Beyond *NP*-Completeness for Problems of Bounded Width: Hardness for the W Hierarchy (Extended Abstract)

Hans L. Bodlaender * Michael R. Fellows † Michael T. Hallett ‡

October 27, 1994

Abstract

The parameterized computational complexity of a collection of well-known problems including: Bandwidth, Precedence constrained k-processor scheduling, Longest Common Subsequence, DNA physical mapping (or Intervalizing colored graphs), Perfect phylogeny (or Triangulating colored graphs), Colored cutwidth, and Feasible register assignment is explored. It is shown that these problems are hard for various levels of the W hierarchy. In the case of Precedence constrained k-processor scheduling the results can be interpreted as providing substantial new complexity lower bounds on the outcome of [OPEN 8] of the Garey and Johnson list. We also obtain the conjectured "third strike" against Perfect phylogeny.

1 Introduction

All of the well-known problems: Min cut linear arrangement, Bandwidth and Precedence constrained k-processor scheduling have important applications for small fixed parameter values. All three of these problems are *NP*-complete in general; the first two are known to be solvable in polynomial-time for each fixed parameter value, and whether for the third this is also true is essentially [OPEN 8] of the famous list of Garey and Johnson [24].

*Department of Computer Science, Utrecht University, Padualaan 14, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands. Email address: hansb@cs.ruu.nl

†Department of Computer Science, University of Victoria, Victoria, British Columbia V8W 3P6, Canada. Research supported by the National Science and Engineering Research Council of Canada under grant OGP0089820, and by the United States National Science Foundation under grant MIP-8919312. Email address: mfellows@csr.uvic.ca

‡Department of Computer Science, University of Victoria, Victoria, British Columbia V8W 3P6, Canada. Email address: mhallett@sanjuan.uvic.ca

After several rounds of progress, we now know that there is a striking contrast between the fixed-parameter complexities of MIN CUT LINEAR ARRANGEMENT and BANDWIDTH. Combining several different lines of development [1] [7], the first can be solved in linear time for each fixed k. The best known algorithms for the latter problem require time $\Omega(n^k)$.

Note that *NP*-completeness has nothing to say about this; it is entirely consistent with $P \neq NP$ for all three problems to be linear-time solvable for each fixed k, and this could be of substantial practical importance. We apply the theory of parameterized computational complexity introduced in [DF1-7] to explore this possibility, obtaining evidence that BANDWIDTH and PRECEDENCE CONSTRAINED k-PROCESSOR SCHEDULING are qualitatively more difficult with respect to fixed-parameter complexity than MIN CUT LINEAR ARRANGEMENT.

The main significance is in the results, rather than in the techniques (which are simply very demanding, multi-stage combinatorial reductions). In the case of PRECEDENCE CONSTRAINED k-PROCESSOR SCHEDULING our result can be interpreted as bearing on the practical significance of [OPEN 8], showing that even if there is no particular k for which the problem is *NP*-complete, it is still likely to be computationally intractable for the fixed parameter values that are important in applications.

1.1 Problems that are fixed parameter intractable

The focus of this paper is on the following problems (the parameter is always k).

(1) PRECEDENCE CONSTRAINED k-PROCESSOR SCHEDULING. An instance consists of a set T of unit length tasks, a partial order $\prec$ on T, a deadline $D \in N^+$ and a number of processors k. The question is whether there is a map $f : T \to \{1, \ldots, D\}$, such that for all $t, t' \in T$: $t \prec t'$ implies $f(t) < f(t')$, and for all i, $1 \leq i \leq D$: $|f^{-1}(i)| \leq k$.

(2) LONGEST COMMON SUBSEQUENCE (LCS). An instance consists of k strings $\{X_i : 1 \leq i \leq k\}$ over an alphabet Σ, and a positive integer m. The question is whether there is a string $Y \in \Sigma^*$ of length at least m that is a subsequence of X_i for all i, $1 \leq i \leq k$.

(3) BANDWIDTH. An instance consists of a graph $G = (V, E)$ and a positive integer k. The question is whether there is a 1:1 "linear layout" $f : V \to \{1, \ldots, |V|\}$ such that $uv \in E$ implies $|f(u) - f(v)| \leq k$.

(4) PERFECT PHYLOGENY (TRIANGULATING COLORED GRAPHS, TCG). An instance consists of a graph $G = (V, E)$ and a coloring $c : V \to \{1, \ldots, k\}$. The question is whether there a supergraph $G' = (V, E')$ of G which is properly colored by c and which is triangulated (equivalently, chordal).

(5) DNA PHYSICAL MAPPING (INTERVALIZING COLORED GRAPHS, ICG). An instance consists of a graph $G = (V, E)$ and a coloring $c : V \to \{1, \ldots, k\}$. The question is whether there is a supergraph $G' = (V, E')$ of G which is properly colored by c and that is an interval graph.

(6) COLORED CUTWIDTH. An instance consists of a graph $G = (V, E)$, an edge coloring $c : E \to \{1, \ldots, r\}$ and a positive integer k. The question is whether there is a 1:1 linear layout

$f : V \to \{1, \ldots, |V|\}$ such that for each color $j \in \{1, \ldots, k\}$ and for every i, $1 \leq i \leq |V| - 1$, we have $|\{uv : c(uv) = j \text{ and } f(u) \leq i \text{ and } f(v) \geq i + 1\}| \leq r$.

(7) DIRECTED BANDWIDTH. Same as *Bandwidth*. The layout must respect arc direction.

(8) FEASIBLE REGISTER ASSIGNMENT. See [24] for the definition.

(9) UNIFORM EMULATION ON A PATH. An instance consists of a graph $G = (V, E)$, and an integer k. The question is whether there exists a function $f : V \to \{1, \ldots, |V|/k$, such that for all $uv \in E$ implies $|f(v) - f(w)| \leq 1$, and for all i, $|f^{-1}(i)| \leq k$.

(10) UNIFORM EMULATION ON A RING. See [6] for the definition.

(11) MODULE ALLOCATION FOR BOUNDED TREEWIDTH NETWORKS. See [23] for the definition.

Most of these problems are well studied and have important applications e.g. for mathematical biology [2,4,5], Choleski factorization [3,8], VLSI-theory [6]. In most cases, it is known that these problems are polynomial time solvable when the parameter is fixed (see, amongst others [6, 23, 24, 26, 27, 29]), but all of these algorithms require time $\Omega(n^k)$.

Let the pair (G, k) denote an instance of problems (3)–(11), where k denotes the parameter. For each of these problems there is a theorem that tells us, "If the answer is *yes*, then the treewidth (or pathwidth) of the graph G (which may be colored or directed) is bounded by a function $b(k)$." For example, a yes-instance of ICG has pathwidth at most $k - 1$. A yes-instance of TCG has treewidth at most $k - 1$. The bandwidth of a graph is never larger than its pathwidth. A yes-instance of FEASIBLE REGISTER ASSIGNMENT has pathwidth at most k. A yes-instance of COLORED CUTWIDTH has pathwidth at most $k \cdot r - 1$.

This opens up the following important and very general possibility. First, by means of the algorithm of Bodlaender [7] (for $k = 4$ see [28]) that runs in time $O(n)$ (for fixed k) we can either determine, for input (G, k), that the answer is "no," or obtain a tree-decomposition of G of width at most $b(k)$. Given such a tree-decomposition, we may well hope to answer the question in linear time (for fixed k) by means of one of the general methods for algorithm design in this setting [1] [2] [3] [4] [5] [9] [11] [30].

In precisely this way it can be shown that many parameterized graph problems (concerning possibly colored and directed graphs), including TREEWIDTH, PATHWIDTH, FEEDBACK VERTEX SET, FEEDBACK ARC SET, SEARCH NUMBER, REGISTER SUFFICIENCY, and MIN CUT LINEAR ARRANGEMENT can be solved in linear time for every fixed k. The similarities of some of these to problems (3)–(11) should be noted.

Because they have the "bounded width" property it is natural to ask whether any of the problems (3)–(11) might also be amenable to this general approach, or something similar. Our results show that no such approach can succeed unless the W hierarchy collapses.

1.2 Main Results

Our main purpose is to show that problems (2)–(11) are hard for $W[t]$ for all t. We conjecture that a similar result holds for PRECEDENCE CONSTRAINED k-PROCESSOR SCHEDULING, however at present we are only able to show hardness for $W[2]$.

ICG was shown to be hard for $W[1]$ in [21]. ICG, TCG and Bandwidth were shown to be non-finite-state, and hence not directly amenable to the usual bounded treewidth techniques, in [21], [8] and [1], respectively. These latter results left open the possibility that fixed-parameter tractability might be achievable by some other approach.

Theorem 1. Problem (1) is hard for $W[2]$.

Theorem 2. Problems (2)–(11) are hard for $W[t]$ for all t. Problem (3), (9), and (10) remain hard for $W[t]$ for all t, when restricted to *trees*.

All of the problems listed have applications for "small" parameter values. Thus NP-completeness could be regarded as irrelevant — unless a problem were NP-complete for a small fixed parameter value. The failure to find such a demonstration for PRECEDENCE CONSTRAINED k-PROCESSOR SCHEDULING has led to some speculation that the problem might be (in our terminology) fixed-parameter tractable, which could be useful. Our result shows that even if there is no parameter value for which the problem is *NP*-complete, it is still likely to be an impractically hard problem for fixed parameter values.

1.3 Parameterized Computational Complexity

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 $L \subseteq \Sigma^* \times N$. For a parameterized problem L and $k \in N$ we write L_k to denote the associated fixed-parameter problem $L_k = \{x | (x, k) \in L\}$. 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 $(x', g(k)) \in B$.

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. Let F be a family of decision circuits (possibly having many different circuits with a given number of inputs). We associate to 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. This gives us the hierarchy of parameterized complexity classes

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

for which there are many natural hard or complete problems [14] [16] [19]. 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 [10] [12] [20]. Thus, any one of these problems is fixed-parameter tractable if and only if all of the others are; and none of the problems for which we here prove W hardness results are fixed-parameter tractable unless all of these are also. DOMINATING SET is proved complete for $W[2]$ in [13]. Fixed parameter tractability for DOMINATING SET, or any other $W[2]$-hard problem implies fixed parameter tractability for all problems in $W[1]$ mentioned above, and all other problems in $W[2] \supseteq W[1]$.

2 The Reductions

Because of the requirements on how the parameter must be encoded, the search for combinatorial reductions in the theory of parameterized complexity tends to be considerably more difficult than in the theory of *NP*-completeness. For the same reason, other issues, such as the parameterized complexity analog of Ladner's Theorem, also present greater technical demands [17]. From a bird's eye view, the conclusions of Theorem 2 follow in all cases from a series of quite complicated reductions, beginning with the completeness for $W[t]$, for t even, of MONOTONE NORMALIZED WEIGHTED SATISFIABILITY established in [14]. From there, by a reduction described below, we show that LCS is hard for $W[t]$ for all t. We also have a reduction from MONOTONE NORMALIZED WEIGHTED SATISFIABILITY to UNIFORM EMULATION ON A PATH. Basically the same reduction can be used to show hardness for UNIFORM EMULATION ON A RING.

From LCS, we have a reduction to COLORED CUTWIDTH with the 'cutwidth per color' $r = 1$, and from this version of COLORED CUTWIDTH, we have a reduction to ICG. A rather technical argument shows that the concatenation of these latter two reductions is, in fact, also a reduction from LCS to TCG. In a similar way, hardness of FEASIBLE REGISTER ASSIGNMENT is shown. A somewhat less complex reduction from UNIFORM EMULATION ON A PATH shows $W[t]$-hardness of BANDWIDTH. DIRECTED BANDWIDTH is shown to be $W[t]$-hard in a very way, via UNIFORM EMULATION ON A PATH for dags.

Because of the space limitations of this abstract, we can only provide a sample of these reductions. We next describe (1) the reduction showing that PRECEDENCE CONSTRAINED

k-PROCESSOR SCHEDULING is hard for $W[2]$ (our easiest but perhaps most important result), and (2) the proof of $W[t]$-hardness of COLORED CUTWIDTH, ICG, and TCG from the intermediate point of LCS.

Proposition 2.1 PRECEDENCE CONSTRAINED k-PROCESSOR SCHEDULING *is hard for* $W[2]$.

Proof: We transform from DOMINATING SET. Let $(G = (V, E), k)$ be an instance to DOMINATING SET. Suppose $|V| = n$, and write $V = \{v_0, \ldots, v_{n-1}\}$. Write $c = n^2 + 1$. Take $D = (k \cdot n) \cdot c + 2n$, and take $K = 2k + 1$. Define a directed acyclic graph $H = (W, F)$ consisting of:
The floor. Take a path with length D: take vertices $\{a_1, \ldots, a_D\}$, and edges (a_i, a_{i+1}) for all i, $1 \leq i \leq D - 1$.
The floor gadgets. 'Parallel' to each floor vertex of the form $a_{n-1+\alpha\cdot c+in}$, $1 \leq i \leq n$, $0 \leq \alpha \leq kn - 1$, we take a floor gadget vertex: take vertices $\{b_{n-1+\alpha\cdot c+in} \mid 1 \leq i \leq n,\ 0 \leq \alpha \leq kn - 1\} = \mathcal{B}$, and add edges: (a_{i-1}, b_i) and (b_i, a_{i+1}) for all $b_i \in \mathcal{B}$.
The selector paths. For each i, $1 \leq i \leq k$, we take a path of length $D - n + 1$. This path will represent the ith vertex from a dominating set of G. Take vertices $\{c_{i,j} \mid 1 \leq i \leq k,\ 1 \leq j \leq D - n\}$, and edges $(c_{i,j}, c_{i,j+1})$ for all i, $1 \leq i \leq k$, j, $1 \leq j \leq D - n$.
The selector gadgets. If $i \neq j$ and $(v_i, v_j) \notin E$, then we take a vertex, which is put 'parallel' to $c_{r,n-1+\alpha\cdot c+in-j}$, for all α, $1 \leq \alpha \leq k \cdot n$, r, $1 \leq r \leq k$. Take vertices $\{d_{r,n-1+\alpha\cdot c+in-j} \mid 1 \leq r \leq k,\ 1 \leq i \leq n,\ 1 \leq j \leq n,\ i \neq j,\ (v_i, v_j) \notin E,\ 1 \leq \alpha \leq kn\} = \mathcal{D}$, and for each vertex $d_{r,\beta} \in \mathcal{D}$, add edges $(c_{r,\beta-1}, d_{r,\beta})$ and $(d_{r,\beta}, c_{r,\beta+1})$.

Let H be the dag, resulting from this construction, and let $\prec$ be the transitive closure of H, i.e, take $v \prec w$, iff there exists a path from v to w in H.

Claim 2.2 *Task set W with partial order $\prec$, deadline D, and number of processors K, is a yes-instance to* PRECEDENCE CONSTRAINED K-PROCESSOR SCHEDULING, *if and only if G has a dominating set of size k.*

Proof: $\Leftarrow$: Suppose $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\} \subseteq V$ is a dominating set of size k of G. Consider the following schedule f of W:
(1) $a_i = i (1 \leq i \leq D)$
(2) $b_i = i (b_i \in \mathcal{B})$
(3) $c_{i,j} = j + \gamma_i (1 \leq i \leq D - n)$
(4) $d_{i,j} = j + \gamma_i (d_{i,j} \in \mathcal{D})$

Clearly f satisfies the precendence constraints. To an integer i, not of the form $n - 1 + \alpha \cdot c + jn$ $(1 \leq j \leq n, 1 \leq \alpha \leq kn)$, one floor vertex, no floor gadget vertex, $\leq k$ selector path vertices, and $\leq k$ selector gadget vertices are mapped, so for such i, $|f^{-1}(i)| \leq 2k + 1 = K$.

Look at i of the form $n - 1 + \alpha \cdot c + jn$ with $1 \leq j \leq n$, $1 \leq \alpha \leq kn$. As $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\}$ is a dominating set of G, there are two cases:

Case 1: v_p is in the dominating set, i.e., $p = \gamma_q$, $1 \leq q \leq k$. As $d_{q,n-1+\alpha\cdot c+pn-p}$ does not exist in $\mathcal{D}$, at most $k-1$ selector gadget vertices are mapped to $i = n-1+\alpha\cdot n^2+pn-p+\gamma_q$.

The total number of vertices mapped to i hence is at most K. (The other vertices mapped to i are: at most one floor vertex, one floor gadget vertex, and k selector path vertices.)

Case 2: v_p is adjacent to vertex v_{γ_q}, $1 \leq q \leq k$. Now $d_{q,n-1+\alpha\cdot c+pn-\gamma_q}$ does not exist in $\mathcal{D}$, so again at most $k-1$ selector gadget vertices are mapped to i.

$\Rightarrow$: Suppose $f : W \to \{1, \ldots, D\}$ is a schedule, fulfilling the required properties. First, as the length of the floor path equals the deadline D, it follows that we have $f(a_i) = i$ for all i, $1 \leq i \leq D$. For floor gadget vertices, this implies $f(b_i) = i$.

Call the interval $[n-1+(i-1)c+1, n-1+ic]$ the ith range $(1 \leq i \leq kn)$. We say that the ith range is *polluted* by the jth selector path, when there exist an integer in this range to which no vertex on this jth selector path is mapped, i.e., when there exists an x, $n-1+(i-1)c+1 \leq x \leq n-1+ic$, with $f^{-1}(x) \cap \{c_{j,j'} \mid 1 \leq j' \leq D-n+1\} = \emptyset$. As each selector path has length $D-n+1$, it can pollute only $n-1$ ranges. The total number of polluted ranges hence is at most $kn-k$, so there is at least one range that is not polluted, say the δth range $[n-1+(\delta-1)c+1, n-1+\delta c]$. We now define numbers $\gamma_1, \ldots, \gamma_k$, such that

$$f(c_{i,n-1+(\delta-1)c+1-\gamma_i}) = n-1+(\delta-1)c+1$$

Note that by the discussion above, $\gamma_1, \ldots, \gamma_k$ are uniquely defined. It easily follows that for all selector path vertices $j \leq f(c_{i,j}) \leq j+n-1$. So then, $\{\gamma_1, \ldots, \gamma_k\} \subseteq \{0, \ldots, n-1\}$.

Now, we show that for all q, v_q belongs to the set $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\}$, or is adjacent to a vertex in this set. As shorthand notation, we write $z = n-1+(\delta-1)c+qn$. Look at $X = f^{-1}(z)$. Note that the set X contains one floor vertex, one floor gadget vertex, and k selector path vertices. So, it can contain at most $k-1$ selector gadget vertices. So, there is an l, $1 \leq l \leq k$, such that X does not contain any vertex of the form $d_{l,\epsilon}$. We claim that $d_{l,z-\gamma_l}$ does not exist in $\mathcal{D}$: Note that $f(c_{l,z-\gamma_l-1}) = z-1$, $f(c_{l,z-\gamma_l+1}) = z+1$. So, $d_{l,z-\gamma_l}$ does not exist in $\mathcal{D}$, otherwise it would be mapped to z. As $d_{l,n-1+(\delta-1)c+qn-\gamma_l}$ does not exist in $\mathcal{D}$, we have that $\gamma_l = q$, or $(v_{\gamma_l}, v_q) \in E$. It follows that $\{v_{\gamma_1}, \ldots, v_{\gamma_k}\}$ is a dominating set of G. □

The theorem follows by Claim 2.2, and $W[2]$-hardness of DOMINATING SET. □

We now sketch a key part of the proof of $W[t]$-hardness for all t for COLORED CUTWIDTH with $r = 1$ (further to be called CC-1), and for ICG and TCG. We note in passing that the intermediate step of the reduction of $W[t]$ to LCS involves a blowup of $O(n^t)$. It is not possible in the space allowed for this abstract even to describe all of the constructions for the various stages of combinatorial reduction which prove this theorem. We make here a somewhat arbitrary choice of which part of the proof to present.

Theorem 2.3 *(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 begin from LCS. Let strings $s^1, \ldots, s^K \in \Sigma^*$ and an integer M be given. We denote the length of a string s^k as L_k. We write $R = |\Sigma|$, and $\Sigma = \{\sigma_0, \ldots, \sigma_{R-1}\}$. 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 set of colors C is defined $C = \{c_i \mid 0 \leq i \leq K\} \cup \{d_i \mid 1 \leq i \leq K\} \cup \{e_{i,j} \mid i \in \{0,1,2\},\ 1 \leq j \leq K\} \cup \{f_{i,j} \mid 1 \leq i \leq K,\ 1 \leq j \leq K,\ i \neq j\}$.

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

1. Two anchors. We take four vertices $v_1^1,\ v_2^1,\ v_1^2,\ v_2^2$. For every color $c \in C$, we take 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. Take vertices $\{w_i^m \mid 1 \leq m \leq M,\ 0 \leq i \leq 3R\}$. Take the following edges:

(i) An edge (v_2^1, w_0^1) with color c_0.

(ii) An edge (w_{3R-1}^M, v_1^2) with color c_0.

(iii) For all m, $1 \leq m \leq M$, an edge (w_{3R-1}^m, w_0^{m+1}) with color c_0.

(iv) For all m, $1 \leq m \leq M$, i, $0 \leq i \leq 3R-1$, take an edge (w_i^m, w_{i+1}^m) with color c_0, and for all k, $1 \leq k \leq K$, take an edge (w_i^m, w_{i+1}^m) with color d_k.

(v) For all m, $1 \leq m \leq M$, i, $0 \leq i \leq R-1$, and for all k, $1 \leq k \leq K$, take an edge (w_{3i}^m, w_{3i+1}^m) with color $e_{0,k}$, take an edge (w_{3i+1}^m, w_{3i+2}^m) with color $e_{1,k}$, and take an edge (w_{3i+2}^m, w_{3i+3}^m) with color $e_{2,k}$.

3. String components. Take vertices $\{x_{l,i}^k \mid 1 \leq k \leq K,\ 1 \leq l \leq L_k,\ 0 \leq i \leq 3R+1\}$. For all k, $1 \leq k \leq K$, take the following edges:

(i) Two edges $(v_2^1, x_{1,0}^k)$, one with color c_k, and one with color d_k.

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

(iii) For all l, $1 \leq l \leq L_k - 1$, take two edges $(x_{l,3R+1}^k, x_{l+1}^k, 0)$, one with color c_k and one with color d_k.

For all k, $1 \leq k \leq K$, and all l, $1 \leq l \leq L_k$, take the following edges:

(i) For all i, $0 \leq i \leq 3R$, take an edge $(x_{l,i}^k, x_{l,i+1}^k)$ with color c_k.

(ii) For all r, $0 \leq r \leq R-1$, take an edge $(x_{l,3r}^k, x_{l,3r+1}^l)$ with color $e_{1,k}$, take an edge $(x_{l,3r+1}^k, x_{l,3r+2}^l)$ with color $e_{2,k}$, and take an edge $(x_{l,3r+2}^k, x_{l,3r+3}^l)$ with color $e_{0,k}$.

(iii) Take an edge $(x_{l,3R}^k, x_{l,3R+1}^k)$ with color $e_{1,k}$.

(iv) 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 $k' \neq k$, take an edge $(x_{l,3r+1}^k, x_{3r+2}^k)$ with color $f_{k',k}$. For all $k' \neq k$, take an edge (x_{3i+1}^l, x_{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.

Claim 2.4 *G with coloring c_G has colored cutwidth 1, if and only if $s^1, \ldots, s^M$ have a common subsequence of length M.*

Proof: $\Rightarrow$: Suppose f is a linear ordering of G with colored cutwidth 1. First note that no vertex x can be placed by f between v_1^1 and v_2^1, as any edge adjacent to x would cause a color conflict with one of the edges between v_1^1 and v_2^1. Also, for no edge $(v, w) \in E$, we can have that v is placed left of v_1^1 and w is placed right of v_1^1, as this would also cause a color conflict. Similarly, for v_2^1, and for the other 'anchor' v_1^2 and v_2^2. It follows that all vertices must be placed between $f(v_2^1)$ and $f(v_1^2)$. So, w.l.o.g., we may assume that for all $x \notin A$, $f(v_1^1) < f(v_2^1) < f(x) < f(v_1^2) < f(v_2^2)$.

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 $c \in \{c_0, c_1, \dots, c_K\}$. If $v_2^1, y_1, y_2, \dots, y_p, v_1^2$ 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, $1 \le m \le M$, i, i', $0 \le i < i' \le 3R$, $f(w_i^m) < f(w_{i'}^m)$, and that for all m, m', $1 \le m < m' \le M$, i, i', $0 \le i, i' \le M$, $f(w_i^m) < f(w_{i'}^{m'})$. Also, for all k, $1 \le k \le K$, l, $1 \le l \le L_k$, i, i', $0 \le i < i' \le 3R+1$, $f(x_{l,i}^k) < f(x_{l,i'}^k)$, and for all k, $1 \le k \le K$, l, l', $1 \le l < l' \le L_k$, i, i', $0 \le i, i' \le 3R+1$, $f(x_{l,i}^k) < f(x_{l',i'}^k)$.

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:

(i) $(f(v_2^1), f(w_0^1))$

(ii) $(f(w_{3R}^m), f(w_0^{m+1}))$ for some m, $1 \le m \le M$.

(iii) $(f(w_{3R}^M), f(v_1^2))$.

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, $1 \le l < L_k$, the two vertices $x_{l,3R+1}^k$ and $x_{l+1,0}^k$ must belong to the same interval.

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

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

Consider a fixed k, $1 \le k \le K$. As $f(x_{1,0}^k) < f(w_0^1)$, we have that all $g(k,m) \ge 1$. Note that for each i, $1 \le i \le 3R-2$, m, $1 \le m \le M$, 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 choice 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 either at w_i^m or at w_{i+1}^m (or at both) a color conflict, as for each of these four colors, at least one of these two vertices is adjacent to an edge of that color. Also, for such a vertex $x_{l,j}^k$, we have that $j \ne 0$ and $j \ne 3R+1$. So, we now have that $g(k,1) < g(k,2) < \cdots g(k,M)$.

Consider some fixed k, $1 \le k \le K$, and m, $1 \le m \le M$. For each of the pairs w_{3i+1}^m, w_{3i+2}^m, $(0 \le i \le R-1)$ there must be at least one vertex $x_{g(k,m),j}^k$ with $f(w_{3i+1}^m) < f(x_{g(k,m),j}^k) < f(w_{3i+2}^m)$. As 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, $0 \le i \le R-1$, $f(w_{3i+1}^m) < f(x_{g(k,m),3i+2}^k) < f(w_{3i+2}^m)$. 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', $1 \le k, k' \le K$, m, $1 \le m \le M$, we have that the open intervals $(f(x_{g(k,m),3i+1}^k), f(x_{g(k,m),3i+2}^l))$ and $(f(x_{g(k',m),3i+1}^{k'}), f(x_{g(k',m),3i+2}^{k'}))$ overlap. Suppose that $s_{g(k,m)}^k = \theta_i \ne s_{g(k',m)}^{k'} = \theta_{i'}$. Now, edges $(x_{g(k,m),3i+1}^k, x_{g(k,m),3i+2}^l)$ 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, contradiction. It follows that all 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 equal.

$\Leftarrow$: We omit the simple construction for space reasons. $\Box$

From claim 2.4 and $W[t]$-hardness of the LONGEST COMMON SUBSEQUENCE problem (with number of strings as parameter) for all $t \in \mathbf{N}$, part (i) of the theorem follows.

(ii) Let $G = (V, E)$ be a graph, with an edge coloring $c_G : E \to C$. Take a new color $a \notin C$. We define a graph $H = (V \cup E, F)$ be a bipartite graph, with $F = \{(v, (v, w)) \mid v \in V,\ (v, w) \in E\}$. (So, we can obtain H, by subdividing every edge in G.) We also 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 the new color a, and for all 'edge-vertices' $e \in E$, color e with its old color in G, i.e., take $c_H(e) = c_G(e)$.

The following claim shows that the transformation, that maps (G, c_G) to (H, c_H) is a reduction from CC-1 to ICG, hence proving part (ii) of the theorem. We omit the (simple) proof. (A main idea is that in any properly colored intervalization of H, the intervals of vertices $v \in V$ do not overlap; this yields a linear ordering of the vertices in V, and this linear ordering has the desired property.)

Claim 2.5 *G has colored cutwidth 1, if and only if H is a subgraph of a properly colored interval graph.*

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

Claim 2.6 *The following statements are equivalent.*
(i) $s^1, \ldots, s^K$ have a common subsequence of length M.
(ii) G has colored cutwidth 1.
(iii) H is a subgraph of a properly colored interval graph.
(iv) H is a subgraph of a properly colored chordal graph.

We omit the proof. Now, part (iii) of the theorem follows, as we have a transformation from LCS to TCG. □

3 Concluding Remarks

The results of this paper make a substantial contribution to showing that the generally greater technical demands of combinatorial reductions in parameterized complexity theory are not an insurmountable obstacle to finding concrete classification information for many natural parameterized problems from a wide variety of application domains — although we note that there are many familiar concrete problems that have so far resisted such efforts (see [14, 16] for lists). Our results also give the first natural examples of problems that are hard for $W[t]$ for all t, but that seem unlikely to be hard for $W[P]$, a matter which could be of interest for the study of the structural complexity of the W hierarchy.

References

[1] K. Abrahamson, M. Fellows: Finite automata, bounded treewidth and well-quasiordering. Proceedings of the AMS Summer Workshop on Graph Minors (Seattle, 1991), A.M.S. Contemporary Mathematics Series vol. 147, *Graph Structure Theory*, eds. N. Robertson and P. Seymour (1993), 539–564.

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

[3] S. Arnborg, J. Lagergren, D. Seese: Easy problems for tree-decomposable graphs. J. Algorithms 12, 308–340 (1991)

[4] S. Arnborg, A. Proskurowski: Linear time algorithms for NP-hard problems restricted to partial k-trees. Disc. Appl. Math. 23, 11–24 (1989)

[5] H. L. Bodlaender: Dynamic programming algorithms on graphs with bounded tree-width. Proceedings of the 15'th International Colloquium on Automata, Languages and Programming, pages 105–119. In: Springer Verlag, Lecture Notes in Computer Science volume 317, 1988.

[6] H. L. Bodlaender. The complexity of finding uniform emulations on paths and ring networks. *Information and Computation*, pages 87 – 106, 1990.

[7] H. L. Bodlaender: A linear time algorithm for finding tree-decompositions of small treewidth. Proceedings of the 25th Annual ACM Symposium on Theory of Computing (1993).

[8] H. L. Bodlaender, M. R. Fellows, T. Warnow: Two Strikes Against Perfect Phylogeny. In: W. Kuich (editor), Proceedings of the 19th International Colloquium on Automata, Languages and Programming (ICALP'92). In: Springer-Verlag, Berlin, Lecture Notes in Computer Science, volume 623, pp. 273-283.

[9] R. B. Borie, R. G. Parker, C. A. Tovey: Automatic generation of linear-time algorithms from predicate calculus descriptions of problems on recursively constructed graph families. Algorithmica 7, 555-582 (1992)

[10] L. Cai, J. Chen, R. Downey and M. Fellows: The parameterized complexity of short computations and factorizations. University of Victoria, Technical Report, Department of Computer Science, July, 1993.

[11] B. Courcelle: The monadic second-order logic of graphs I: Recognizable sets of finite graphs. Information and Computation 85, 12–75 (1990)

[12] R. G. Downey, P. A. Evans and M. R. Fellows: Parameterized learning complexity. *Proc. Sixth ACM Workshop on Computational Learning Theory (COLT)*, pp. 51–57, 1993.

[13] R. G. Downey, M. R. Fellows: Fixed parameter tractability and completeness. Congr. Num., 87 161–187 (1992)

[14] R. G. Downey, M. R. Fellows: Fixed parameter tractability and completeness I: basic results. To appear in SIAM J. Comp.

[15] R. G. Downey, M. R. Fellows: Fixed parameter tractability and completeness II: on completeness for $W[1]$. To appear in TCS-A.

[16] R. G. Downey, M. R. Fellows: Fixed parameter intractability. Proceedings of the Seventh Annual IEEE Conference on Structure in Complexity Theory, 36–49 (1992)

[17] R. G. Downey and M. R. Fellows: Fixed parameter tractability and completeness III: some structural aspects of the W-hierarchy. To appear in *Proc. 1992 Dagstuhl Workshop on Structural Complexity Theory* (Cambridge University Press).

[18] R. G. Downey and M. R. Fellows: Parameterized computational feasibility. To appear in *Proc. Second Cornell Workshop on Feasible Mathematics* (Birkhauser, Boston).

[19] R. G. Downey and M. R. Fellows: *Parameterized complexity.* Monograph in preparation.

[20] R. G. Downey, M. R. Fellows, B. M. Kapron, M. T. Hallett and H. T. Wareham: The parameterized complexity of some problems in logic and linguistics. University of Victoria, Technical Report, Department of Computer Science, July, 1993.

[21] M. R. Fellows, M. T. Hallett and H. T. Wareham, DNA physical mapping: three ways difficult. *Proc. First European Symposium on Algorithms*, pp.157–168, 1993.

[22] M. R. Fellows and N. Koblitz: Fixed-parameter complexity and cryptography. *Proceedings of the Tenth International Conference on Algebraic Algorithms and Error-Correcting Codes (AAECC 10)*, Springer-Verlag, Lecture Notes in Computer Science, 1993.

[23] D. Fernández-Baca and A. Medipalli. Parametric module allocation on partial k-trees. *IEEE Trans. on Computers*, 42:738–742, 1993.

[24] M. R. Garey and D. S. Johnson: *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).

[25] M. Golumbic, H. Kaplan, R. Shamir: On the complexity of DNA physical mapping. Technical Report 271/93, Tel Aviv University, January 1993.

[26] E. M. Gurari and I. H. Sudborough. Improved dynamic programming algorithms for bandwidth minimization and the mincut linear arrangement problem. *J. Algorithms*, 5:531–546, 1984.

[27] F. R. McMorris, T. Warnow, T. Wimer: Triangulating vertex colored graphs. In: Proceedings, SODA'92. To appear, SIAM J. Discr. Math.

[28] D. P. Sanders: On linear recognition of tree-width at most four. Manuscript, Mathematics Department, Georgia Tech., November 1992.

[29] J. B. Saxe. Dynamic programming algorithms for recognizing small-bandwidth graphs in polynomial time. *SIAM J. Alg. Disc. Meth.*, 1:363–369, 1980.

[30] T. V. Wimer: Linear algorithms on k-terminal graphs. PhD thesis, Dept. of Computer Science, Clemson University, 1987.