Parameterized Complexity of Bandwidth of Caterpillars and Weighted Path Emulation

Hans L. Bodlaender(✉)

Department of Information and Computing Sciences, Utrecht University, P.O. Box 80.089, 3508 TB Utrecht, The Netherlands
h.l.bodlaender@uu.nl

Abstract. In this paper, we show that Bandwidth is hard for the complexity class $W[t]$ for all $t \in \mathbf{N}$, even for caterpillars with hair length at most three. As intermediate problem, we introduce the Weighted Path Emulation problem: given a vertex-weighted path P_N and integer M, decide if there exists a mapping of the vertices of P_N to a path P_M, such that adjacent vertices are mapped to adjacent or equal vertices, and such that the total weight of the pre-image of a vertex from P_M equals an integer c. We show that Weighted Path Emulation, with c as parameter, is hard for $W[t]$ for all $t \in \mathbf{N}$, and is strongly NP-complete. We also show that Directed Bandwidth is hard for $W[t]$ for all $t \in \mathbf{N}$, for directed acyclic graphs whose underlying undirected graph is a caterpillar.

Keywords: Bandwidth · Parameterized complexity · Weighted path emulation · W-hierarchy · Caterpillars

1 Introduction

The Bandwidth problem is one of the classic problems from algorithmic graph theory. In this problem, we are given an undirected graph $G = (V, E)$ and integer k, and want to find a bijection from V to $\{1, 2, \ldots, n\}$, with $n = |V|$, such that for each edge $\{v, w\} \in E$: $|f(v) - f(w)| \leq k$. The problem was proved to be NP-complete in 1976 by Papadimitriou [21]. Later, several special cases were proven to be NP-complete. In 1986, Monien [19] showed that Bandwidth stays NP-complete when the input is restricted to caterpillars with hair length at most three. A caterpillar is a tree where all vertices of degree at least three are on the same path; the hairs are the paths attached to this central path, and have here at most three vertices.

In this paper, we consider the parameterized complexity of this problem. We consider the standard parameterization, i.e., we ask for the complexity of Bandwidth as a function of n and k. This problem is long known to belong to XP: already in 1980, Saxe [22] showed that Bandwidth can be solved in time $f(k) \cdot n^{k+1}$ for some function f; this was later improved to $f(k) \cdot n^k$ [17].

Ł. Kowalik et al. (Eds.): WG 2021, LNCS 12911, pp. 15–27, 2021.
https://doi.org/10.1007/978-3-030-86838-3_2

In 1994, Bodlaender et al. [5] reported that Bandwidth is $W[t]$-hard for all positive integers t, even when we restrict the input to trees. However, the proof of this fact was so far never published. In the current paper, we give the proof of a somewhat stronger result: Bandwidth is $W[t]$-hard for all positive integers t, even when we restrict the input to caterpillars with maximum hair length three. A sketch of a proof that Bandwidth is $W[t]$-hard for all positive integers t for general graphs appears in the monograph by Downey and Fellows [10]. In recent years, Dregi and Lokshtanov [12] gave a proof that Bandwidth is $W[1]$-hard for trees of pathwidth at most two, and showed that there does not exist an algorithm for Bandwidth on such trees with running time of the form $f(k)n^{o(k)}$ assuming that the Exponential Time Hypothesis holds.

Our proof uses techniques from the NP-hardness proof for Bandwidth on caterpillars by Monien [19]. In particular, one gadget in the proof is identical to a gadget from Moniens proof. Also, the proof is inspired by ideas behind the proof of the result reported in [5], and a proof for $W[t]$-hardness of a scheduling problem for chains of jobs with delays, which was obtained by Bodlaender and van der Wegen [7].

To obtain our main result, we obtain an intermediate result that is also interesting on itself. We consider a variation of the notion of uniform emulation. The notion of emulation was introduced by Fishburn and Finkel [15], to describe the simulation of processor networks on smaller processor networks. An emulation of a graph $G = (V, E)$ on a graph $H = (W, F)$ is a function $f : V \to W$, such that for each edge $\{v, w\} \in E$, $f(v) = f(w)$ or $\{f(v), f(w)\} \in F$, i.e., neighboring vertices are mapped to the same or neighboring vertices. An emulation is uniform when each vertex in H has the same number of vertices mapped to it, i.e., there is a constant c, called the *emulation factor*, such that for all $w \in W$: $|f^{-1}(w)| = c$. An analysis of the complexity to decide whether for given G and H, there exists a uniform emulation was made by Bodlaender and van Leeuwen [8], and Bodlaender [2]. In particular, in [2], the complexity of deciding if there is a uniform emulation on a path or cycle was studied. It was shown that Uniform Emulation on a Path belongs to XP, parameterized by the emulation factor c, belongs to XP for connected graphs and is NP-complete, even for $c = 4$, when we allow that G is not connected. Bodlaender et al. [5] claimed that Uniform Emulation on a Path is hard for $W[t]$ for all positive integers t. In this paper, we show a variation of this result, where the input is a *weighted path*. We name the problem of finding an uniform emulation of a weighted path to a path Weighted Path Emulation. It is straightforward to modify the algorithm from [2] to weighted graphs. This shows that Weighted Path Emulation belongs to XP, with the emulation factor as parameter.

There is a sharp distinction between the complexity of the Bandwidth problem for caterpillars with hairs of length at most two, and caterpillars with hairs of length three (or larger). Assmann et al. [1] give a characterization of the bandwidth for caterpillars whose hair length is at most two, and show that one can compute a layout of optimal width in $O(n \log n)$ time. This contrasts with the NP-hardness and fixed parameter intractability for caterpillars with hairs of length three, by Monien [19] and this paper. For related results, see [18, 20, 23].

Very recently, the results in the paper were strengthened, and it was shown that Bandwidth for caterpillars of hair length at most three, and Weighted Path Emulation are complete for a class of parameterized problems called XNLP. For a brief discussion of these results, see Sect. 6.

This paper is organized as follows. In Sect. 2, we give a number of definitions. In Sect. 3, we discuss hardness for the Weighted Path Emulation problem. Section 4 gives the main result: hardness for the bandwidth of caterpillars with hairs of length at most three. In Sect. 5, we discuss a variation of the proof, to obtain that Directed Bandwidth is hard for $W[t]$ for all positive integers $t \in \mathbf{N}$, for directed acyclic graphs whose underlying undirected graph is a caterpillar with hair length one. Some final remarks are made in Sect. 6. In this extended abstract, we describe the intuition behind the proofs; the precise proofs can be found in the version of the paper on arXiv [3].

2 Definitions

All graphs in this paper are considered to be simple and undirected. We assume that the reader is familiar with standard notions from graph theory and fixed parameter complexity (see e.g. [10,11,16]).

P_n denotes the path graph with n vertices. We denote the vertices of P_n by the first n positive integers, $1, 2, \ldots, n$; the edges of P_n are the pairs $\{i, i+1\}$ for $1 \leq i < n$.

A *caterpillar* is a tree such that there is a path that contains all vertices of degree at least three. A caterpillar can be formed by taking a path P_N (the *spine*), and then attaching to vertices of P_N zero or more paths. These latter paths are called the *hairs* of the caterpillar.

A *linear ordering* of a graph $G = (V, E)$ is a bijective function $f : V \rightarrow \{1, 2, \ldots, n\}$. The *bandwidth* of a linear ordering f of G is $\max_{\{v,w\} \in E} |f(v) - f(w)|$. The *bandwidth* of a graph is the minimum bandwidth over its linear orderings.

Let $G = (V, E)$ be an undirected graph, and $w : V \rightarrow \mathbf{Z}^+$ be a function that assigns to each vertex a positive integer weight. An *emulation* of G on a path P_M is a function $f : V \rightarrow \{1, 2, \ldots, M\}$, such that for all edges $\{v, w\} \in E$, $|f(v) - f(w)| \leq 1$. An emulation $f : V \rightarrow \{1, 2, \ldots, M\}$ is said to be *uniform*, if there is an integer c, such that for all $i \in \{1, 2, \ldots, M\}$, $\sum_{v:f(v)=i} w(v) = c$. c is called the *emulation factor*.

For a directed acyclic graph $G = (V, A)$, the *directed bandwidth* of a topological ordering of G is $\max_{(v,w) \in A} f(w) - f(v)$; the *directed bandwidth* of a directed acyclic graph G is the minimum directed bandwidth over all topological orderings of G.

If we have a directed graph $G = (V, A)$, the *underlying undirected graph* of G is the undirected graph $G' = (V, E)$, with $E = \{\{v, w\} \mid (v, w) \in A\}$; i.e., we forget the direction of edges; if we obtain a pair of parallel edges, we take only one.

We consider the following parameterized problems.

Bandwidth
Given: An undirected graph $G = (V, E)$, integer k.
Parameter: k.
Question: Is the bandwidth of G at most k?

Directed Bandwidth
Given: A directed acyclic graph $G = (V, E)$, integer k.
Parameter: k.
Question: Is the directed bandwidth of G at most k?

Weighted Path Emulation
Given: Integers N, M, c, weight function $w : \{1, 2, \ldots, N\} \to \mathbf{Z}^+$, such that $\sum_{i=1}^{N} w(i)/M = c \in \mathbf{Z}^+$.
Parameter: c.
Question: Is there a uniform emulation f of P_N with weight function w on P_M?

Note that in the problem statement above, c is the emulation factor, i.e., we have for a solution f that for each j, $1 \leq j \leq M$, $\sum_{i:f(i)=j} w(i) = c$.

A Boolean formula is said to be *t-normalized*, if it is the conjunction of the disjunction of the conjunction of ... of literals, with t alternations of AND's and OR's. So, a Boolean in Conjunctive Normal Form is 2-normalized. Downey and Fellows [9] consider the following parameterized problem; this is the starting point for our reductions.

Weighted t-Normalized Satisfiability
Given: A t-normalized Boolean formula F and a positive integer $k \in \mathbf{Z}^+$.
Parameter: k
Question: Can F be satisfied by setting exactly k variables to true?

Theorem 1 (Downey and Fellows [9]). Weighted t-Normalized Satisfiability *is $W[t]$-complete.*

3 Hardness of Weighted Path Emulation

Our first main result is the following; the proof can be found in the full paper [3].

Theorem 2. Weighted Path Emulation *is $W[t]$-hard for all positive integers t.*

Suppose we are given a t-normalized Boolean expression F over n variables, say $x_1, \ldots, x_n$, and integer k. We let t' be the number of nested levels of disjunction. We consider the problem to satisfy F by making exactly k variables true.

We will define a path P_N with a weight function $w : \{1, \ldots, N\} \to \mathbf{Z}^+$, an emulation factor c, and an integer M, such that P_N has a uniform emulation on a path P_M if and only if F can be satisfied by setting exactly k variables to true. Before giving the proof, we give a high level overview of some main ideas of the proof.

3.1 Intuition and Techniques

In this subsection, we give some ideas behind the construction. The precise construction and formal proofs are given in the next subsection.

We assume we have given a t-normalized Boolean formula F. We transform the formula to a weighted path P_N, such that P_N has a uniform emulation on P_M with c the emulation factor, if and only if F can be satisfied by setting exactly k variables to true.

We can view F as a tree, with internal nodes marked with disjunction or conjunction, and each leaf with a literal, then we alternatingly have a level in the tree with disjunctions, and with conjunctions. We set t' to be the number of levels with disjunctions.

The path P_N is formed by taking, in this order, the following: a part called the 'floor', k 'variable parts', t' 'disjunction parts', and a 'filler path'. t' is the number of levels in the formula tree with disjunctions, and for each 'level' of disjunction we have one disjunction part. E.g., if F is in conjunctive normal form, then $t' = 1$.

The *floor* has M vertices, each with a weight that is larger than $c/2$. Thus, we cannot map two floor vertices to the same element of P_M, and thus, can assume, without loss of generality, that the ith floor vertex is mapped to i. The different weights for floor vertices help to build the further gadgetry of the construction.

The variable and disjunction parts are forced to start at M, then move to 1, and then move (possibly with some 'zigzagging') back to M, where then the next part starts. This is done by giving each part one vertex of large weight that only can fit at vertex 1, and another vertex of even larger weight, that only can fit at vertex M. These large weight vertices are called left and right turning points. See Fig. 1 for an illustration of the construction.

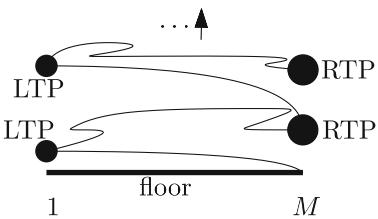

Fig. 1. Impression of a first part of the construction. The ith floor vertex is mapped to i; after this, the path then moves from M to 1 and back, with left turning points (LTR) mapped to 1, and right turning points (RTP) mapped to M. The picture shows the floor and first two variable parts.

We have k *variable parts*. Each models one variable that is set to true. We start with a left turning point, $M - 2$ vertices of weight one, and a right turning point: this is to move back from M to 1. Then, we have $n - 1$ vertices of weight one, $M - 2n - 2$ 'heavy' vertices, and again $n - 1$ vertices of weight one.

The heavy vertices are mapped consecutively (except possibly at the first n and last n positions); the weight one vertices before and after the heavy vertices allows us to shift the sequence of heavy vertices in n ways—each different such shift sets another variable to true. By using two different heavy weights, combined with weight settings for floor vertices and vertices from disjunction parts, we can check that all variable parts select a different variable to be true (which is done at positions $n+2, \ldots, 2n+1$), and that F is satisfied (which is done at positions $2n+2, \ldots, M-n-2$).

We have for each level in the formula tree with disjunctions a *disjunction part*. Thus, we have t' disjunction parts. With help of 'anchors' (vertices of large weight that can go only to one specific position), we can ensure that a subpart for a disjunction has to be mapped to the part of the floor that corresponds with this disjunction. Such a subpart consist of a path with weight one vertices, a path with $3m(F')$ *selecting vertices* (which have larger weight), and another path with weight one vertices. Now, each term in the disjunction has an associated interval of size $m(F')$ and between these intervals we have $m(F')$ elements. Then, we can show that the selecting vertices must cover entirely one of the intervals of a term—this corresponds to that term being satisfied. See Fig. 2 below. Say $F' = F_1 \vee F_2 \vee F_3$. In the illustration we see the intervals assigned to F_1, F_2, and F_3, and the space between, before and after these. Each of the seven intervals has size $m(F')$. We can show that the $3m(F')$ selecting vertices must be mapped to consecutive vertices between the left and right anchor of F', and thus these cover the interval of each least one F_i entirely.

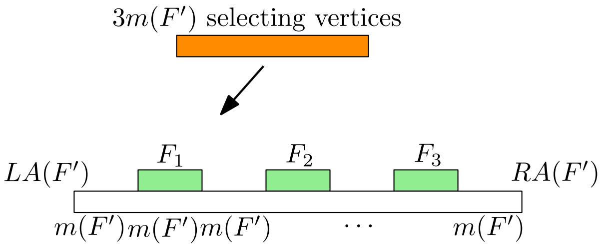

Fig. 2. Illustration: consecutive selecting vertices cover the interval of one term

Heavy vertices of variable parts come in two weights: c^v and $c^c + c^u$. This is used for checking that F is satisfied. As an example, consider a negative literal $\neg x_j$ in F. We have one specific position on P_M, say i, that checks whether this literal is satisfied, in case its satisfaction contributes to the satisfaction of F—that case corresponds to having a selecting vertex mapped to i for each level of disjunction. Now, the weight of the floor vertex mapped to i is such that when this floor vertex and all selecting vertices are mapped to i, then we can only fit k heavy vertices of weight c^v here; if at least one of these heavy vertices has weight $c^v + c^u$, then the total weight mapped to i exceeds c. If this happens, then this heavy vertex belongs to a variable part which corresponds to setting x_i to be true; thus, this enforces that x_i is false. A somewhat similar construction is used for positive literals.

The last part of P_N is the *filler path*. This is a long path with vertices of weight one. This is used to ensure that the mapping becomes uniform: if the total weight of vertices of floor, variable part, and disjunction parts vertices mapped to i is z_i, then we map $c - z_i$ (consecutive) vertices of the filler path to i. See Fig. 3 for an illustration.

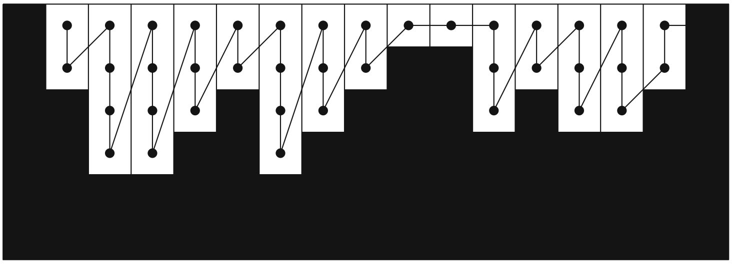

Fig. 3. Illustration of the mapping of the filler path. The black area represents the weights of floor, variable part and disjunction part vertices mapped to the element of P_M

We need in the proof for Bandwidth actually a slightly different result (for an easier proof), namely, we require that the first vertex of P_N is mapped to M. From the proof of Theorem 2 we also can conclude the next result.

Corollary 1. Weighted Path Emulation with $f(1) = 1$ *and* Weighted Path Emulation with $f(1) = M$ *are $W[t]$-hard for all positive integers t.*

We also have the following corollary; see also [3].

Corollary 2. Weighted Path Emulation *is strongly NP-complete.*

4 Hardness of Bandwidth of Caterpillars

The main result of this section is the following theorem. Many details of the proof can be found in [3].

Theorem 3. Bandwidth *is $W[t]$-hard for all positive integers t, when restricted to caterpillars with hair length at most three.*

To prove Theorem 3, we transform from the Weighted Path Emulation with $f(1) = M$ problem.

Let P_N and P_M be paths with weight function $w : \{1, \ldots, N\} \to \{1, \ldots, c\}$, and $c = \sum_{i=1}^{N} w(i)/M$ the emulation factor. Thus, $\sum_{i=1}^{N} w(i) = cM$. Recall that we parameterized this problem by the value c.

Assume that $c > 6$; otherwise, obtain an equivalent instance by multiplying all weights by 7.

Let $b = 12c + 6$. Let $k = 9bc + b$. Note that k is even. We give a caterpillar $G = (V, E)$ with hair length at most three, with the property that P_N has a uniform emulation on P_M, if and only if G has bandwidth at most k.

G is constructed in the following way:

- We have a *left barrier*: a vertex p_0 which has $2k - 1$ hairs of length one, and is neighbor to p_1.
- We have a path with $5M - 3$ vertices, $p_1, \ldots, p_{5M-3}$. As written above, p_1 is adjacent to p_0. Each vertex of the form $5i - 2$ or $5i$ $(1 \leq i \leq M - 1)$ receives $2k - 2b$ hairs of length one. See Fig. 4. We call this part the *floor*.
- Adjacent to vertex p_{5M-3}, we add the *turning point* from the proof of Monien [19]. We have vertices $v_a = p_{5M-3}$, v_b, v_c, v_d, v_e, v_f, v_g, which are successive vertices on a path. I.e., we identify one vertex of the turning point (v_a) with the last vertex of the floor p_{5M-3}. To v_c, we add $\frac{3}{2}(k - 2)$ hairs of length one; to v_d, we add k hairs of length three, and to v_f we add $\frac{3}{2}(k - 2)$ hairs of length one. Note that this construction is identical to the one by Monien [19]; vertex names are chosen to facilitate comparison with Moniens proof. See Fig. 5.
- Add a path with $6N - 5$ vertices, say $y_1, \ldots y_{6n-5}$, with y_1 adjacent to v_g. To each vertex of the form y_{6i-5}, add $9b \cdot w(i)$ hairs of length one. We call this part the *weighted path gadget*.
- Note that the number of vertices that we defined so far and that is not part of the turning point equals $2k+5M-3+2(M-1)(2k-2b)+6N-5+9b\sum_{i=1}^{N} w_i = 5M + 4Mk - 2k - 4Mb + 4b + 9bcM$. Let this number be α. One easily sees that $\alpha \leq (5M - 2)k - 1$. Add a path with $(5M - 2)k - 1 - \alpha$ vertices and make it adjacent to y_{6n}. We call this the *filler* path.

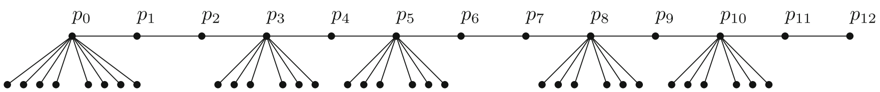

Fig. 4. First part of the caterpillar

Let G be the remaining graph. Clearly, G is a caterpillar with hair length at most three. It is interesting to note that the hair lengths larger than one are only used for the turning point.

The correctness of the construction follows from the following lemma. The proof can be found in the full paper [3].

Lemma 1. *P_N has a uniform emulation g on P_M with emulation factor c with $f(1) = M$, if and only if the bandwidth of G is at most k.*

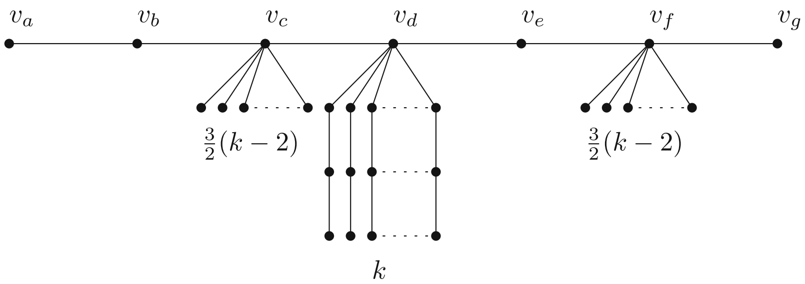

Fig. 5. The Turning Point, after Monien [19]

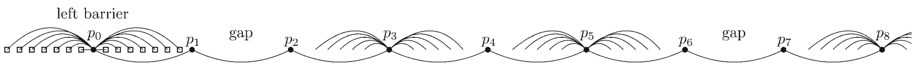

Fig. 6. Initial part of the floor, with left barrier and two gaps

Some Intuition. We sketch some ideas behind the proof. Suppose we have a linear ordering g of G with bandwidth at most k. We have a number of observations.

- p_0 with hairs form a blockage (called the *left barrier*, in the sense that these must either entirely at the left side or entirely at the right side of the linear ordering.
- In the same way, the turning point forms a blockage; the proof of this is due to Monien [19]. Without loss of generality, we can assume p_0 is at the left side, the turning point is at the right side.
- By considering the total number of vertices, we can show that the successive vertices p_i, p_{i+1} always have distance $k-1$ or k, with $g(p_{i+1}) = g(p_i) + k$ or $g(p_{i+1}) = g(p_i) + k - 1$.
- Vertices p_3, p_5, p_8 have 'many' hairs: these fills most of the nearby positions. E.g., in intervals $[g(p_2), g(p_3)]$, $[g(p_3), g(p_4)]$, $[g(p_4), g(p_5)]$, and $[g(p_5), g(p_6)]$ we have many hairs of the vertices p_i, while the interval $[g(p_6), g(p_7)]$ has not. So, every fifth interval has 'more space', which we call a *gap*. See Fig. 6 for the initial part of the floor with two gaps.
- Vertices of the form y_{6i-5} also have a large number of hairs. We must have that most of these hairs must be mapped to intervals of the form $[g(p_{5j+1}), g(p_{5j+2})]$. In such a case, map the ith vertex of P_N to the jth vertex of P_M. Let f be the resulting mapping
- An interval of the form $[g(p_{5j+1}), g(p_{5j+2})]$ (and the neighboring intervals, after taking hairs of the floor into account) cannot fit $9b(c+1)$ hairs of the weighted path gadget. This implies that the total weight of all vertices mapped by f to j is bounded by c; and, as we have M such intervals, must be exactly c. This shows uniformity of the mapping f.
- As discussed, a vertex of the form y_{6i-5} has hairs mapped to an interval $[g(p_{5j+1}), g(p_{5j+2})]$. Thus, when we map i to j, y_{6i-5} is mapped to an integer between $g(p_{5j})$ and $g(p_{5j+3})$. Then, y_{6i+1} is mapped to an integer between

$g(p_{5j-6})$ and $g(p_{5j+9})$—as there is a path with six edges from y_{6i-5} to y_{6i+1}, they can be mapped at most six intervals apart. This shows that there are hairs of y_{6i+1} that are mapped to the interval $[g(p_{5j-5}), g(p_{5j-4})]$ or to the interval $[(g(p_{5j+1}), g(p_{5j+2})]$ or to the interval $[g(p_{5j+7}), g(p_{5j+8})]$. And thus, vertex $i+1$ from P_N is mapped to $j-1$, j or $j+1$. This shows that the mapping f is an emulation.

An illustration of the construction of a linear ordering, given a uniform emulation is given in Fig. 7.

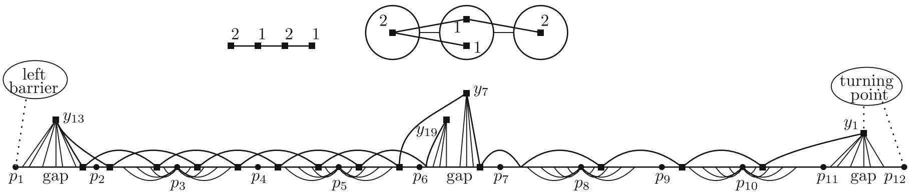

Fig. 7. Illustration of part of the construction. Shown are P_4 with successive vertex weights 2, 1, 2, 1; a uniform emulation on P_3 with emulation factor 2; a layout of a part of G.

As the construction of the caterpillar G can be done in polynomial time, given M, N and w, the main result of this section now follows.

Theorem 4. Bandwidth *for caterpillars with hair length at most three is* $W[t]$*-hard for all* $t \in \mathbf{N}$.

5 Directed Bandwidth

A minor variation of the proof of Theorem 3 gives the following result. The details can be found in the full paper [3].

Theorem 5. Directed Bandwidth *is hard for* $W[t]$ *for all positive integers* t*, when restricted to directed acyclic graphs whose underlying undirected graph is a caterpillar with hair length at most one.*

6 Conclusions

In this paper, we showed that Bandwidth is hard for the complexity class $W[t]$ for all positive integers $t \in N$, even when the input graph is a caterpillar with hairs of length at most three. The proof uses some techniques and gadgets from the NP-completeness proof of Bandwidth for this class of graphs by Monien [19]. Monien also shows NP-completeness of Bandwidth for caterpillars of maximum degree three (with arbitrary hair length). This raises the question whether Bandwidth for caterpillars with maximum degree three is $W[t]$-hard for all t. We conjecture that this is the case; perhaps with a modification of our proof such a result can be achieved?

The intermediate result of the $W[t]$-hardness of Weighted Path Emulation is of independent interest. We used this result as a stepping stone for our main result, but expect that the result may also be useful for proving hardness for other problems as well.

It is unlikely that Bandwidth belongs to $W[P]$. In [14], Fellows and Rosamond describe an argument, due to Hallett, that gives the intuition behind the conjecture that Bandwidth does not belong to $W[P]$. From the works of Bodlaender et al. [4] and Drucker [13], it follows that problems that are AND-compositional do not have a polynomial kernel unless $NP \subseteq coNP/poly$. The intuition behind this methodology is that such a polynomial kernel for an AND-compositional problem would give an unlikely strong compression of information. While Bandwidth is not in FPT, assuming $W[t] \not\subseteq FPT$, for some t, and thus has no kernel (of any size), it is AND-compositional. If Bandwidth would belong to $W[P]$, it would have a certificate of $O(k \log n)$ bits (namely, the indices of the variables that are set to true), and it is unlikely that an AND-compositional problem has such a small certificate. We thus can formulate the following conjecture, due to Hallett.

Conjecture 1 (Hallett, see also [14]*).* Bandwidth does not belong to $W[P]$, unless $NP \subseteq coNP/poly$.

Very recently, the author showed with Groenland, Nederlof and Swennenhuis [6] that Weighted Path Emulation is complete for the class of problems that can be solved with a non-deterministic algorithm that uses $f(k)n^c$ time and $f(k) \log n$ space (f a computable function, c a constant). This class is known as $N[fpoly, f \log]$ and denoted as $XNLP$ in [6]. From the observation that the transformation described in Sect. 4 can be carried out in logarithmic space, it follows that Bandwidth for caterpillars with hair length at most three is also XNLP-complete. We thus also have that Bandwidth does not belong to $W[P]$ unless $W[P] \subseteq XNLP$.

Finally, we conjecture that with modifications of the techniques from this paper, it is possible to show for more problems hardness for the $W[t]$-classes.

Acknowledgements. I thank Michael Fellows and Michael Hallett for discussions and earlier joint work on the parameterized complexity of Bandwidth, as reported in [5]. Some techniques in this paper are based upon techniques, underlying the results reported in [5]. I thank Marieke van der Wegen for discussions.

References

1. Assmann, S.F., Peck, G.W., Sysło, M.M., Zak, J.: The bandwidth of caterpillars with hairs of length 1 and 2. SIAM J. Algebraic Discrete Methods **2**, 387–392 (1981)

2. Bodlaender, H.L.: The complexity of finding uniform emulations on paths and ring networks. Inf. Comput. **86**(1), 87–106 (1990)
3. Bodlaender, H.L.: Parameterized complexity of bandwidth of caterpillars and weighted path emulation. arXiv:2012.01226 (2020)
4. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hermelin, D.: On problems without polynomial kernels. J. Comput. Syst. Sci. **75**(8), 423–434 (2009)
5. Bodlaender, H.L., Fellows, M.R., Hallett, M.: Beyond NP-completeness for problems of bounded width: hardness for the W hierarchy. In: Proceedings of the 26th Annual Symposium on Theory of Computing, STOC 1994, pp. 449–458, New York. ACM Press (1994)
6. Bodlaender, H.L., Groenland, C., Nederlof, J., Swennenhuis, C.M.F.: Parameterized problems complete for nondeterministic FPT time and logarithmic space. arXiv:2105.14882 (2021)
7. Bodlaender, H.L., van der Wegen, M.: Parameterized complexity of scheduling chains of jobs with delays. In: Cao, Y., Pilipczuk, M. (eds.) 15th International Symposium on Parameterized and Exact Computation, IPEC 2020, volume 180 of LIPIcs, pp. 4:1–4:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020)
8. Bodlaender, H.L., van Leeuwen, J.: Simulation of large networks on smaller networks. Inf. Control **71**(3), 143–180 (1986)
9. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness I: basic results. SIAM J. Comput. **24**, 873–921 (1995)
10. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, New York (1999). https://doi.org/10.1007/978-1-4612-0515-9
11. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, London (2013). https://doi.org/10.1007/978-1-4471-5559-1
12. Dregi, M.S., Lokshtanov, D.: Parameterized complexity of bandwidth on trees. In: Esparza, J., Fraigniaud, P., Husfeldt, T., Koutsoupias, E. (eds.) ICALP 2014. LNCS, vol. 8572, pp. 405–416. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43948-7_34
13. Drucker, A.: New limits to classical and quantum instance compression. SIAM J. Comput. **44**(5), 1443–1479 (2015)
14. Fellows, M.R., Rosamond, F.A.: Collaborating with Hans: some remaining wonderments. In: Fomin, F.V., Kratsch, S., van Leeuwen, E.J. (eds.) Treewidth, Kernels, and Algorithms. LNCS, vol. 12160, pp. 7–17. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-42071-0_2
15. Fishburn, J.P., Finkel, R.A.: Quotient networks. IEEE Trans. Comput. **C-31**, 288–295 (1982)
16. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. Springer, Heidelberg (2006). https://doi.org/10.1007/3-540-29953-X
17. Gurari, E.M., Sudborough, I.H.: Improved dynamic programming algorithms for bandwidth minimization and the MinCut linear arrangement problem. J. Algorithms **5**, 531–546 (1984)
18. Hung, L.T.Q., Sysło, M.M., Weaver, M.L., West, D.B.: Bandwidth and density for block graphs. Discret. Math. **189**(1–3), 163–176 (1998)
19. Monien, B.: The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete. SIAM J. Algebraic Discrete Methods **7**, 505–512 (1986)
20. Muradian, D.: The bandwidth minimization problem for cyclic caterpillars with hair length 1 is NP-complete. Theor. Comput. Sci. **307**(3), 567–572 (2003)

21. Papadimitriou, C.H.: The NP-completeness of the bandwidth minimization problem. Computing **16**, 263–270 (1976)
22. Saxe, J.B.: Dynamic programming algorithms for recognizing small-bandwidth graphs in polynomial time. SIAM J. Algebraic and Discrete Methods **1**, 363–369 (1980)
23. Sysło, M.M., Zak, J.: The Bandwidth problem: critical subgraphs and the solution for caterpillars. Ann. Discrete Math. **16**, 281–286 (1982)