).

[1] Supported by the ESPRIT Basic Research Actions of the EC under contract 7141 (project ALCOM II).

[2] Supported by the National Science and Engineering Research Council of Canada.

[3] This research was done while these authors were at the University of Victoria, Canada.

[4] Supported in part by a NYI award from the National Science Foundation, CCR-9457800.

PII: S0304-3975(98)00342-9

Triangulating Colored Graphs, and Colored Cutwidth are NP-Complete.

Keywords: Fixed parameter intractability; Triangulation of colored graphs; Graph problems; Complexity

1. Introduction

This paper focuses on a number of graph decision problems which share the characteristic that all have a uniform upper bound on their path or tree width in the following sense. Each of these problems takes as input a graph G (it may be colored or directed) and a positive integer k and asks a particular question regarding G. If, in fact, the answer is "yes" for this instance, then one can prove that there exists an upper bound $b(k)$ on the path or tree width of the graph.

This bound opens up the following possibility: using the algorithm of Bodlaender [7] we can find a decomposition of width $b(k)$ for G or determine that no such decomposition exists. In either case, the running time for this procedure is linear in the size of G but exponential only in k. By means of one of several general algorithmic design methodologies (see [1, 5, 6, 15, 19, 51]) we may then answer the original question in time linear in the size of G. Hence, for small values of k, this procedure may lead to algorithms that are practical even for very large instances. Examples where these methods have been successful include Treewidth, Pathwidth, Min Cut Linear Arrangement, Feedback Vertex Set, Feedback Arc Set and Search Number.

Unfortunately, we show several parameterized variants of the problems Intervalizing Colored Graphs, Triangulating Colored Graphs, (Modified) (Directed) Colored Cutwidth, Feasible Register Assignment, and Module Allocation on graphs of bounded pathwidth hard for $W[t]$ for all $t \in \mathbf{N}$. This excludes the possibly of applying these techniques and, in fact, goes further to exclude the possibility of an $O(|G|^{\alpha})$ algorithm (where α is independent of both the size of G and k) under the assumption (very similar to the more familiar $P \neq NP$ hypothesis) that the tth level, for any t, of the parameterized hierarchy does not collapse to the lowest level.

The reductions that we describe also demonstrate that the problems Intervalizing Colored Graphs, Triangulating Colored Graphs and Colored Cutwidth (with one color) are NP-Complete.

The plan of the paper is as follows. In Section 2, we introduce basic notions from Parameterized Complexity theory. In Section 3, definitions and some basic properties of the problems considered in this paper are given. Section 4, shows fixed parameter intractability for these problems and NP-completeness proofs are given for unparameterized variants where appropriate. Section 5 provides a short discussion of open problems.

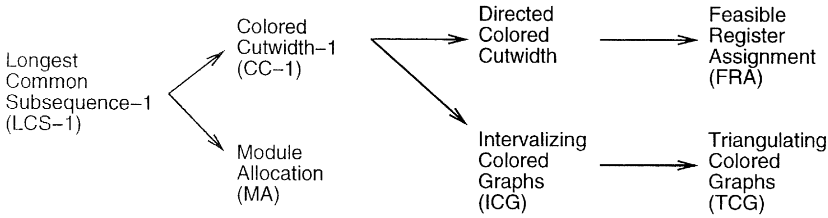

Fig. 1. The order of the parameterized reductions in this paper.

2. Parameterized computational complexity

2.1. Parameterized problems, fixed-parameter tractability and reductions

A *parameterized problem* is a set $L \subseteq \Sigma^* \times \Sigma^*$ where Σ is a fixed alphabet. For convenience, we consider that a parameterized problem L is a subset of $\Sigma^* \times N$. For a parameterized problem L and $k \in N$ we write L_k to denote the associated fixed-parameter problem $L_k = \{x \mid (x,k) \in L\}$. We say that a parameterized problem L is (uniformly) *fixed-parameter tractable* if there is a constant α and an algorithm Φ such that Φ decides if $(x,k) \in L$ in time $f(k)|x|^\alpha$ where $f : N \to N$ is an arbitrary function. Let A, B be parameterized problems. We say that A is (uniformly many: 1) *reducible* to B if there is an algorithm Φ which transforms (x,k) into $(x', g(k))$ in time $f(k)|x|^\alpha$, where $f, g : N \to N$ are arbitrary functions and α is a constant independent of k, so that $(x,k) \in A$ if and only if $(x', g(k)) \in B$. Fig. 1 shows the order of the parameterized reductions in this paper.

2.2. Complexity classes

A Boolean circuit is of *mixed type* if it consists of circuits having gates of the two kinds:

1. *Small gates*: *not* gates, *and* gates and *or* gates with bounded fan-in.
2. *Large gates*: *and* gates and *or* gates with unrestricted fan-in.

The *depth* of a circuit C is defined to be the maximum number of gates (small or large) on an input–output path in C. The *weft* of a circuit C is the maximum number of large gates on an input–output path in C. A family of decision circuits F has *bounded depth* if there is a constant h such that every circuit in the family F has depth at most h, and F has *bounded weft* if there is constant t such that every circuit in the family F has weft at most t. The *weight* of a boolean vector x is the number of 1's in the vector.

Definition 1. Let F be a family of decision circuits (possibly having many different circuits with a given number of inputs). We associate with F the parameterized problem $L_F = \{(C,k)$: C accepts an input vector of weight $k\}$. A parameterized problem L belongs to $W[t]$ if L reduces to the parameterized circuit problem $L_{F(t,h)}$ for the family $F(t,h)$ of mixed type decision circuits of weft at most t, and depth at most h, for some constant h. A parameterized problem L belongs to $W[P]$ if L reduces to the circuit problem L_F, where F is the set of all circuits (no restrictions). We designate the class of fixed-parameter tractable problems FPT.

These definitions give us the hierarchy of parameterized complexity classes

$$FPT \subseteq W[1] \subseteq W[2] \subseteq \cdots W[t] \cdots \subseteq W[P]$$

for which there are many natural hard or complete problems [23–25, 35]. For example, all of the following problems are now known to be complete for $W[1]$: Square tiling, Independent set, Clique, and Bounded post correspondence problem, k-Step derivation for context-sensitive grammars, Vapnik-Chervonenkis dimension, and the k-Step halting problem for nondeterministic turing machines [17, 21, 26]. Thus, any one of these problems is fixed-parameter tractable if and only if all of the others are. Dominating set, Weighted $\{0,1\}$ Integer Programming, and Tournament Dominating Set are shown complete for $W[2]$ in [22].

In this paper, we will use as a starting point for our reductions the following problem:

Longest Common Subsequence (LCS-1)
Instance: Alphabet Σ, strings $s^1,\ldots,s^K \in \Sigma^*$, integer $M \in \mathbf{N}$.
Parameter: K.
Question: Does there exist a string in Σ^* of length at least M, that is a subsequence of each string $s^1,\ldots,s^K$?

Theorem 1 (Bodlaender et al. [8, 10]). *For all $t \in \mathbf{N}$, LCS-1 is hard for $W[t]$.*

Other problems hard (or complete) for $W[t]$ for all t include Weighted t-Normalized Satisfiability, Bandwidth, Domino Treewidth, and Uniform Emulation on a Path (see [35]). If any one of these problems is FPT, then all problems in $W[t]$ for any t are also FPT. We will describe all problems in this paper in the same format as above. We will not describe the unparameterized variants as these can be obtained by simply ignoring the parameter field of the description.

3. Problem definitions

Common to all of the problems we consider, a uniform upper bound exists for the width of the graphs. For Intervalizing Colored Graphs, Triangulating Colored Graphs, Feasible Register Assignment and Colored Cutwidth this upper bound holds only in “yes” instances. We state the appropriate definitions relating to treewidth and pathwidth below and provide several lemmas which will be used in our hardness proofs. The following subsections provide a brief overview, relevant references and, where appropriate, the said upper bound on the width of the graphs in “yes” instances.

Definition 2. A *tree-decomposition* of a graph $G=(V,E)$ is a pair $(\{X_i \mid i \in I\}, T=(I,F))$ with $\{X_i \mid i \in I\}$ a collection of subsets of V, and $T=(I,F)$ a tree, such that
- $\bigcup_{i \in I} X_i = V$.
- For all $(v,w) \in E$, there exists an $i \in I$ with $v,w \in X_i$.
- For all $v \in V$, $\{i \in I \mid v \in X_i\}$ forms a connected subtree of T.

The *width* of a tree-decomposition $(\{X_i \mid i \in I\}, T=(I,F))$ is $\max_{i \in I} |X_i| - 1$. The treewidth of a graph is the minimum width over all possible tree-decompositions of that graph.

Definition 3. A tree-decomposition $(\{X_i \mid i \in I\}, T=(I,F))$ is a *path-decomposition*, if T is a path. The pathwidth of a graph is the minimum width over all possible path-decompositions of that graph.

Path-decompositions are also often denoted by the sequence of the successive subsets X_i: $(X_1, X_2, \ldots, X_r)$. The following well known result can easily be proved.

Lemma 2. *Let $(\{X_i \mid i \in I\}, T=(I,F))$ be a tree-decomposition of $G=(V,E)$. Let $v_0, v_1, \ldots, v_r$ be a path in G. Suppose $v_0 \in X_i$, $v_r \in X_j$, and suppose that k is on the path between i and j in T. Then $\{v_0, \ldots, v_r\} \cap X_k \neq \emptyset$.*

Lemma 3 (Bodlaender and Mohring [13]). *Let $(\{X_i \mid i \in I\}, T=(I,F))$ be a tree-decomposition of $G=(V,E)$. Let $W_1, W_2 \subseteq V$, such that for all $v \in W_1$, $w \in W_2$, $(v,w) \in E$. Either for all $v \in W_1$, there exists an $i \in I$ with $\{v\} \cup W_2 \subseteq X_i$, or for all $v \in W_2$, there exists an $i \in I$ with $\{v\} \cup W_1 \subseteq X_i$.*

3.1. Intervalizing colored graphs (or DNA physical mapping)

A graph $G=(V,E)$ with a coloring $c: V \to C$ is *properly colored*, if there is no edge between vertices with the same color. An undirected graph $G=(V,E)$ is an *interval graph*, if one can associate with each vertex $v \in V$, an interval $[L_v, R_v] \subset \mathbf{R}$, such that for all $v,w \in V$, $v \neq w$: $(v,w) \in E \Leftrightarrow [L_v, R_v] \cap [L_w, R_w] \neq \emptyset$.

The following problem introduced in [28] models in a straightforward but limited way the determination of contig assemblies in DNA physical mapping.

Intervalizing Colored Graphs (ICG)
Instance: A graph $G=(V,E)$ and a coloring $c: V \to C$.
Parameter: $|C| = k$.
Question: Does there exist a supergraph $G'=(V,E')$ of G which is properly colored by c and which is an interval graph?

In practice, the size of the coloring (the number of distinct colors) is usually a small fixed constant which is independent of the size of the graph. For instance, in the sequencing of the yeast genome, a typical working parameter value of $k=8$ is reported in [18]. Hence, the complexity of ICG when parameterized by k is of importance. More on intervalizing graphs and its application to physical mapping can be found in [32, 41].

A related problem, where G' is requested to be a proper interval graph was shown to be $W[1]$-hard by Kaplan and Shamir [40].

Lemma 4. *Let $G=(V,E)$ be a graph with a vertex coloring $c:V\to\{1,2,\ldots,k\}$, that is a subgraph of a properly colored interval graph G'. Then the pathwidth of G is at most $k-1$.*

Proof. It is easy to see that the pathwidth of an interval graph is one less than its maximum clique size, which equals its chromatic number (since interval graphs are perfect, see [31]). Hence the pathwidth of G' is at most $k-1$ implying the pathwidth of G is at most $k-1$. □

3.2. Triangulating colored graphs (or perfect phylogeny)

A *phylogeny* for the set S of species, is a rooted tree in which the leaves represent the species in S and the internal nodes of the tree represent the ancestral species.

One of the standard models uses *characters* to describe species. Here, a character is an equivalence relation on the species set, partitioning the set into the different *character states* (see [27, 42] for a more complete treatment of this subject).

The Character Compatibility problem (also known as the Perfect Phylogeny problem [34]) was shown to be polynomially equivalent to the following problem in [16, 38]:

TRIANGULATING COLORED GRAPHS (TCG)
Instance: Graph $G=(V,E)$, coloring $c:V\to C$.
Parameter: $|C|=k$.
Question: Does there exist a supergraph $G'=(V,E')$ of G which is properly colored by c and which is triangulated?

(A graph is said to be *triangulated* (or chordal) if it does not contain an induced cycle of length at least four.)

The number of colors, k, of *TCG* corresponds to the number of characters in the Perfect Phylogeny problem. Since perfect phylogenies rarely occur in practice, it is often of more interest to find the maximally-true phylogenies produced by the Perfect Phylogeny problem. However, this problem is NP-complete even for binary characters [20]. One approach to approximating such phylogenies is to look for perfect phylogenies on small subsets of characters. Hence, the complexity of the perfect phylogeny problem for fixed k is still of some importance to computational biologists.

See [2, 3, 12, 34, 36, 37, 39, 43, 44, 47, 50] for previous NP-completeness and fixed parameter algorithm results.

Lemma 5. *Let $G=(V,E)$ be a triangulated graph with a proper vertex coloring $c:V\to C$. Then G does not contain a simple cycle with only two colors used for the vertices on the cycle.*

Lemma 6. *Let $G=(V,E)$ be a graph with a vertex coloring $c: V \to \{1,2,\dots,k\}$, that is a subgraph of a properly colored, triangulated graph G'. Then the treewidth of G is at most $k-1$.*

Proof. The same proof as of that for Lemma 4 replacing "triangulated" for "interval" and "treewidth" for "pathwidth". □

3.3. Colored cutwidth

Interesting variations on a number of "classical" graph-theoretic decision problems can be defined by considering an input consisting of k distinct graphs on *the same set of vertices*, and asking whether there is a solution (described in terms of the vertex set) that simultaneously solves the problem for all of the k graphs. We may equivalently view k graphs on one vertex set V as a k-edge colored multi-graph. The following problem asks whether there is a permutation of V that simultaneously has cutwidth k for each induced monochromatic subgraph.

A linear ordering of a graph $G=(V,E)$ is a bijective function $f: V \to \{1,\dots,|V|\}$. The colored cutwidth of a linear ordering f of an edge colored graph $G=(V,E)$, with edge coloring $c: E \to C$ is

$$\max_{r \in C} \max_{1 \leqslant i \leqslant |V|} |\{(v,w) \in E \mid (c((v,w))=r) \wedge f(v) \leqslant i < f(w)\}|$$

Note that a linear ordering f has colored cutwidth 1 if and only if for every two edges (v,w) and (x,y) of the same color, the open intervals $(\min(f(v),f(w)), \max(f(v),f(w)))$ and $(\min(f(x),f(y)), \max(f(x),f(y)))$ have an empty intersection. If we have two edges of the same color for which these two open intervals intersect, then we call this a *color conflict*. The colored cutwidth of G with edge coloring c is the minimum over the colored cutwidths of all possible linear orderings of G. The following is the decision version of this problem:

Colored Cutwidth One (CC-1)
Instance: a graph $G=(V,E)$, an edge coloring $c: E \to C$.
Parameter: $|C|=k$.
Question: Does G have colored cutwidth 1?

We also consider the directed colored cutwidth problem where the input is a directed acyclic graph with a coloring of its edges. We require that if $(v,w) \in E$, then $f(v) < f(w)$, i.e. we look for a topological ordering f of G with minimum colored cutwidth. Denote this problem DirectedCC-1.

Define the modified colored cutwidth (ModifiedCC-1) of a graph as follows: the modified colored cutwidth of a linear ordering f of an edge colored graph $G=(V,E)$, with edge coloring $c: E \to C$, is

$$\max_{r \in C} \max_{1 \leqslant i \leqslant |V|} |\{(v,w) \in E \mid c((v,w))=r \wedge f(v) < i < f(w)\}|$$

The modified colored cutwidth of G with edge coloring c is the minimum over the modified colored cutwidths of all possible linear orderings of G.

It is easy to show that a yes-instance of CC-1 has pathwidth at most $k-1$.

We remark that the method of Gurari and Sudborough from [33] can be generalized to solve colored cutwidth (or its directed variant) with a fixed number k of colors, and a fixed cutwidth r per color, in time $O(|V|^{kr})$.

3.4. Feasible register assignment

One of the most fundamental problems encountered in computer system design is to efficiently allocate registers during execution of a program. Consider the following restricted system consisting of a single processor and an arbitrarily high number of general purpose registers. Programs consist of a sequence of assignment instructions which take one of two possible forms: (1) load a register with the contents from a specified memory location and (2) apply an operator to the contents of two registers placing the result in a third register. See [4].

The order of execution of a program is represented by G, a directed acyclic graph. We may view the act of placing a value into a register as placing a "pebble" on a vertex of the graph. Pebbles are originally placed on vertices of in-degree 0 and moved according to the arcs of the graph. At any point during execution there are at most k pebbles on the graph.

Feasible Register Assignment (FRA)

Instance: Directed acyclic graph $G=(V,E)$, positive integer k, and a register assignment $r: V \to \{R_1,\ldots,R_k\}$.

Parameter: k.

Question: Is there a linear ordering f of G and a sequence $S_0,S_1,\ldots,S_{|V|}$ of subsets of V such that $S_0=\emptyset$, $S_{|V|}$ contains all vertices of in-degree 0 in G, and for all i, $1\leqslant i\leqslant |V|$, $f^{-1}(i)\in S_i$, $S_i-\{f^{-1}(i)\}\subseteq S_{i-1}$, S_{i-1} contains all vertices u for which $(f^{-1}(i),u)\in E$ and for all j, $1\leqslant j\leqslant k$, there is at most one vertex $u\in S_i$ with $r(u)=R_j$?

The Feasible Register Assignment problem has been well studied and it is known that the decision version which asks whether there exists a feasible register assignment with k registers is NP-Hard (see [45]). Several restricted versions of this problem have been considered and linear time algorithms have been found if, for example, the programs compute solutions to expressions which have no common subexpressions (see [46]).

In our case, we consider a parameterized variant of Feasible Register Assignment where the maximum number of registers allowed during the execution of a program is small relative to the size of the program (i.e. the number of registers k is independent of the size of the graph G).

Denote by G^R the directed graph obtained from the directed graph G by reversing the direction of all arcs.

Lemma 7. *Let f be a linear ordering of directed acyclic graph $G=(V,E)$. Let $r: V \to \{R_1,\dots,R_k\}$ be given. Write $n=|V|$. Then there exists a sequence of subsets $S_0, S_1, \dots, S_n \subseteq V$ such that this sequence and f satisfy together the conditions of the* FRA *problem if and only if*

1. *f is a topological order of G^R.*
2. *For the sequence of subsets $S'_0, S'_1, \dots, S'_n$, defined by $S'_0=\emptyset$ and for all $i, 1\leqslant i\leqslant n$, $S'_i=\{v \mid f(v)\leqslant i$ and the indegree of v in G is $0\} \cup \{v \mid f(v)\leqslant i \wedge \exists w\in V\colon (w,v)\in E \wedge f(w)>i\}$, it holds that no set contains vertices assigned to the same register, i.e., for all i, $1\leqslant i\leqslant n$, for all j, $1\leqslant j\leqslant k$, there is at most one vertex $u\in S'_i$ with $r(u)=R_j$.*

Proof. (If) One can directly verify that f and the sequence $S'_0,\dots,S'_n$ fulfil the requirements of the FRA problem.

(Only if) First note that it must be the case that for all $v\in V, f(v)=\min\{i \mid 1\leqslant i\leqslant n, v\in S_i\}$. For every edge $(u,v)\in E$, note that $v\in S_{f(i)-1}$, hence $f(v)\leqslant f(i)-1$. So f is a topological order of G^R. Next observe that we can remove a vertex w that has indegree at least 1 simultaneously from all sets S_i with $i\geqslant \max_{w\mid(w,v)\in E} f(w)$, without violating the conditions of the Feasible Register Assignment problem. □

It is not hard to show that a yes-instance for FRA has pathwidth at most k.

3.5. Module allocation on graphs of bounded pathwidth

The Module Allocation problem seeks to minimize the overall cost of executing a set of modules on a set of processors in a distributed system. The cost of executing a module is a function of (1) which processor it is executed on, (2) interference with other modules (ie. two modules require the same processor), and (3) the need to communicate with other modules.

We assume tables are given describing (1) and (2) above. The information for (3) is encoded as a graph and supplied as part of the input. In our case, we seek to minimize overall cost when this graph has a bound on its pathwidth independent of its size. More formally,

Module allocation on graphs of bounded pathwidth (MA)

Instance: A set of modules $V=\{V_1,V_2,\dots,V_m\}$,
a set of processors $P=\{P_1,P_2,\dots,P_p\}$,
a cost function $e:(V\times P)\to \mathbf{R}:(x,y)\mapsto t$ where t is the cost of executing module $x\in V$ on processor $y\in P$,
a communication cost function $C:(V\times P\times V\times P)\to\mathbf{R}:(x,y,x',y')\mapsto t$ where t is the communication cost when module x is assigned to processor y and module x' is assigned to processor y',
a communication graph $G=(V,E)$,
and a positive real number l.
Parameter: *pathwidth*$(G)=k$.

Question: Does there exist an assignment of modules to processors such that the total cost of execution is less than or equal to l?

For $C(x, y, x', y) = \varepsilon$ then we interpret ε as the amount of interference caused by assigning both x and x' to execute on processor y.

The MA problem is known to be NP-Hard in general (see [14, 29, 48, 49]) but polynomial for several restricted families of graphs. When G is restricted to be a series–parallel graph, the best known algorithm is $O(mp^3)$ (see [14]) and when G is a tree, MA can be solved in time $O(mp^2)$ (see [49]). Fernàndez-Baca [29] generalizes this result to graphs of bounded treewidth k giving an $O(mp^{k+1})$ assignment algorithm. Furthermore, Fernàndez-Baca and Medepalli [30] consider a restricted version of MA (Parametric MA) where the cost functions e and C are linear functions of a new parameter τ; that is, $C(\cdot, \cdot) = \tau a + b$, and give an $O(m^{1+(k+1)\log_2 p} m)$ assignment algorithm. Of course, the question remains whether there exist algorithms for MA and Parametric MA with running times $O(|V|^{\alpha})$ where α is independent of the input parameters and the pathwidth of G (equal to k in this discussion). The later sections of this paper address this directly.

4. Hardness for the W-hierarchy

4.1. Hardness of CC-1, ICG, and TCG

In this section, we show that CC-1, ICG, and TCG are $W[t]$-hard for all $t \in \mathbf{N}$. First, hardness for CC-1 is shown with a reduction from LCS-1. The basic idea of this reduction is the following. We have two anchor components that are meant to be mapped to the beginning and the end of the ordering. 'Between' the anchors, we add 'choice' components: this is a sequence of vertices with between them a number of parallel edges (see Fig. 2(b).) By the edges of color c_0, the order of these vertices in the linear ordering is fixed (except that the entire ordering may be reversed). To this, we add a string component as shown in Fig. 2(c) for each string in the instance of ICG-1. (The diagram shows the part of one character of the string.) Fig. 2(a) shows how all parts are put together. The edges with color c_k fix the ordering of the vertices in the kth string component. Now, the edges with color d_k in the string components cannot be overlap areas where choice components are mapped. The edges with colors $e_{i,k}$ force a precise way how a character part is interleaved with a choice component: only one vertex of a specific character part can be in the linear ordering between two successive vertices of the choice component. This forces that exactly one character part of each string component interleaves with a choice component. The edges with colors $f_{k',k}$ force that all character parts interleaving a specific choice component correspond to the same character of Σ. A detailed proof follows below.

Hardness of ICG is shown by an (easy) reduction from CC-1. It then is shown that composing the transformations LCS-1 $\rightarrow$ CC-1 and CC-1 $\rightarrow$ ICG actually also gives a reduction from LCS-1 to TCG.

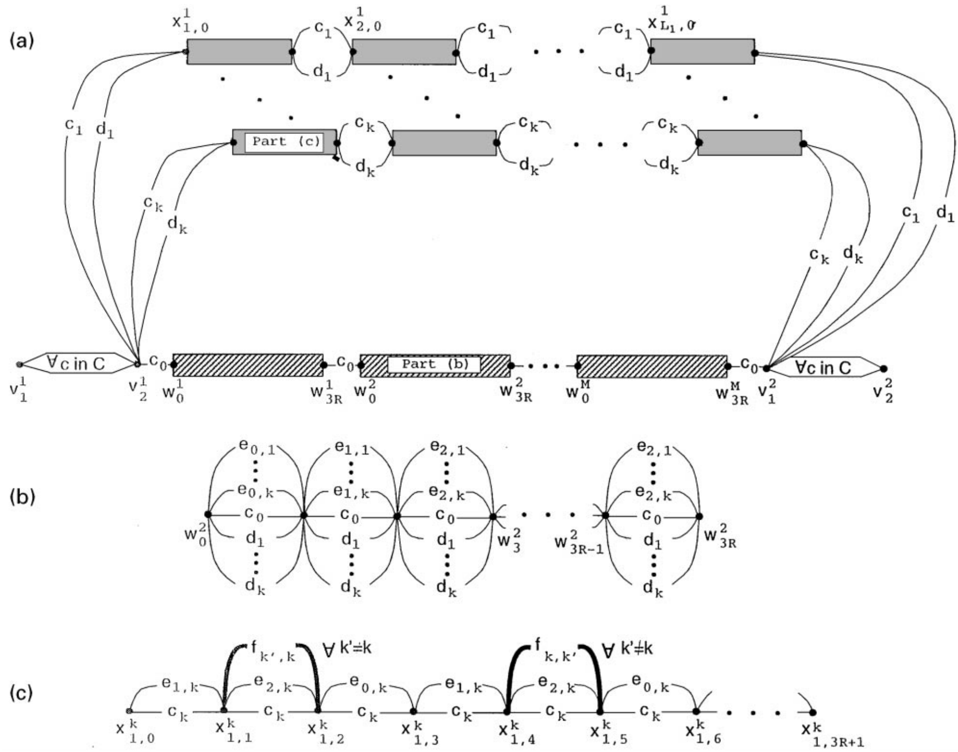

Fig. 2. The construction for the reduction from LCS-1 to Colored Cutwidth-1. (a) The general design of the reduction. (b) A choice component. (c) A character part of a string component. Note that the first symbol in the kth string is the second element of Σ ($s_1^k = \sigma_1$).

Theorem 8. (i) *CC-1 is $W[t]$-Hard for all $t \in \mathbf{N}$.*
(ii) *ICG is $W[t]$-Hard for all $t \in \mathbf{N}$.*
(iii) *TCG is $W[t]$-Hard for all $t \in \mathbf{N}$.*

Proof. (i) We reduce from LCS-1 (see definition in Section 2.2).

Let strings $s^1, \dots, s^K \in \Sigma^*$ and an integer M be an instance of LCS-1. We denote the length of a string s^k as l_k. We write $R = |\Sigma|$, and $\Sigma = \{\sigma_0, \dots, \sigma_{R-1}\}$. We now construct an edge colored graph $G = (V, E)$. We allow that G has parallel edges. (To remove the parallel edges without changing the colored cutwidth of G, we can subdivide every edge and give a subdivided edge the color of the corresponding original edge. The hardness of CC-1 for simple graphs follows from hardness of CC-1 for graphs with parallel edges.)

The set of colors C is defined as follows:

$$C = \{c_i \mid 0 \leqslant i \leqslant K\} \cup$$

$$\{d_i \mid 1 \leqslant i \leqslant K\} \cup$$

$\{e_{i,j} \mid i \in \{0,1,2\}, 1 \leqslant j \leqslant K\} \cup$

$\{f_{i,j} \mid 1 \leqslant i \leqslant K, 1 \leqslant j \leqslant K, i \neq j\}$.

We now describe G and the coloring of its edges. G consists of the following components:

1. *Two anchors*. We create four vertices $v_1^1, v_2^1, v_1^2, v_2^2$. For every color $c \in C$, we create an edge (v_1^1, v_2^1) with color c and an edge (v_1^2, v_2^2) with color c. Write $A = \{v_1^1, v_2^1, v_1^2, v_2^2\}$.

2. *Choice components*. Create vertices $\{w_i^m \mid 1 \leqslant m \leqslant M, 0 \leqslant i \leqslant 3R\}$. Create the following edges:

- An edge (v_2^1, w_0^1) with color c_0.
- An edge (w_{3R}^M, v_1^2) with color c_0.
- For all m, $1 \leqslant m < M$, an edge (w_{3R}^m, w_0^{m+1}) with color c_0.
- For all m, $1 \leqslant m \leqslant M, i, 0 \leqslant i \leqslant 3R-1$, edges (w_i^m, w_{i+1}^m) with color c_0, and for all k, $1 \leqslant k \leqslant K$, an edge (w_i^m, w_{i+1}^m) with color d_k.
- For all $m, 1 \leqslant m \leqslant M, i, 0 \leqslant i \leqslant R-1$, and for all $k, 1 \leqslant k \leqslant K$, an edge (w_{3i}^m, w_{3i+1}^m) with color $e_{0,k}$, an edge (w_{3i+1}^m, w_{3i+2}^m) with color $e_{1,k}$, and an edge (w_{3i+2}^m, w_{3i+3}^m) with color $e_{2,k}$.

3. *String components*. Create vertices $\{x_{l,i}^k \mid 1 \leqslant k \leqslant K,\ 1 \leqslant l \leqslant l_k,\ 0 \leqslant i \leqslant 3R+1\}$. For all $k, 1 \leqslant k \leqslant K$, create the following edges:

- Two edges $(v_2^1, x_{1,0}^k)$, one with color c_k, and one with color d_k.
- Two edges $(x_{l_k,3R+1}^k, v_1^2)$, one with color c_k and one with color d_k.

For all $k, 1 \leqslant k \leqslant K$, and all l, $1 \leqslant l < L_k$, create the following edges:

- Two edges $(x_{l,3R+1}^k, x_{l+1,0}^k)$, one with color c_k and one with color d_k.
- For all i, $0 \leqslant i \leqslant 3R$, an edge $(x_{l,i}^k, x_{l,i+1}^k)$ with color c_k.
- For all r, $0 \leqslant r \leqslant R-1$, an edge $(x_{l,3r}^k, x_{l,3r+1}^k)$ with color $e_{1,k}$, an edge $(x_{l,3r+1}^k, x_{l,3r+2}^k)$ with color $e_{2,k}$, and an edge $(x_{l,3r+2}^k, x_{l,3r+3}^k)$ with color $e_{0,k}$.
- An edge $(x_{l,3R}^k, x_{l,3R+1}^k)$ with color $e_{1,k}$.
- Suppose σ_i is the l'th character of string s^k, i.e., $s_l^k = \sigma_i$. Then, for all $r \neq i$, and for all $k' \neq k$, create an edge $(x_{l,3r+1}^k, x_{l,3r+2}^k)$ with color $f_{k',k}$. For all $k' \neq k$, create an edge $(x_{l,3i+1}^k, x_{l,3i+2}^k)$ with color $f_{k,k'}$.

Let $G = (V, E)$ be the resulting graph, and let $c_G : E \to C$ be the resulting coloring of the edges of G. See Fig. 2 for an illustration of this construction.

Claim 8.1. *G with coloring c_G has colored cutwidth 1 if and only if $s^1, \ldots, s^K$ have a common subsequence of length M.*

Proof. (Only if) Suppose f is a linear ordering of G with colored cutwidth 1. Note that no vertex u can be placed by f between v_1^1 and v_2^1, as any edge adjacent to u would cause a color conflict with one of the edges between v_1^1 and v_2^1. Furthermore, for no edge $(s,t) \in E$, can it be the case that s is placed to the left of v_1^1 and t is placed right of v_1^1, as this also causes a color conflict. A similar argument is valid for v_2^1, and for the other 'anchor' vertices v_1^2 and v_2^2. It follows that all vertices not in A

must be placed between $f(v_2^1)$ and $f(v_1^2)$. So, w.l.o.g., we may assume that for all $x \notin A$, $f(v_1^1) < f(v_2^1) < f(u) < f(v_1^2) < f(v_2^2)$.

Note that every vertex $x \in V - A$ lies on a path from v_2^1 to v_1^2 with all edges of this path of the same color $c \in \{c_0, c_1, \ldots, c_K\}$. If $v_2^1, y_1, y_2, \ldots, y_p, v_1^2$ is such a path, we must have that $f(v_2^1) < f(y_1) < f(y_2) < \cdots < f(y_p) < f(v_1^2)$, otherwise we have a color conflict. It follows that we have for all m, $1 \leqslant m \leqslant M$, i, i', $0 \leqslant i < i' \leqslant 3R$, $f(w_i^m) < f(w_{i'}^m)$, and that for all m, m', $1 \leqslant m < m' \leqslant M$, i, i', $0 \leqslant i, i' \leqslant 3R$, $f(w_i^m) < f(w_{i'}^{m'})$. Also, for all k, $1 \leqslant k \leqslant K$, l, $1 \leqslant l \leqslant l_k$, i, i', $0 \leqslant i < i' \leqslant 3R+1$, $f(x_{l,i}^k) < f(x_{l,i'}^k)$, and for all k, $1 \leqslant k \leqslant K$, l, l', $1 \leqslant l < l' \leqslant l_k$, i, i', $0 \leqslant i, i' \leqslant 3R+1$, $f(x_{l,i}^k) < f(x_{l',i'}^k)$.

Now, look at vertices of the form $x_{l,0}^k$ and $x_{l,3R+1}^k$. As these are adjacent to an edge with color d_k, they cannot be placed between two vertices of the form w_i^m, w_{i+1}^m, so they must be placed in one of the following open intervals:

- $(f(v_2^1), f(w_0^1))$
- $(f(w_{3R}^m), f(w_0^{m+1}))$ for some m, $1 \leqslant m \leqslant M$.
- $(f(w_{3R}^M), f(v_1^2))$.

Moreover, all vertices $x_{1,0}^k$ must be placed in the first of these intervals, and all vertices $x_{l_k,3R+1}^k$ must be placed in the last of these intervals. Also, for all l, $1 \leqslant l < l_k$, the two vertices $x_{l,3R+1}^k$ and $x_{l+1,0}^k$ must belong to the same interval.

Write, for all k, $1 \leqslant k \leqslant K$, and all m, $1 \leqslant m \leqslant M$,

$$g(k,m) = \max\{l \mid 1 \leqslant l \leqslant M,\ f(x_{l,0}^k) < f(w_0^m)\}.$$

Consider a fixed k, $1 \leqslant k \leqslant K$. As $f(x_{1,0}^k) < f(w_0^1)$, we have that all $g(k,m) \geqslant 1$. Note that for each i, $1 \leqslant i \leqslant 3R-2$, and for each m, $1 \leqslant m \leqslant M$, there must be at least one vertex of the form $x_{l,j}^k$ with $f(w_i^m) < f(x_{l,j}^k) < f(w_{i+1}^m)$. If not, then there is an edge (between two vertices in the kth string component) with color d_k, $e_{0,k}$, $e_{1,k}$, or $e_{2,k}$, that crosses both w_i^m and w_{i+1}^m. But this gives a color conflict at either w_i^m or w_{i+1}^m (or both) since at least one of these two vertices is incident to an edge of one of these four colors. Also, for any such vertex $x_{l,j}^k$, we have that $j \neq 0$ and $j \neq 3R+1$. So, we now have that $g(k,1) < g(k,2) < \cdots < g(k,M)$.

Consider some fixed k, $1 \leqslant k \leqslant K$, and m, $1 \leqslant m \leqslant M$. For each of the pairs w_{3i+1}^m, w_{3i+2}^m, $(0 \leqslant i \leqslant R-1)$ there must be at least one vertex $x_{g(k,m),j}^k$ with $f(w_{3i+1}^m) < f(x_{g(k,m),j}^k) < f(w_{3i+2}^m)$. Since between w_{3i+1}^m and w_{3i+2}^m there is an edge with color $e_{1,k}$, $x_{g(k,m),j}^k$ may not be adjacent to an edge with color $e_{1,k}$, so j must be of the form $j = 3j' + 2$. As we have R intervals $(f(w_{3i+1}^m), f(w_{3i+2}^m))$, and R vertices of the form $x_{g(k,m),3j'+2}^k$, it follows that for all i, $0 \leqslant i \leqslant R-1$, $f(w_{3i+1}^m) < f(x_{g(k,m),3i+2}^k) < f(w_{3i+2}^m)$. With a similar argument it follows that $f(w_{3i+2}^m) < f(x_{g(k,m),3i+3}^k)$.

So, now for all k, k', $1 \leqslant k, k' \leqslant K$, m, $1 \leqslant m \leqslant M$, we have that the open intervals $(f(x_{g(k,m),3i+1}^k), f(x_{g(k,m),3i+2}^k))$ and $(f(x_{g(k',m),3i+1}^{k'}), f(x_{g(k',m),3i+2}^{k'}))$ overlap. Suppose that $s_{g(k,m)}^k = \theta_i \neq s_{g(k',m)}^{k'} = \theta_{i'}$. Now, edges $(x_{g(k,m),3i+1}^k, x_{g(m),3i+2}^k)$ and $(x_{g(k',m),3i+1}^{k'}, x_{g(k',m),3i+2}^{k'})$ exist with color $f_{k,k'}$. This gives a color conflict, and the contradiction follows. It follows that, for all k, character sequences $s_{g(k,1)}^k s_{g(k,2)}^k \cdots s_{g(k,m)}^k$ are subsequences of s^k of length m, and that all these sequences are equivalent.

(If) Now suppose $s^1,\dots,s^K$ have a common subsequence of length M. Let $g:\{1,\dots,K\}\times\{1,\dots,M\}\to\mathbf{N}$ be a function, such that for all k, $1\leqslant k\leqslant K$, m,m', $1\leqslant m<m'\leqslant M$: $1\leqslant g(k,m)<g(k,m')\leqslant l_k$, and that all subsequences $s^k_{g(k,1)}s^k_{g(k,2)}\cdots s^k_{g(k,m)}$ are equivalent. We denote, for all k, $1\leqslant k\leqslant K$, $g(k,0)=0$. The following procedure produces a linear ordering f of G with colored cutwidth 1.

$f(v^1_1):=1;$
$f(v^1_2):=2;$
$p:=3;$
for $m:=1$ **to** M **do** (*Number the vertices of the form* $x^k_{l,i}$ *for* $g(k,m-1)<l\leqslant g(k,m)$, *and the vertices of the form* w^m_i).
  (*First, number the vertices of the form* $x^k_{l,i}$ *for* $g(k,m-1)<l<g(k,m)$)
  for $k:=1$ **to** K
  do for $l:=g(k,m-1)+1$ **to** $g(k,m)-1$
    do (*If* $g(k,m)=g(k,m-1)+1$, *then nothing happens in this step.*)
      for $i:=0$ **to** $3R+1$
      do $f(x^k_{l,i}):=p;\ p:=p+1;$
      enddo;
    enddo;
  enddo;
  (*Number the vertices of the form* $x^k_{g(k,m),i}$ or w^m_i.)
  for $i:=0$ **to** $3R$
  do for $k:=1$ **to** K
    do $f(x^k_{g(k,m),i}):=p;\ p:=p+1;$
    enddo;
    $f(w^m_i):=p;\ p:=p+1;$
  enddo;
  for $k:=1$ **to** K
  do $f(x^k_{g(k,m),3R+1}):=p;\ p:=p+1;$
  enddo;
enddo;
(*Number the vertices of the form* $x^k_{l,i}$ *for* $l>g(k,M)$.)
for $k:=1$ **to** K **do for** $l:=g(k,M)+1$ to l_k
  do for $i:=0$ **to** $3R+1$
    do $f(x^k_{l,i}):=p;\ p:=p+1;$
    enddo;
  enddo;
enddo;
$f(v^2_1):=p;\ p:=p+1;$
$f(v^2_2):=p;$

It is an easy, but tedious verification that the function f, yielded by this procedure, indeed is a linear ordering of G with colored cutwidth 1. We only will discuss one case here, and omit the other cases.

Suppose there is a color conflict between a pair of edges $(x^k_{l,3r+1}, x_{l,3r+2})^k$ and $(x^{k'}_{l',3r'+1}, x^{k'}_{l',3r'+2})$ with color $f_{k,k'}$. By construction of the function f, l must be of the form $l = g(k,m)$ for some m, $1 \leqslant m \leqslant M$, and $l' = g(k',m)$. Also, it must be the case that $r = r'$. Existence of the edge $(x^k_{g(k,m),3r+1}, x^k_{g(k,m),3r+2})$ with color $f_{k,k'}$ shows that $s^k_{g(k,m)} = \sigma_r$. Existence of the edge $(x^{k'}_{g(k',m),3r+1}, x^{k'}_{g(k',m),3r+2})$ with color $f_{k,k'}$ shows that $s^{k'}_{g(k',m)} \neq \sigma_r$. This is a contradiction with the assumption that we have chosen equal subsequences. This ends the proof of Claim 8.1. □

From Claim 8.1 and the $W[t]$-Hardness, for all $t \in \mathbf{N}$, of the LCS-1 problem, part (i) of the theorem follows.

(i)$\Leftrightarrow$(ii) Let $G = (V, E)$ be a graph, with an edge coloring $c_G : E \to C$. We define a bipartite graph $H = (V \cup E, F)$ with $F = \{(v,(v,w)) \mid v \in V,\ (v,w) \in E\}$. ($H$ is obtained from G by subdividing every edge.) Furthermore, using a new color $a \notin C$, we define a vertex coloring $c_H : V \cup E \to C \cup \{a\}$ of H as follows: for all $v \in V$, color v with a ($c_H(v) = a$) and for all 'edge-vertices' $e \in E$, color e with its old color in G ($c_H(e) = c_G(e)$).

The following claim shows that this transformation from (G, c_G) to (H, c_H) is in fact a reduction from $CC - 1$ to ICG, hence proving part (ii) of the theorem.

Claim 8.2. *Let G and H be constructed as above. G has colored cutwidth 1 if and only if H is a subgraph of a properly colored interval graph.*

Proof. (Only if) Let $f : V \to \{1, \ldots, |V|\}$ be a linear ordering of G with colored cutwidth 1. Assign to each $v \in V$ the interval $[f(v) - \frac{1}{3}, f(v) + \frac{1}{3}]$. To every edge $(v,w) \in E$, assign the interval $[\min(f(v), f(w)) + \frac{1}{6}, \max(f(v), f(w)) - \frac{1}{6}]$. One can easily verify that these intervals form an interval model of a properly colored interval graph that contains H as a subgraph; that is, intervals of adjacent vertices intersect and no two intervals of vertices with the same color intersect. The latter condition follows from the condition that the colored cutwidth of f is 1.

(If) Suppose that we have for every vertex $z \in V \cup E$ an interval $I_z = [L_z, R_z]$ such that intervals of adjacent vertices intersect and intervals of vertices with the same color do not intersect. As all vertices $v \in V$ have the same color, all intervals I_v are disjoint. Number the vertices $v \in V$ in the following manner: take a bijective function $f : V \to \{1, \ldots, |V|\}$ such that, for all $v, w \in V$, $L_v < L_w \Leftrightarrow f(v) < f(w)$. Now $f(v) < f(w) \Rightarrow L_v \leqslant R_v < L_w \leqslant R_w$. We claim that f is a linear ordering of G with colored cutwidth 1. Consider edges $(v,w) \in E$ and $(x,y) \in E$, $f(v) < f(w)$, $f(x) < f(y)$. Note that $[R_v, L_w] \subseteq I_{(v,w)}$ and $[R_x, L_y] \subseteq I_{(x,y)}$. So, $[R_v, L_w] \cap [R_x, L_y] = \emptyset$. When analyzing the different cases with respect to the order and possible equalness of $f(v), f(w)$, $f(x), f(y)$, one easily can verify that no color conflict between (v,w) and (x,y) is possible. □

(i) $\Leftrightarrow$ (ii) $\Leftrightarrow$ (iii) Let $s^1,\dots,s^K,M$ be an instance of LCS-1, let $G=(V,E)$ with coloring $c_G:E\rightarrow C$ be the edge colored graph constructed as in part (i) of this proof, and let $H=(V\cup E,F)$ with vertex coloring $c_H:V\cup E\rightarrow C\cup\{a\}$ be the vertex colored graph constructed from G as in part (ii). The following now holds:

Claim 8.3. *Let H be constructed as above. H is a subgraph of a properly colored (by c_H) interval graph if and only if H is a subgraph of a properly colored (by c_H) triangulated graph.*

Proof. (Only if) Trivial.

(If) Suppose H is a subgraph of a properly colored triangulated graph H'. There exists a tree-decomposition $(\{X_i \mid i\in I\},T=(I,\mathscr{F}))$ of H', such that for all $i\in I$, X_i is a clique in H' and hence no two vertices in X_i have the same color.

Note that vertices v_1^1 and v_2^1 and the edges between them in G form a complete bipartite subgraph in H. By Lemma 3, we know that either there exists an $i_0\in I$ with $v_1^1,v_2^1\in X_{i_0}$ or there exists an $i_0\in I$ such that $v_1^1\in X_{i_0}$ and all edges (v_1^1,v_2^1) belong to X_{i_0}. The former cannot be the case as v_1^1 and v_2^1 have the same color hence we may assume the latter. There cannot be any other vertices included in X_{i_0} since v_1^1 and its adjacent edges together have all possible colors. Similarly, there exists a node i_1 such that X_{i_1} contains precisely v_2^2 and all edges (v_1^2,v_2^2). We may suppose that i_0 and i_1 are leaves of T, as neither set X_{i_0} or X_{i_1} is a separator of H.

Let I' be the set of all nodes $i\in I$ that are on the path from i_0 to i_1 in T (i_0, i_1 inclusive).

We claim that for all $v\in V$ there exists an $i\in I'$ with $v\in X_i$. Suppose the contrary. Note that v is on a path in G from v_1^1 to v_2^2 with all edges of the same color, say c_α. This path corresponds to a path Y in H from v_1^1 to v_2^2 containing v where vertices are alternately colored c_α and a. Let $Y=y_0,y_1,\dots,y_q$ where $y_0=v_1^1$, $y_q=v_2^2$ and suppose $v=y_j$. Let $v\in X_{i_2}$ and let i_3 be the first node on the path from i_2 to i_0. X_{i_3} must contain a vertex y_{j_1} with $j_1<j$ and a vertex y_{j_2} with $j_2>j$ (by Lemma 2). Now the subpath of Y between y_{j_1} and y_{j_2} forms a cycle with the edge (y_{j_1},y_{j_2}) in H'. Therefore, H' contains a cycle with only two colors used for the vertices on the cycle; and this contradicts Lemma 5. Hence, for all $v\in V$, there exists an $i\in I$ with $v\in X_i$.

We now claim that, for all $(v,w)\in E$, there exists an $i\in I'$ with $v,(v,w)\in X_i$. For all $z\in V\cup E$, let $I_z=\{i\in I \mid z\in X_i\}$. There exist nodes $i_4\in I_v\cap I_{(v,w)}$, $i_5\in I_w\cap I_{(v,w)}$. Let i_6 be the first node in I' on the path from i_4 to i_5. Since T is a tree, $I_v\cap I_w=\emptyset$, $I_v\cap I'\neq\emptyset$, and $I_w\cap I'\neq\emptyset$, i_6 must exist. We must have that $(v,w)\in X_{i_6}$, (by definition of a tree-decomposition) and $v\in X_{i_6}$ as otherwise, for all $i\in I'$, $v\notin X_i$.

We now can conclude that $(\{X_i \mid i\in I'\},T[I'])$ is a path-decomposition of H, for which for all $i\in I'$ it is true that all vertices in X_i have different colors. So H is a subgraph of a properly colored interval graph. □

The preceding claim gives us a transformation from LCS-1 to TCG and part (iii) of the theorem follows. □

Corollary 9. *The following problems are* NP-*Complete*: *CC*-1, *ICG*, *TCG*.

Proof. Membership in NP is trivial. Note that the reduction used in Theorem 8 is many:1. □

Corollary 10. *For every class of graphs $\mathscr{G}$ such that every graph in $\mathscr{G}$ is chordal and every interval graph belongs to $\mathscr{G}$, the following problem is $W[t]$-Hard for all $t \in \mathbf{N}$ and its unparameterized version is* NP-Hard:

Instance: *Graph $G=(V,E)$, coloring $c: V \to C$.*

Parameter: $|C| = k$.

Question: *Does there exist a graph $H \in \mathscr{G}$, that contains G as a subgraph and is properly colored by c?*

Proof. Note that in Claim 8.3 the statements are also equivalent to the following statement:

H is a subgraph of a properly colored graph $H' \in \mathscr{G}$.

Therefore the same reduction from LCS-1 can be used. □

Corollary 11. *The following are $W[t]$-Hard for all $t \in \mathbf{N}$ and NP-Complete*:

(i) DIRECTEDCC-1, (ii) DIRECTEDCC-1 FOR GRAPHS WITH ONLY ONE VERTEX WITH OUTDEGREE 0 *and* (iii) MODIFIEDCC-1.

The proofs of these consist of easy modifications of the above arguments and are omitted.

The version of DIRECTEDCC-1 with only one vertex with outdegree 0 will be used in a subsequent proof for the parameterized hardness of FRA.

4.2. Hardness for Feasible Register Assignment (FRA)

Theorem 12. FEASIBLE REGISTER ASSIGNMENT *is $W[t]$-Complete for all $t \in \mathbf{N}$.*

Proof. We reduce from DIRECTEDCC-1 WITH ONLY ONE VERTEX OF OUTDEGREE 0. By Corollary 11, the $W[t]$-Hardness for all $t \in \mathbf{N}$ of FRA follows.

Let $G=(V,E)$ be a directed acyclic graph with z the unique node in G with out-degree 0 and let $c_G: E \to C$ be an edge coloring of G where $C = \{1,2,\ldots,k\}$. Our argument is similar to that used for ICG.

Let $H=(V \cup E, F)$ be the directed, acyclic graph defined by $F = \{(v,(v,w)) \mid v \in V, (v,w) \in E\} \cup \{((v,w),w) \mid w \in V, (v,w) \in E\}$; that is, H is obtained by subdividing every edge of G whilst retaining the same directions for edges. Note that H is a directed acyclic graph. We define a register assignment r (or, equivalently, a coloring) of the vertices of H as follows: $\forall v \in V$: $r(v) = R_0$; $\forall e \in E$: $r(e) = R_{c_G(e)}$.

Claim 12.1. *H^R with register assignment r is a "yes"-instance to the FRA problem if and only if the directed colored cutwidth of G with coloring c_G is* 1.

Proof. (Only if) Let f, $S_0,\dots,S_{|V|+|E|}$ be a solution to the FRA problem for H^R and r. Note that z is the unique node in H with indegree 0, and hence $f(z)=|V|+|E|$. Let g be the linear ordering of G, such that for all $v,w\in V$: $f(v)<f(w)\Leftrightarrow g(v)<g(w)$. As f is a topological order of $H=(H^R)^R$, we have that g is a topological order of G. We claim the directed colored cutwidth of g is 1. Suppose there is a color conflict between edges (v,w), (x,y) in E. Let $u=g^{-1}(g(v)+1)$, i.e., u is the next vertex in V after v, in both orderings f and g. v cannot belong to $S_{f(u)}$, as u and v have the same color. So, if $f((w,v))>f(u)$, we get a contradiction. So, $f(v)<f((w,v))<f(u)$, and (w,v) belongs to all sets S_i, with $f(u)-1\leqslant i\leqslant f(w)-1$. A similar analysis holds for the edge (x,y) (or vertex (y,x)). Case analysis now shows there is a set S_i with $(w,v),(y,x)\in S_i$. This is a contradiction as both of these two vertices are assigned to the same register.

(If) Let g be a topological sort of G with directed colored cutwidth 1. Take a linear ordering f of H^R that fulfils: $\forall v,w\in V: f(v)<f(w)\Leftrightarrow g(v)<g(w)$, and $\forall v\in V,(v,w)\in E: f(v)<f((w,v))<f(g^{-1}(g(v)+1))$, i.e., all vertices (w,v) representing a reversed edge (v,w), are placed after v in the ordering f, but before the next vertex from G. f is a topological order of H^R.

Let $S'_0,\dots,S'_{|V|+|E|}$ be defined as in Lemma 7. We must verify that for all S'_i all vertices have a different register assigned to them. We cannot have two vertices with register R_0 in the same set S'_i, as these are vertices in V, and all successor of a vertex in V are placed in the ordering f before the next vertex in V, i.e., before the next vertex that is assigned to R_0. Also, the only vertex with indegree 0 in H^R is z, and z belongs only to $S'_{|V|+|E|}$ and no other S_i. Suppose now there exist vertices $(w,v),(y,x)\in S_i$, with $R_{(w,v)}=R_{(y,x)}$. There is a color conflict (w.r.t. g) between the edges (v,w), and (x,y): $f(v)<f((w,v))\leqslant i<f(w)$, and $f(x)<f((y,x))\leqslant i<f(y)$, hence the open intervals $(g(v),g(w))$ and $(g(x),g(y))$ intersect. This is a contradiction. Therefore f must satisfy the conditions of Lemma 7. □

This completes the theorem. □

4.3. Hardness for module allocation on graphs with bounded pathwidth

Theorem 13. Module Allocation on graphs of bounded pathwidth *is* $W[t]$*-Hard, for all* t*, even when all communication costs are restricted to* 0 *or* 1.

Proof. We reduce from LCS-1. Let $s^1,\dots,s^K\in\Sigma^*$ and integer M be our instance of LCS-1. Denote by L_i the length of string s^i. Denote by s^i_j the jth character of string i.

We create a graph $G=(V,E)$ with $K\cdot M$ vertices, as follows:

$$V=\{v_{i,j}\mid 1\leqslant i\leqslant M,\ 1\leqslant j\leqslant K\},$$

$$E=\{(v_{k,l},v_{k',l'})\mid (k=k'\wedge l\neq l')\vee(k'=k+1\wedge l=l')\}.$$

It is easy to see that the pathwidth of G is at most $K+1$.

We create $\sum_{i=1}^{K} L_i$ processors where each processor corresponds to one character in one of the strings. We write processor $p_{i,j}$ for the processor that corresponds to the ith character in string s^j.

We assign costs as follows:

The execution cost for all modules $v_{k,l}$ and processors $p_{i,j}$ is defined to be:

$$e(v_{k,l}, p_{i,j}) = 0.$$

The communication cost between module $v_{k,l}$ and module $v_{k,l'}$ when assigned to processors $p_{i,j}$ and $p_{i',j'}$ respectively where $i \neq i'$ and $l \neq l'$ is defined to be:

$$C((v_{k,l}, p_{i,j}), (v_{k,l'}, p_{i'j'})) = \begin{cases} 0 & \text{if } s_i^j = s_{i'}^{j'}, \\ 1 & \text{otherwise.} \end{cases}$$

The communication cost between module $v_{k,l}$ and module $v_{k+1,l}$ when assigned to processors $p_{i,j}$ and $p_{i',j'}$ respectively is defined to be:

$$C((v_{k,l}, p_{i,j}), (v_{k+1,l}, p_{i',j'})) = \begin{cases} 0 & \text{if } l = j = j' \text{ and } i' > i, \\ 1 & \text{otherwise.} \end{cases}$$

Let $C((x, y), (x', y')) = 0$, for all y and y' when $(x, x') \notin E$.

Claim 13.1. *Let A be the instance of the* Module Allocation *problem constructed above. There exists a module assignment of total cost* 0 *if and only if the strings $s^1, \ldots, s^K$ have a common subsequence of length M.*

Proof. (If) Suppose the common subsequence is of the form $s^i_{f_i(1)} \cdots s^i_{f_i(M)}$, $f_i(1) < f_i(2) < \cdots < f_i(M)$, for all i, $1 \leqslant i \leqslant K$. Now assign each module $v_{k,l}$ to processor $p_{f_l(k),l}$. One can verify that this gives cost 0. For instance, for modules $v_{k,l}$ and $v_{k,l'}$ the communication cost is 0, as $s^l_{f_l(k)}$ and $s^{l'}_{f_l(k')}$ (the kth character in the common substring) are equal.

(Only if) Suppose we have a module allocation with cost 0. To get a communication cost of 0 between modules $v_{k,l}$ and $v_{k+1,l}$, we must assign each module $v_{k,l}$ to a processor $p_{f_l(k),l}$, for some $f_l(k)$, $1 \leqslant f_l(k) \leqslant L_l$. Moreover, it must be the case that $f_l(k) < f_l(k+1)$. So, each sequence $s^l_{f_l(1)} \cdots s^l_{f_l(M)}$ forms a subsequence of the string s^l.

These subsequences must be equal. The kth character of the ith subsequence is $s^i_{f_i(k)}$. Take $i \neq i'$. As the communication cost between $v_{k,i}$ and $v_{k,i'}$ must be 0, it follows that $s^i_{f_i(k)} = s^{i'}_{f_{i'}(k)}$. □

From the above claim and the $W[t]$-Hardness, for all $t \in \mathbf{N}$ of the LCS-1, the result follows. □

Corollary 14. Parametric MA *is $W[t]$-Hard for all $t \in \mathbf{N}$.*

Proof. Note that all costs are either 0 or 1 and therefore trivially linear functions of the parameter τ. □

5. Conclusions

In this paper, we have shown hardness for several graph problems on bounded width graphs. All of the problems considered are NP-Complete and $W[t]$-Hard for all $t \in \mathbf{N}$.

Recently in [9] it has been shown that ICG is in fact NP-Complete for any fixed $k \geqslant 4$. For the case of $k=3$, they give an $O(|V_G|^2)$ algorithm. This provides an interesting contrast with Triangulating Colored Graphs, which can be solved in time $O(|V_G|^{k+1})$ for any fixed k [43], see also [3].

For Module Allocation and Parametric Module Allocation, our results suggest that the algorithms found in [29, 30] respectively are in some sense optimal. The algorithm for Module Allocation has running time $O(mp^{k+1})$ and it appears unlikely that the factor of k can be removed from the exponent. Likewise, the same conclusion can be drawn for Parametric Module Allocation although it remains open whether algorithms without the factor of p in the exponent exist.

Membership in the W hierarchy remains open for all of these problems although it is noted that the result of [9] implies that ICG is not in any level of the W-hierarchy unless $P = NP$.

References

[1] K. Abrahamson, M.R. Fellows, Finite automata, bounded treewidth and well-quasiordering, Contemp. Math. 147 (1993) 539–563.

[2] R. Agarwala, D. Fernández-Baca, A polynomial-time algorithm for the perfect phylogeny problem when the number of character-states is fixed, SIAM J. Comput. 23 (6) (1994) 1216–1224.

[3] R. Agarwala, D. Fernández-Baca, Fast and simple algorithms for perfect phylogeny and triangulating colored graphs, Internat. J. Found. Comput. Sci. 7 (1) (1996) 11–22.

[4] A.V. Aho, J.D. Ullman, Optimization of straight line programs, SIAM J. Comput. 1 (1972) 1–19.

[5] S. Arnborg, Efficient algorithms for combinatorial problems on graphs with bounded decomposability: a survey, BIT 25 (1985) 2–23.

[6] H.L. Bodlaender, Dynamic programming algorithms on graphs with bounded tree-width, in: Proc. 15th Internat. Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 317, Springer, Berlin, 1998, pp. 105–119.

[7] H.L. Bodlaender, A linear time algorithm for finding tree-decompositions of small treewidth, SIAM J. Comput. 25 (1996) 1305–1317.

[8] H.L. Bodlaender, R.G. Downey, M.R. Fellows, H.T. Wareham, The parameterized complexity of sequence alignment and consensus, Theoret. Comput. Sci. 147 (1995) 31–54.

[9] H.L. Bodlaender, B. de Fluiter, Intervalizing k-colored graphs, in: Proc. 22nd Internat. Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 944, Springer, Berlin, 1995, pp. 87–98.

[10] H.L. Bodlaender, M.R. Fellows, M.T. Hallett, Beyond NP-completeness for problems of bounded width: hardness for the W hierarchy, in: Proc. 26th Annu. ACM Symp. on the Theory of Computing, 1994, pp. 449–458.

[11] H.L. Bodlaender, M.R. Fellows, T.J. Warnow, Two strikes against perfect phylogeny, in: Proc. 19th Internat. Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 623, Springer, Berlin, 1992, pp. 373–383.

[12] H.L. Bodlaender, T. Kloks, Efficient and constructive algorithms for the pathwidth and treewidth of graphs, J. Algorithms, 21 (1996) 358–402.

[13] H.L. Bodlaender, R.H. Möhring, The pathwidth and treewidth of cographs, SIAM J. Discrete Math. 6 (1993) 181–188.

[14] S.H. Bokhari, A shortest tree algorithm for optimal assignments across space and time in distributed processor systems, IEEE Trans. Software Eng. SE-7 (6) (1981) 583–589.
[15] R.B. Borie, R.G. Parker, C.A. Tovey, Automatic generation of linear-time algorithms from predicate calculus and descriptions of problems on recursively constructed graph families, Algorithmica 7 (1992) 555–582.
[16] P. Buneman, A characterization of rigid circuit graphs, Discrete Math. 9 (1974) 205–212.
[17] L. Cai, J. Chen, R.G. Downey, M.R. Fellows, The parameterized complexity of short computations and factorizations, Technical report, Department of Computer Science, University of Victoria, July 1993.
[18] N.G. Cooper (Ed.), The human genome project, Los Alamos Sci. 20 (1992) 119.
[19] B. Courcelle, The monadic second-order logic of graphs I: recognizable sets of finite graphs, Inform. and Comput. 85 (1990) 12–75.
[20] W.H. Day, D. Sankoff, Computational complexity of inferring phylogenies by compatibility, Systematic Zoology 35 (2) (1986) 224–229.
[21] R.G. Downey, P.A. Evans, M.R. Fellows, Parameterized learning complexity, in: Proc. 6th ACM Workshop on Computational Learning Theory (COLT), 1993, pp. 51–57.
[22] R.G. Downey, M.R. Fellows, Fixed-parameter intractability (extended abstract), Proc. 7th Annual Conf. on Structure in Complexity Theory (Structures'92), 1992, pp. 36–49.
[23] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness III: some structural aspects of the W-hierarchy, in: K. Ambos-Spies, S. Homer, U. Schoning (Eds.), Complexity Theory, Cambridge University Press, Cambridge, 1993, pp. 166–191.
[24] R.G. Downey, M.R. Fellows, Fixed-parameter tractability and completeness I: basic results, SIAM J. Comput. 24 (1995) 873–921.
[25] R.G. Downey, M.R. Fellows, Fixed parameter tractability and completeness II: on completeness for $W[1]$, Theoret. Comput. Sci. 141 (1995) 109–131.
[26] R.G. Downey, M.R. Fellows, B.M. Kapron, M.T. Hallett, H.T. Wareham, The parameterized complexity of some problems in logic and linguistics, in: Proc. Symp. on Logical Foundations of Computer Science (LFCS '94), Lecture notes in Computer Science, vol. 813, Springer, Berlin, 1994, pp. 89–100.
[27] G.F. Estabrook, Some concepts for the estimation of evolutionary relationships in systematic botany, Systematic Botany 3 (2) (1978) 146–158.
[28] M.R. Fellows, M.T. Hallett, H.T. Wareham, DNA physical mapping: 3 ways difficult, in: Tom Lengauer (Ed.), Proc. 1st Annu. European Symp. on Algorithms (ESA'93), Lecture Notes in Computer Science, vol. 726, Springer, Berlin, pp. 157–168.
[29] D. Fernàndez-Baca, Allocating modules to processors in a distributed system, IEEE Trans. Software Eng. 15 (11) (1989) 1427–1436.
[30] D. Fernàndez-Baca, A. Medipalli, Parametric module allocation on partial k-trees, IEEE Trans. Computers 42 (1993) 738–742.
[31] M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, Academic Press, New York, 1980.
[32] M.C. Golumbic, H. Kaplan, R. Shamir, On the complexity of physical mapping, Adv. Appl. Math. 15 (1994) 251–261.
[33] E.M. Gurari, I.H. Sudborough, Improved dynamic programming algorithms for bandwidth minimization and the mincut linear arrangement problem, J. Algorithms 5 (1984) 531–546.
[34] D. Gusfield, Efficient algorithms for inferring evolutionary trees, Networks 21 (1981) 19–28.
[35] M.T. Hallett, H.T. Wareham, A compendium of parameterized results, SIGACT News 25 (3) (1994) 122–123.
[36] R. Idury, A. Schaffer, Triangulating three-colored graphs in linear time and linear space, SIAM J. Discrete Math. 2 (1993) 289–293.
[37] S. Kannan, T.J. Warnow, Inferring evolutionary history from DNA sequences, in: Proc. 31rd Annu. Symp. on Foundations of Computer Science, 1990, pp. 362–371.
[38] S.K. Kannan, T.K. Warnow, Triangulating 3-colored graphs, SIAM J. Discrete Math. 5 (1992) 249–258.
[39] S.K. Kannan, T.J. Warnow, Inferring evolutionary history from DNA sequences, SIAM J. Comput. 23 (4) (1994) 713–737.
[40] H. Kaplan, R. Shamir, Pathwidth, bandwidth and completion problems to proper interval graphs with small cliques, SIAM J. Comput. 25 (1996) 540–561.
[41] H. Kaplan, R. Shamir, R.E. Tarjan, Tractability of parameterized completion problems on chordal and interval graphs, Found Comput. Ser. (1994) 780–791.

[42] C.A. Meacham, G.F. Estabrook, Compatibility methods in systematics, Annu. Rev. Ecol. Systematics 16 (1985) 431–446.
[43] F.R. McMorris, T.J. Warnow, T. Wimer, Triangulating vertex-colored graphs, SIAM J. Discrete Math. 7 (2) (1994) 296–306.
[44] S.-I. Nakano, T. Oguma, T. Nishizeki, A linear time algorithm for c-triangulating three-colored graphs, Trans. Inst. Electron. Inform. Commun. Eng. A 377-A (3) (1994) 543–546 (in Japanese).
[45] R. Sethi, Complete register allocation problems, SIAM J. Comput. 4 (1975) 226–248.
[46] R. Sethi, J.D. Ullman, The generation of optimal code for arithmetic expressions, J. Assoc. Comput. Mach. 17 (1970) 715–728.
[47] M.A. Steel, The complexity of reconstructing trees from qualitative characters and subtrees, J. Classification 9 (1992) 91–116.
[48] H. Stone, Critical load factors in two-processor distributed systems, IEEE Trans. Software Eng. SE-4 (1978) 254–258.
[49] D. Towsley, Allocating programs containing branches and loops within a multiple processor system, IEEE Trans. Software Eng. SE-12 (10) (1986) 1018–1024.
[50] T. Warnow, Combinatorial algorithms for constructing phylogenetic trees, Ph.D. Thesis, University of California, Berkeley, 1991.
[51] T.V. Wimer, Linear algorithms on k-terminal graphs, Ph.D. Thesis, Dept. Computer Science, Clemson University, 1987.