Parameterized Complexity of Bandwidth on Trees

Markus Dregi* Daniel Lokshantov*

April 30, 2014

Abstract

The bandwidth of a n-vertex graph G is the smallest integer b such that there exists a bijective function $f : V(G) \to \{1, ..., n\}$, called a layout of G, such that for every edge $uv \in E(G)$, $|f(u) - f(v)| \le b$. In the Bandwidth problem we are given as input a graph G and integer b, and asked whether the bandwidth of G is at most b. We present two results concerning the parameterized complexity of the Bandwidth problem on trees.

First we show that an algorithm for Bandwidth with running time $f(b)n^{o(b)}$ would violate the Exponential Time Hypothesis, even if the input graphs are restricted to be trees of pathwidth at most two. Our lower bound shows that the classical $2^{O(b)}n^{b+1}$ time algorithm by Saxe [SIAM Journal on Algebraic and Discrete Methods, 1980] is essentially optimal.

Our second result is a polynomial time algorithm that given a tree T and integer b, either correctly concludes that the bandwidth of T is more than b or finds a layout of T of bandwidth at most $b^{O(b)}$. This is the first parameterized approximation algorithm for the bandwidth of trees.

1 Introduction

A layout for a graph G is a bijective function $\alpha : V(G) \to \{1, \dots, |V(G)|\}$, and the bandwidth of the layout α is the maximum over all edges $uv \in E(G)$ of $|\alpha(u) - \alpha(v)| \le b$. The bandwidth of G is the smallest integer b such that G has a layout of bandwidth b. In the Bandwidth problem we are given as input a graph G and an integer b and the goal is to determine whether the bandwidth of G is at most b. In the optimization variant we are given G and the task is to find a layout with smallest possible bandwidth.

The problem arises in sparse matrix computations, where given an $n \times n$ matrix A and an integer k, the goal is to decide whether there is a permutation matrix P such that PAP^T is a matrix whose all non-zero entries lie within the k diagonals on either side of the main diagonal. Standard matrix operations such as inversion and multiplication as well as Gaussian elimination can be sped up considerably if the input matrix A can be transformed into a matrix PAP^T of small bandwidth [15].

Bandwidth is one of the most well-studied NP-complete [14, 26] problems. The problem remains NP-complete even on very restricted subclasses of trees, such as caterpillars of hair length at most 3 [24]. Furthermore, it is NP-hard to approximate the bandwidth within any constant factor, even on trees [9]. The best approximation algorithm for Bandwidth on general graphs is by Dungan and Vempala [10], this algorithm has approximation ratio $(\log n)^3$. For trees Gupta [18] gave a slightly better approximation algorithm with ratio $(\log n)^{9/4}$, while for caterpillars a $O(\frac{\log n}{\log \log n})$-approximation [12] can be achieved.

One could argue that the Bandwidth problem is most interesting when the bandwidth of the graph is very small compared to the size of the graph. Indeed, when the bandwidth of G is

[*] Department of Informatics, University of Bergen, Norway

constant the matrix operations discussed above can be implemented in linear time. For each $b \geq 1$ it is possible to recognize the graphs with bandwidth at most b in time $2^{\mathcal{O}(b)}n^{b+1}$ using the classical algorithm of Saxe [27]. At this point it is very natural to ask how much Saxe's algorithm can be improved. Our first main result is that assuming the Exponential Time Hypothesis of Impagliazzo, Paturi and Zane [21], no sigificant improvement is possible, even on very restricted subclasses of trees. In particular we show the following theorem.

Theorem 1. *Assuming the Exponential Time Hypothesis there is no $f(b)n^{o(b)}$ time algorithm for* BANDWIDTH *of trees of pathwidth at most* 2*.*

The proof of Theorem 1 also implies that BANDWIDTH is $W[1]$-hard on trees of pathwidth at most 2 (see [8, 13, 25] for an introduction to parameterized complexity).

As a counterweight to the bad news of Theorem 1 we give the first approximation algorithm for BANDWIDTH of trees whose approximation ratio depends only on the bandwidth b, and not on the size of the graph. Specifically we give a polynomial time algorithm that given as input a tree T and integer b either correctly concludes that the bandwidth of T is greater than b or outputs a layout of width at most $b^{O(b)}$. A key subroutine of our algorithm for trees is an approximation algorithm for the bandwidth of caterpillars with ratio $O(b^3)$. Our algorithm for trees outperforms the $(\log n)^{9/4}$-approximation algorithm of Gupta [18] whenever $b = o(\frac{\log \log n}{\log \log \log n})$. Our algorithm is the first *parameterized approximation* algorithm for the BANDIWTH problem on trees, that is an algorithm with approximation ratio $g(b)$ and running time $f(b)n^{O(1)}$. A parameterized approximation algorithm for the closely related TOPOLOGICAL BANDWIDTH problem has been known for awhile [23], while the existence of a parameterized approximation algorithm for BANDWIDTH, even on trees was unknown prior to this work.

An interestng aspect of our approximation algorithm is the way we lower bound the bandwidth of the input tree T. It is well known that the bandwidth of a graph G is lower bounded by its *pathwidth*, and by its *local density*[1]. One might wonder how far these lower bounds could be from the true bandwidth of G. It was conjectured that the answer to this question is "not too far", in particular that any graph with pathwidth c_1 and local density c_2 would have bandwidth at most c_3 where c_3 is a constant depending only on c_1 and c_2. Chung and Seymour [6] gave a counterexample to this conjecture by constructing a special kind of trees, called *cantor combs*, with pathwidth 2, local density at most 10, and bandwidth approximately $\frac{\log n}{\log \log n}$. Our approximation algorithm essentially shows that the only structures driving up the bandwidth of a tree are pathwidth, local density and cantor comb-like subgraphs.

Related Work. There is a vast literature on the BANDWIDTH problem. For example the problem has been extensively studied from the perspective of approximation algoritms [9, 10, 11, 12, 18], parameterized complexity [2, 16, 27], polynomial time algorithms on restricted classes of graphs [1, 20, 22, 29], and graph theory [4, 6]. We focus here on the study of algorithms for BANDWIDTH for small values of b.

Following the $2^{O(b)}n^{b+1}$ time algorithm of Saxe [27], published in 1980, there was no progress on algorithms for the recognition of graphs of constant bandwidth. With the advent of parameterized complexity in the late 80's and early 90's [8] it became an intriguing open problem whether one could improve the algorithm of Saxe to remove the dependency on b in the exponent of n, and obtain a $f(b)n^{O(1)}$ time algorithm.

In a seminal paper from 1994, Bodlaender, Fellows, and Hallet [2] proved that a number of layout problems do not admit fixed parameter tractable algorithms unless $\mathsf{FPT} = \mathsf{W[t]}$ for every $t \geq 1$, a collapse considered by many to be almost as unlikely as $\mathsf{P} = \mathsf{NP}$. In the same paper Bodlaender, Fellows, and Hallet [2] claim that their techniques can be used to show that a $f(b)n^{O(1)}$ time algorithm for BANDWIDTH would also imply $\mathsf{FPT} = \mathsf{W[t]}$ for every $t \geq 1$.

[1] A definition of these notions can be found in the preliminaries

Downey and Fellows ([8], page 468) further claim that the techniques of [2] imply that even fixed parameter algorithm for Bandwidth *on trees* would yield the same collapse. Unfortunately a full version of [2] substantiating these claims is yet to appear.

2 Preliminaries

All graphs in this paper are undirected and unweighted. For a graph G, we will use the notation $V(G)$ and $E(G)$ for the vertex set and edge set respectively. Or just V and E whenever the graph is clear from the context. The degree of a vertex v is denoted by $\deg(v)$ and the maximum degree in a graph by $\deg(G)$. By $\operatorname{diam}(G)$ we will mean the diameter of a graph G. A *clique* of size n, denoted K_n is a graph where every pair of vertices are connected by an edge. We will use the notation P_l to describe a path of length l and $\hat{P}_l$ for a specific instance of P_l. When we need to index paths this will be done by superscript, i.e. P^i. For two graphs G and H, we say that H is a *subgraph* of G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. Furthermore, we say that H is an induced subgraph of G if $V(H) \subseteq V(G)$ and $E(H) = E(G) \cap V(H)^2$. An induced subgraph of G whose vertices are X is denoted by $G[X]$. When removing a set of vertices X from a graph G, we will use the notation $G - X$ for the graph $G[V(G) \setminus X]$. And furthermore, if we are removing a single vertex v we will write this as $G - v$, and this is short for $G - \{v\}$.

If a function f is defined on a set X and $Y \subseteq X$ we will use the notation $f(Y)$ for $\cup_{y \in Y} f(y)$. When it is clear from the context that we are referring to a vertex set of a graph, we will refer to just the graph. Furthermore, when a function f is defined on the vertex set of a graph, we will sometimes use the sloppy notation $f(G)$ instead of $f(V(G))$.

For intervals of natural numbers we will use the notation $[n]$ for the interval $[1, \ldots, n]$. A k-coloring of a graph G is a function from $V(G)$ to $[k]$ such that two adjacent vertices are given different values. The chromatic number of G, denoted $\chi(G)$ is the minimum k such that there is a k-coloring of G.

Graph Classes

A tree is a connected graph without any cycles. A caterpillar is a tree T with a path B as a subgraph, such that all vertices of degree 3 or more lie on B. We then say that B is a backbone of T and every connected component of $T - B$ is a stray or a hair. We say that a caterpillar is of stray length s if there exists a backbone such that all strays are of size at most s. An interval graph is a graph such that there exists a function from $V(G)$ into intervals of $\mathbb{N}$ such that the images of two vertices have a non-empty intersection if and only if the two vertices are adjacent.

Decompositions

A *tree decomposition* $\mathcal{T}$ of a graph G is a pair (T, X) with $T = (I, M)$ being a tree and $X = \{X_i \mid i \in I\}$ a collection of subsets of V such that:

1. $\bigcup_{i \in I} X_i = V$,
2. for every edge uv there is a bag X_i such that both u and v are contained in X_i and
3. for every vertex $v \in V$ the set $\{i \in I \mid v \in X_i\}$ induces a tree in T.

The *treewidth* of a tree decomposition $\mathcal{T}$, denoted $\operatorname{tw}(G, \mathcal{T}) = \max_{i \in I} |X_i| - 1$ and the treewidth of a graph G is defined as $\operatorname{tw}(G) = \min\{\operatorname{tw}(G, \mathcal{T}) \mid \mathcal{T}$ is a tree decomposition of $G\}$. A *path decomposition* $\mathcal{P}$ of a graph is a tree decomposition such that T is a path. And the *pathwidth* of a graph G, denoted $\operatorname{pw}(G)$ is the minimum width over all path decompositions.

Orderings and Bandwidth

A *linear ordering* or *layout* α of a set S is a bijection between S and $[|S|]$. Given a graph $G = (V, E)$ and a linear ordering α over V, the *bandwidth* of α denoted $\mathrm{bw}(G, \alpha) = \max_{uv \in E} |\alpha(u) - \alpha(v)|$. And furthermore, the bandwidth of G denoted $\mathrm{bw}(G) = \min\{\mathrm{bw}(G, \alpha) \mid \alpha \text{ is a linear ordering over } V\}$. We say that α is a *k-bandwidth ordering* of a graph G if $\mathrm{bw}(G, \alpha) \leq k$. And we say that a bandwidth ordering α of G is optimal if $\mathrm{bw}(G, \alpha) = \mathrm{bw}(G)$.

Let u and v be a pair of vertices of a graph G and α an ordering of $V(G)$. We then say that u is left of v in α if $\alpha(u) < \alpha(v)$ and that u is right of v if $\alpha(v) < \alpha(u)$. A sparse ordering β of a graph G is an injective function from $V(G)$ to $\mathbb{Z}$. And the bandwidth of a sparse ordering β of G, denoted $\mathrm{bw}(G, \beta) = \max_{uv \in E} |\beta(u) - \beta(v)|$. We say that a linear ordering α of G is a compression of a sparse ordering β of G if for every pair of vertices u, v in G it holds that $\beta(u) < \beta(v)$ if and only if $\alpha(u) < \alpha(v)$.

Definition 1. *For a graph G we define the local density of G as*

$$D(G) = \max_{G' \subseteq G} \frac{|V(G')| - 1}{diam(G')}.$$

The following proposition will be used repeatedly in our arguments.

Proposition 1 (Folklore)**.** *For every graph G it holds that $D(G) \leq \mathrm{bw}(G)$ and $\mathrm{pw}(G) \leq \mathrm{bw}(G)$.*

For a graph T, an integer b and a b-bandwidth ordering α we provide the following definitions. Given a set of vertices $Y \subseteq V(T)$ we define the *inclusion interval of* Y, denoted $I(Y)$ as $[\min \alpha(Y), \max \alpha(Y)]$ and for two vertices u and v we define $I(u, v)$ as $I(\{u, v\})$ or equivalently $[\min\{\alpha(u), \alpha(v)\}, \max\{\alpha(u), \alpha(v)\}]$. Given a subgraph H of T we define $I(H)$ as $I(V(H))$. Whenever necessary, we will use subscript to avoid confusion about which ordering is considered.

Problems

We will differentiate the parametrized version of a problem (parameterized by the natural parameter) from the classical one by putting a p in front of the name, i.e. p-Bandwidth is the parameterized version of Bandwidth. We will face two other problems in this paper. The first one is Clique, where given a graph G and an integer k, one is asked whether there is a clique of size k in G. The second one is Even Clique, which is an instance of Clique where you are promised that k is an even number. Both of the problems will be discussed in their parametrized form.

3 Lower Bounds

In this section we will give a reduction from p-Even Clique to p-Bandwidth with a linear blowup of the parameter. For the rest of this section we will refer to the parameter of the instance of p-Even Clique as k and the parameter of the resulting p-Bandwidth instance as $b = 4k + 16$. Before we continue, we introduce some definitions we will use throughout the section. For a subpath $\hat{P}_l = \{v_1, \ldots, v_l\}$ of a graph T we say that $\hat{P}_l$ is *stretched* with respect to a b-bandwidth ordering α if $|\alpha(v_{i+1}) - \alpha(v_i)| = b$ for every $i \in [1, l)$. Observe that as α is injective, stretched implies either $\alpha(v_1) < \alpha(v_2) < \cdots < \alpha(v_l)$ or $\alpha(v_l) < \cdots < \alpha(v_2) < \alpha(v_1)$. Furthermore, we say that a path P *passes through* some subgraph H in α if $I(H) \subseteq I(P)$.

3.1 A Gentle Introduction to the Reduction

We will now give an informal description of the reduction. We hope it will provide the reader with some intuition of why p-Bandwidth is as hard as it is. As already mentioned, the reduction will be from instances (G, k) of p-Even Clique to instances (T, b) of p-Bandwidth. To obtain the results of Theorem 1 we must first of all ensure that (G, k) is a yes-instance if and only if (T, b) is a yes-instance. And furthermore, we require T to be a tree of size polynomial in $|V(G)|$ and k, and that the path-width of T is at most 2. Last, b must be of size $O(k)$.

We start, by providing some boundaries for b-bandwidth orderings of T. Meaning that we force specific parts of T to be the leftmost and rightmost elements of every such ordering. This is done by introducing two stars with $2b$ leafs and adding a path from one of the leafs of the first star to one of the leafs of the second. The two stars will be referred to as walls and the path between them as the main path. Observe that for both of the walls, the leafs must occupy the $2b$ values closest to the value of the center in any b-bandwidth ordering. It follows that the main path must be within the inclusion interval of the two walls, since otherwise the main path would be stretched all to long at some edge passing through a wall.

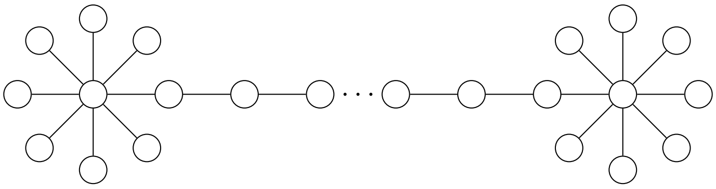

Figure 1: An illustration of the walls for $b = 4$.

We are now controlling the first and last vertices in any b-bandwidth ordering of the graph and hence it is time to start encoding our instance of p-Even Clique. To keep control, the rest of T will be attached to the main path. Before we continue, we select one of the walls and base an ordering of the reduction graph on this selection. This wall will from now on be referred to as the first wall and the other wall will be referred to as the last wall. We now attach k paths, from now on referred to as threads, to the vertex of the main path that is also a leaf of the first wall. Each thread will encode a selection of a vertex in G, and then we will check whether this set of vertices in fact forms a clique or not.

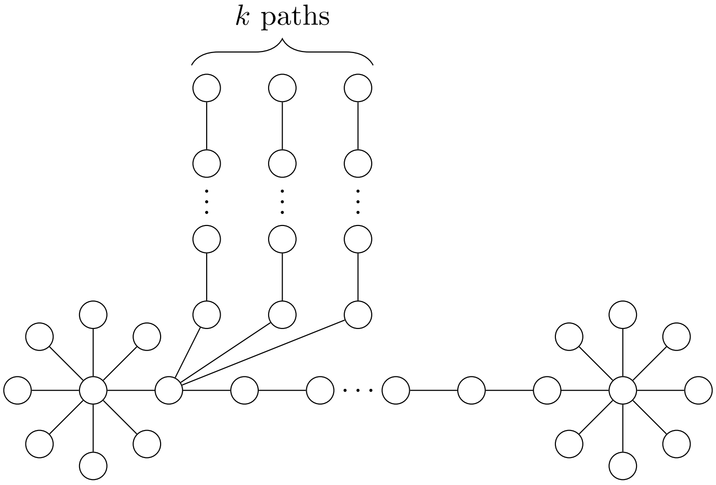

Figure 2: We will use k paths to encode the selection of vertices to be in the clique.

To control how information propagates through a bandwidth ordering, we introduce gates.

A k-gate is a vertex on the main path with $2(b-k-1)$ leafs attached to it, that is in addition to the two neighbours it has on the main path. The goal is to force every thread to pass through every k-gate. Then every thread will position two vertices within the positions of distance at most b away from the center of the gate. And hence there will be $2(b-k-1)+2k+2=2b$ vertices that have to be positioned close to the center, leaving no available room.

A hole is basically two vertices on the main path with some extra space in between. This extra space is obtained by attaching not so many leafs to the two vertices. A knot is a large star centered at one of the threads. The idea is that a knot requires so much space that it cannot be positioned close to a gate. And hence, if a subpath of the main path consists of only gates and holes, a knot that is to be positioned within the inclusion interval of this subpath must be positioned within the hole.

Before we start the process of embedding gadgets on the main path and the threads, we need a guarantee ensuring that any resulting bandwidth ordering will behave nicely. Consider the following situation, we have a graph T and a b-bandwidth ordering α of T. T contains $k+1$ disjoint paths, one of the paths P being of length l such that all the other paths are passing through P in α. In addition there is a set of $(l-1)(b-k-1)$ vertices X disjoint from all the paths, such that the image of X is contained in the inclusions interval of P. Lemma 2 then tells us that that P must be stretched with respect to α, meaning that the vertices of P appear in the same order in T as in α up to reversion and that the distance between two consecutive vertices is b. Furthermore, each of the paths passing through will position exactly one vertex in between any two consecutive vertices of P. As the reader probably can image, we will apply this result with the main path as P and the threads as the paths passing through. This will ensure that how and in which order the vertices appear in α is highly similar to how they are ordered in T.

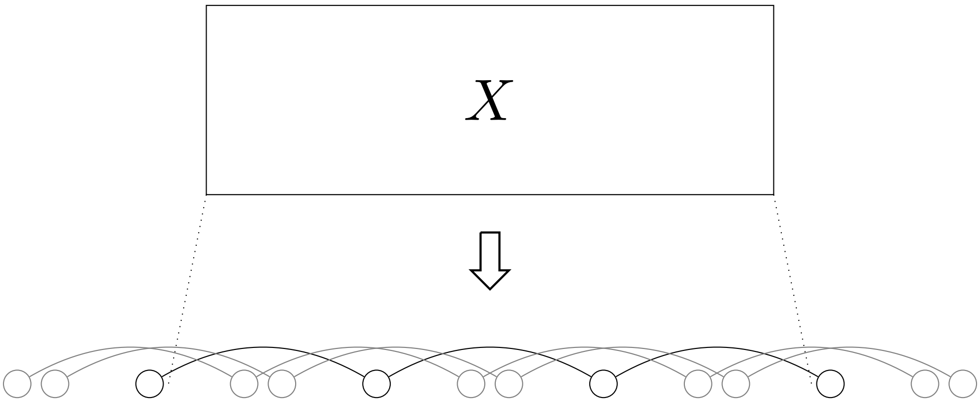

Figure 3: An illustration of Lemma 2. The black path is P and the grey are the ones passing through P.

We will now start to embed gadgets. First we introduce three long sequences of gates on the main path. These sequences naturally partitions our graph into nine sectors. We will refer to them as the first wall, the first wasteland, the first gateland, the selector, the middle gateland the validator, the last gateland, the last wasteland and the last wall. See Figure 4 for an illustration. By making the threads very long, one can force them to pass through every gate. This together with the lemma described above implies that the sectors will appear in the same order in any b-bandwidth ordering as they do in the graph up to reversion.

We aim at forcing a large set over vertices to be embedded in between the fist and the last wasteland. It follows that this part of the main path will be stretched and every thread will position exactly one vertex in between every two consecutive vertices of the main path. Recall that the threads are to encode which vertices we take as our clique. This will be done by how much of the thread is positioned within the inclusion interval of the first wasteland before it

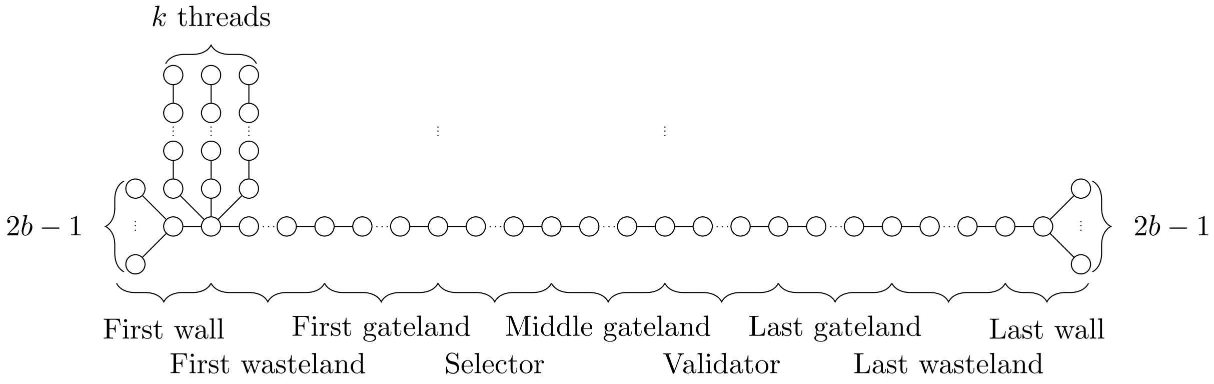

Figure 4: The sectors of our reduction graph.

starts its journey towards the last wasteland. And the job of the wastelands are exactly this, to handle the slack produced by different choices of vertices to form the clique.

We now describe how we enforce the selection of vertices in a manner that allows us to extract this information in a useful way in the validator. First, we order the vertices of G by labeling them with numbers from 1 to n. Basically, we want there to be a linear function describing the number of vertices positioned in the first wasteland given the label of the vertex this thread choose. This is obtained by embedding n holes within the selector, with a certain number of gates in between every pair of consecutive holes. Then we embed a knot on each thread. The idea is that each thread must position its knot within a hole and every hole can contain at most one knot. Which hole the knot is positioned within gives the vertex the thread selects for the clique.

We should now ensure that the selected vertices forms a clique in G. This is done by the validator. The validator is partitioned into $2n - 1$ zones. The first $n - 1$ and last $n - 1$ zones are referred to as neutral zones and nothing is embedded on this part of the main path. The middle zone is referred to as the validation zone. Like the selector, also the validator zone consists of n holes separated by a series of gates. Now the idea is to embed the adjacency matrix of G on the threads row by row in such a way that if vertex i is selected by the thread, then the part representing row number i of the matrix is positioned within the validator zone. The matrix will be represented as follows; Partition the subpath of the thread representing row i into n parts. At part number i we embed a knot. And then, for every non-neighbour j we will attach a leaf to part j. What will happen is that when the vertices are selected the corresponding holes in the validator will be filled up by knots. And then, if two vertices are not adjacent there will also be a leaf that should be positioned within the same hole as a knot, and this there will not be room for. Furthermore, if a vertex is not selected there will not be a knot in the corresponding hole so that it can contain as many leafs as necessary. The last crucial observation is that in the neutral zone, there is room for both leafs and knot to co-exist close in the bandwidth ordering.

The observant reader might recall that we promised some large set of vertices that should be embedded within the first and the last wasteland. This will be handled by attaching paths of appropriate size right after both the first and the second gateland. By making every hole and gate within the selector and validator into $(k + 1)$-holes and $(k + 1)$-gates these paths can travel around in the two sectors filling up the remaining space. We are now done with the informal introduction and for the details we refer to the rest of this section.

3.2 Tools

In this section we give some definitions and results for bandwidth which are crucial for our reduction.

Lemma 1. *Let (T, b) be an instance of p-*Bandwidth *and $\hat{P}_2, P^1, \ldots, P^k$ be $k+1$ disjoint subpaths of T. Given a b-bandwidth ordering α such that $P^1, \ldots, P^k$ pass through $\hat{P}_2$ and there is a set of vertices X disjoint from $\hat{P}_2, P^1, \ldots, P^k$ such that $|X| \geq b-k-1$ and $\alpha(X) \subseteq I(\hat{P}_2)$, then $|\alpha(P^i) \cap I(\hat{P}_2)| = 1$ for every i.*

Proof. Let $\hat{P}_2 = (u, v)$ and assume without loss of generality that $\alpha(u) < \alpha(v)$. From $|I(\hat{P}_2)| \leq b+1$ and

$$\begin{aligned}
|I(\hat{P}_2)| &= |I(\hat{P}_2) \cap \alpha(V(T))| \\
&\geq |I(\hat{P}_2) \cap \alpha(\bigcup P^i \cup X \cup \hat{P}_2)| \\
&= |I(\hat{P}_2) \cap \alpha(\bigcup P^i)| + |I(\hat{P}_2) \cap \alpha(X)| + |I(\hat{P}_2) \cap \alpha(\hat{P}_2)| \\
&\geq |I(\hat{P}_2) \cap \alpha(\bigcup P^i)| + b - k + 1
\end{aligned}$$

it follows that $|I(\hat{P}_2) \cap \alpha(\bigcup P^i)| \leq k$.

Assume for a contradiction that there is a j_1 such that $|\alpha(P^{j_1}) \cap I(\hat{P}_2)| \neq 1$. Then, since $|I(\hat{P}_2) \cap \alpha(\bigcup P^i)| \leq k$ it follows that there is a j_2 such that $|\alpha(P^{j_2}) \cap I(\hat{P}_2)| = 0$. For a path P^i let (v_l^i, v_r^i) maximize $\alpha(v_l^i)$ among the edges in P^i with $\alpha(v_l^i) < \alpha(u)$ and $\alpha(v) < \alpha(v_r^i)$. Let P^j be the path minimizing $\alpha(v_l^j)$ among all paths P^i such that $|\alpha(P^i) \cap I(\hat{P}_2)| = 0$. It follows that for every path P^i either $|\alpha(P^i) \cap I(\hat{P}_2)| \geq 1$ or $|\alpha(P^i) \cap I(v_l^j, u)| \geq 1$. Hence for each i it holds that $|I(v_l^j, v_r^j) \cap \alpha(P^i)| \geq 1$. Furthermore, observe that $|I(v_l^j, v_r^j) \cap \alpha(P^j)| \geq 2$. It follows that

$$\begin{aligned}
|I(v_l^j, v_r^j)| &\geq |I(v_l^j, v_r^j) \cap \alpha(X)| + |I(v_l^j, v_r^j) \cap \alpha(\hat{P}_2)| + |I(v_l^j, v_r^j) \cap \alpha(\bigcup P^i)| \\
&\geq (b-k-1) + 2 + (k+1) \\
&\geq b+2
\end{aligned}$$

Observe that X, $\hat{P}_2$ and $\bigcup P^i$ are disjoint and hence the first line above is valid. Since (v_l^j, v_r^j) is an edge in T and $|I(v_l^j, v_r^j)| \geq b+2$ we have a contradiction to α being a b-bandwidth ordering and hence our proof is complete. □

Corollary 1. *Let (T, b) be an instance of p-*Bandwidth *and $\hat{P}_2, P^1, \ldots, P^k$ be $k+1$ disjoint subpaths of T. Given a b-bandwidth ordering α such that $P^1, \ldots, P^k$ pass through $\hat{P}_2$ and there is a set of vertices X disjoint from $\hat{P}_2, P^1, \ldots, P^k$ such that $|X| \geq b-k-1$ and $\alpha(X) \subseteq I(\hat{P}_2)$, then $|X| = b-k-1$.*

Proof. Assume for a contradiction that $|X| \geq b-k$. Apply Lemma 1 to obtain $|\alpha(P^i) \cap I(\hat{P}_2)| = 1$ for every i. It follows that

$$\begin{aligned}
|I(\hat{P}_2)| &\geq |I(\hat{P}_2) \cap \alpha(X \cup \hat{P}_2 \cup \bigcup P^i)| \\
&\geq |I(\hat{P}_2) \cap \alpha(X)| + |I(\hat{P}_2) \cap \alpha(\hat{P}_2)| + |I(\hat{P}_2) \cap \alpha(\bigcup P^i)| \\
&\geq (b-k) + 2 + k \\
&\geq b+2.
\end{aligned}$$

which is a contradiction to α being a b-bandwidth ordering. □

Lemma 2. *Let (T, b) be an instance of p-*Bandwidth *and $\hat{P}_l, P^1, \ldots, P^k$ be $k+1$ disjoint subpaths of T. Given a b-bandwidth ordering α such that $P^1, \ldots, P^k$ pass through $\hat{P}_l$ and there is a set of vertices X disjoint from $\hat{P}_l, P^1, \ldots, P^k$ such that $|X| \geq (l-1)(b-k-1)$ and $\alpha(X) \subseteq I(\hat{P}_l)$, then $\hat{P}_l$ is stretched with respect to α and $|P^i \cap I(\hat{P}_2)| = 1$ for every i and every $\hat{P}_2 \subseteq \hat{P}_l$.*

Proof. We start by proving $\alpha(v_1) < \alpha(v_2) < \cdots < \alpha(v_l)$ or $\alpha(v_l) < \cdots < \alpha(v_2) < \alpha(v_1)$. Assume otherwise for a contradiction. Then there exists three vertices v_{j-1}, v_j and v_{j+1} such that either $\max\{\alpha(v_{j-1}), \alpha(v_{j+1})\} < \alpha(v_j)$ or $\alpha(v_j) < \min\{\alpha(v_{j-1}), \alpha(v_{j+1})\}$. Since all properties of the lemma is preserved with respect to reversing α we can assume without loss of generality that $\min\{\alpha(v_{j-1}), \alpha(v_{j+1})\} < \alpha(v_j)$. We define a function $f : 2^{\hat{P}_l} \setminus \{\hat{P}_l\} \to \hat{P}_l$ as $f(B) = v_j$ such that $j = \min\Big\{i \mid v_i \in \hat{P}_l \setminus B \text{ and } \{v_{i-1}, v_{i+1}\} \cap B \neq \emptyset\Big\}$. In other words, f gives you the smallest indexed vertex in the open neighbourhood of B. Notice that since $\hat{P}_l$ is connected f is a well-defined function. We will now define $a_1, \ldots, a_t$ and $B_1, \ldots, B_t$. First let $a_1 = \alpha^{-1}(\min\Big\{\alpha(\hat{P}_l)\Big\})$ and $B_1 = \{a_1\}$. Then we let $a_i = f(B_{i-1})$ and $B_i = I(a_1, a_i) \cap \hat{P}_l$ as long as $B_{i-1} \neq \hat{P}_l$. Observe that $B_{i-1} \subset B_i$.

First we will prove that $t < l$. Assume otherwise for a contradiction, clearly then $t = l$. It follows by the construction and our assumption that $\{a_1, \ldots, a_i\} = B_i$ for every i. And by a simple induction we get that $T[\{a_1, \ldots, a_i\}]$ is connected, since this clearly holds for $i = 1$ and for $i > 1$ observe that a_i has a neighbour in B_{i-1} by construction. Let c be so that $a_c = v_j$. Since v_j is separating v_{j-1} and v_{j+1} in $\hat{P}_l$ and $v_j \notin B_{c-1}$ it follows that $\{v_{j-1}, v_{j+1}\} \nsubseteq B_{c-1}$. Furthermore, since $\max\{\alpha(v_{j-1}), \alpha(v_{j+1})\} < \alpha(v_j)$ it holds that $\{v_{j-1}, v_j, v_{j+1}\} \subseteq B_c$. But this contradicts $\{a_1, \ldots, a_i\} = B_i$ and hence we know that $t < l$. It follows, due to the pidgin hole principle, that there is a d such that $|I(a_{d-1}, a_d) \cap \alpha(X)| > b-k-1$. By construction there is a neighbour a' of a_d among $a_1, \ldots, a_{d-1}$. Observe that $|I(a', a_d) \cap \alpha(X)| > b-k-1$ and apply Corollary 1 with $\hat{P}_2 = (a', a_d)$ to obtain a contradiction. Hence we can conclude that $\alpha(v_1) < \alpha(v_2) < \cdots < \alpha(v_l)$ or $\alpha(v_l) < \cdots < \alpha(v_2) < \alpha(v_1)$.

We will now prove $|P^i \cap I(\hat{P}_2)| = 1$ for every i and every $\hat{P}_2 \subseteq \hat{P}_l$. Observe that if there is a $\hat{P}_2$ such that $|I(\hat{P}_2) \cap \alpha(X)| \neq b-k-1$, then there is a $\hat{P}_2'$ such that $I(\hat{P}_2') \cap \alpha(X)| > b-k-1$. But this contradicts Corollary 1 and hence we get that $|I(\hat{P}_2) \cap \alpha(X)| = b-k-1$ for every $\hat{P}_2 \subseteq \hat{P}_l$ and then it follows directly from Lemma 1 that $|P^i \cap I(\hat{P}_2)| = 1$ for every $\hat{P}_2 \subseteq \hat{P}_l$. Hence

$$\begin{aligned}
|I(\hat{P}_2)| &\geq |I(\hat{P}_2) \cap \alpha(X \cup \hat{P}_2 \cup \bigcup P^i)| \\
&\geq |I(\hat{P}_2) \cap \alpha(X)| + |I(\hat{P}_2) \cap \alpha(\hat{P}_2)| + |I(\hat{P}_2) \cap \alpha(\bigcup P^i)| \\
&\geq b-k-1+2+k \\
&\geq b+1
\end{aligned}$$

and it follows that $\hat{P}_l$ is stretched with respect to α. □

Corollary 2. *Let (T, b) be an instance of p-*Bandwidth *and $\hat{P}_l, P^1, \ldots, P^k$ be $k+1$ disjoint subpaths of T. Given a k-bandwidth ordering α such that $P^1, \ldots, P^k$ passes through $\hat{P}_l$ and there is a set of vertices X disjoint from $\hat{P}_l, P^1, \ldots, P^k$ such that $|X| \geq (l-1)(b-k-1)$ and $\alpha(X) \subseteq I(\hat{P}_l)$, then $|X| = (l-1)(b-k-1)$.*

Proof. Assume for a contradiction that $|X| > (l-1)(b-k-1)$. Then there is a $\hat{P}_2 \subseteq \hat{P}_l$ such that $|X \cap I(\hat{P}_2)| \geq b-k$ which is a contradiction by Corollary 1. □

3.3 Gadgets

We will now introduce the gadgets used for the reduction. They will all be defined on paths of various lengths. And later on when we say that a gadget is embedded on some path, this means that the path referred to together with some of its neighbours is an instantiation of the gadget.

Definition 2. *Let (T, b) be an instance of p-Bandwidth and H be a subgraph of T with a vertex labeled in and another vertex labeled out. We say that H is* functioning *in T if T contains two walls W_{in} and W_{out} such that*

- *W_{in}, W_{out} and H are disjoint,*
- *there is a path P_{in} from in to W_{in} avoiding $(H - in)$ and W_{out} and*
- *there is a path P_{out} from out to W_{out} avoiding $(H - out)$, W_{in} and P_{in}.*

If H is functioning in T let $W_{in}(H, T), W_{out}(H, T), P_{in}(H, T)$ and $P_{out}(H, T)$ denote a witness of this.

Walls

A *wall* is a star with $2b$ leaves. The high degree vertex of a wall W will be referred to as the *center* of the wall. We will turn the endpoints of the main path into walls to control the endpoints of all valid b-bandwidth orderings. The next lemma gives us this behaviour.

Lemma 3. *Let (T, b) be an instance of p-BANDWIDTH such that T contains two disjoint walls W_1 and W_2 with centers c_1 and c_2 as subgraphs. Let H be a connected component of $T - (W_1 \cup W_2)$ connected by edges to both walls in T. Then, for any b-bandwidth ordering α of T and any vertex $v \in H$ it follows that $\alpha(v) \in I(c_1, c_2)$.*

Proof. Assume without loss of generality that $\alpha(c_1) < \alpha(c_2)$. For a contradiction, assume that $\alpha(v) < \alpha(c_1)$. Let u_l be the leaf in W_1 minimizing α and u_r the leaf maximizing α. Furthermore, let P^1 be a path from v to c_2 in $T[V(H) \cup W_2]$ and $\hat{P}_3$ the path (u_l, c_1, u_r). Observe that P^1 passes through $\hat{P}_3$, since $\alpha(W_1) = [\alpha(c_1) - b, \alpha(c_1) + b]$. Let $X = V(W_1) - \hat{P}_3$ and note that $|X| = 2b - 2$. Apply Corollary 2 on $\hat{P}_3, P^1$ and X to obtain a contradiction, since $(3 - 1)(b - 1 - 1) = 2b - 4 < 2b - 2 = |X|$. For $\alpha(v) > \alpha(c_2)$ we apply a symmetric argument and hence our proof is complete. □

Gates

For an integer $k \geq 0$ a *k-gate*, denoted Π_k, is a star with $2(b - k)$ leaves. The function of the k-gate will be to reduce the number of paths passing this point to at most k. The high degree vertex of the star will be referred to as the *center* of the gate. In addition one leaf will be labeled *in* and another labeled *out*.

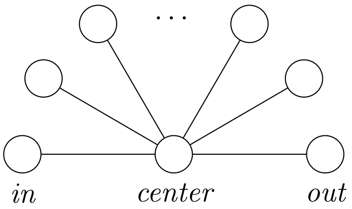

Figure 5: A k-gate with the special vertices marked with tags below.

Lemma 4. *Let (T, b) be an instance of p-BANDWIDTH such that T contains a gate Π_k and paths $P^1, \ldots, P^k$ as disjoint subgraphs with Π_k being functioning in $T - (\bigcup P^i)$. Given a b-bandwidth ordering α such that $\max\{\alpha(W_{in}(\Pi_k, T - \bigcup p^i))\} < \min\{\alpha(W_{out}(\Pi_k, T - \bigcup p^i))\}$ and every path P^i passes through the gate it follows that:*

(I) $\alpha(N[\mathit{center}]) \subseteq B \subseteq \alpha(\bigcup P^i \cup N[\mathit{center}])$,

(II) $\alpha(\mathit{in}) < \alpha(\mathit{center}) < \alpha(\mathit{out})$ *and*

(III) $|\alpha(P^i) \cap B_l| = |\alpha(P^i) \cap B_r| = 1$ *for every* $i \in [1, k]$

for $c = \alpha(\mathit{center})$, $B = [c - b, c + b]$, $B_l = \{i \in B \mid i < c\}$ *and* $B_r = \{i \in B \mid c < i\}$.

Proof. We start by proving *(III)*. For every path P^i we know that there are $u, v \in P^i$ such that $\alpha(u) < \min \alpha(\Pi_k)$ and $\max \alpha(\Pi_k) < \alpha(v)$. Assume that $u \notin B_l$ and follow the path from u to v until you reach the first vertex u' such that $\alpha(u') \geq c - b$. Let u'' be the vertex we reached right before u'. From the definition of α it follows that $\alpha(u') - \alpha(u'') \leq b$ and hence $u' \in B_l$ and $|P^i \cap B_l| = 1$. Reverse α and apply the argument on the path from v to u to obtain $|P^i \cap B_r| = 1$.

We continue by proving *(I)*. It follows directly from the fact that $\mathrm{bw}(T, \alpha) \leq b$ that $N[\mathit{center}] \subseteq B$. Since $|B \cap (\bigcup P^i \cup N[\mathit{center}])| = |B \cap \bigcup P^i| + |B \cap N[\mathit{center}]| = 2k + 2(b - k) + 1 = 2b + 1$ and $|B| = 2b + 1$ it follows that $B \subseteq \bigcup P^i \cup N[\mathit{center}]$. It remains to prove *(II)*. Observe that $\max \alpha(W_{\mathit{in}}) < \min\{\alpha(\mathit{in}), \alpha(\mathit{center})\}$ by Lemma 3. Assume for a contradiction that $\alpha(\mathit{in}) > \alpha(\mathit{center})$. Since $P_{\mathit{in}}(\Pi_k, T - (\bigcup P^i))$ is a path from *in* to $W_{\mathit{in}}(\Pi_k, T - (\bigcup P^i))$ and the bandwidth of α is b it follows that $|B \cap W_{\mathit{in}}(\Pi_k, T - (\bigcup P^i)| \geq 2$, but this contradicts *(I)* and hence $\alpha(\mathit{in}) < \alpha(\mathit{center})$. A symmetric argument gives us $\alpha(\mathit{center}) < \alpha(\mathit{out})$ and our proof is complete. □

Knots and Holes

Assuming $b \geq 2k + 14$ and b to be dividable by 4 we give the following two definitions. A *k-knot* is a path $P = (\mathit{first}, \mathit{center}, \mathit{last})$ with $\frac{3}{2}b - k - 1$ leaves attached to *center*. A *k-hole* consists of a path $P = (\mathit{in}, \mathit{in\ center}, \mathit{out\ center}, \mathit{out})$ with $\frac{3}{4}b - k - 1$ leaves attached to both *in center* and *out center*.

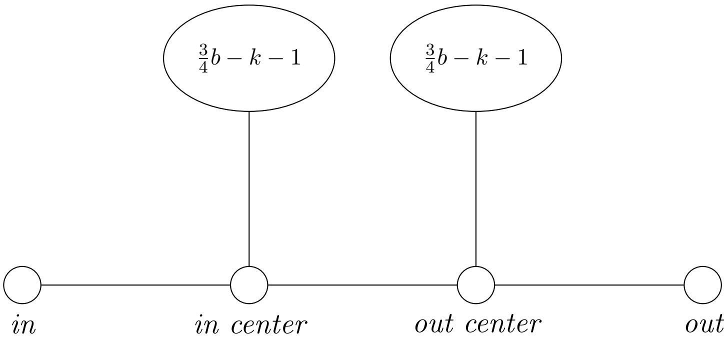

Figure 6: A hole. The ellipse shaped vertices represent some number of leafs.

Lemma 5. *Let (T, b) be an instance of p-*Bandwidth *such that T contains a k-hole H and paths $P^1, \ldots, P^k$ with a k-knot K embedded on one of the paths as disjoint subgraphs with H being functioning in $T - (\bigcup P^i)$. Given a b-bandwidth ordering α such that $P^1, \ldots, P^k$ passes though H, $\max\{\alpha(W_{\mathit{in}}(\Pi_k, T - \bigcup p^i))\} < \min\{\alpha(W_{\mathit{out}}(\Pi_k, T - \bigcup p^i))\}$ and $I(K \cup H) \subset I(\mathit{in}, \mathit{out})$ it holds that*

(I) $\alpha(\mathit{in}) < \alpha(\mathit{in\ center}) < \alpha(\mathit{out\ center}) < \alpha(\mathit{out})$,

(II) $|I(\hat{P}_2) \cap \alpha(P^i)| = 1$ *for every i and every* $\hat{P}_2 \subset (\mathit{in}, \mathit{in\ center}, \mathit{out\ center}, \mathit{out})$ *and*

(III) $\alpha(\mathit{in\ center}) < \alpha(\mathit{center}) < \alpha(\mathit{out\ center})$.

Proof. First we prove the correctness of *(I)* and *(II)*. Let $\hat{P}_4 = (\mathit{in}, \mathit{in\ center}, \mathit{out\ center}, \mathit{out})$ and let X_c, X_i and X_o be the set of leaves attached to *center*, *in center* and *out center* respectively. Apply Lemma 2 with $X = X_i \cup X_c \cup X_o$ to obtain *(II)* and either $\alpha(\mathit{in}) < \alpha(\mathit{in\ center}) < \alpha(\mathit{out\ center}) < \alpha(\mathit{out})$ or $\alpha(\mathit{out}) < \alpha(\mathit{out\ center}) < \alpha(\mathit{in\ center}) < \alpha(\mathit{in})$ since $|X| = 2\left(\frac{3}{4}b - k - 1\right) + \frac{3}{2}b - k - 1 = (4-1)(b-k-1)$. Assume for a contradiction that $\alpha(\mathit{out}) < \alpha(\mathit{out\ center}) < \alpha(\mathit{in\ center}) < \alpha(\mathit{in})$. Then there is a vertex $v \in P_{in}(H, T - (\bigcup P^i)) \cap \alpha^{-1}(I(H)) \setminus \{\mathit{in}\}$. Apply Corollary 2 with $X = X_i \cup X_c \cup X_o \cup \{v\}$ to get a contradiction and hence *(I)* holds.

It remains to prove *(III)*. Assume for a contradiction that $\alpha(\mathit{center}) \notin I(\mathit{in\ center}, \mathit{out\ center})$. Furthermore, assume without loss of generality that $\alpha(\mathit{center}) \in I(\mathit{in}, \mathit{in\ center})$. It follows from Lemma 2 that $\hat{P}_4$ is stretched and hence $X_l \cup X_m \subseteq I(\hat{P}_3)$ for $\hat{P}_3 = (\mathit{in}, \mathit{in\ center}, \mathit{out\ center})$. Apply Corollary 2 with $X = X_i \cup X_c$ to obtain a contradiction since $|X| = \frac{3}{4}b - k - 1 + \frac{3}{2}b - k - 1 = \frac{9}{4}b - 2k - 2 > (3-1)(b-k-1)$ and hence our proof is complete. □

3.4 The Reduction

We will now give a reduction from an instance (G, k) of p-Even Clique to an instance (T, b) of p-Bandwidth. The correctness and implications will be given in the two following sections. The resulting instance T can be divided into eleven parts. Nine of them lie on the main path and will in the future be referred to as the sectors of the main path. The nine sectors are the first wall, the first wasteland, the first gateland, the selector, the middle gateland, the validator, the last gateland, the last wasteland and the last wall. The two other components will be referred to as threads and fillers. Each of the components have a specific purpose with respect to how a b-bandwidth ordering can be. The walls will force everything else to be positioned within them. The threads are k paths attached to the first wasteland and each of them represents a vertex in the supposed clique in G. To encode how G looks like we attach leaves to the threads, which will be referred to as the dangelments of the threads. How much of a thread that is in the inclusion interval of the first wasteland decides which vertex in G this thread represents. To propagate this information the threads are made so long that they will have to enter the inclusion interval of the last wasteland. The selectors job is to make sure that the decisions made by the threads are unique and valid. The validator will verify that the selected vertices in fact is a clique. And the fillers and the gatelands will control how information propagates between the other components. When describing the components on the main path we will assume the vertices of the path to be named $u_1, \ldots,$ with u_1 being the leftmost vertex in Figure 2.

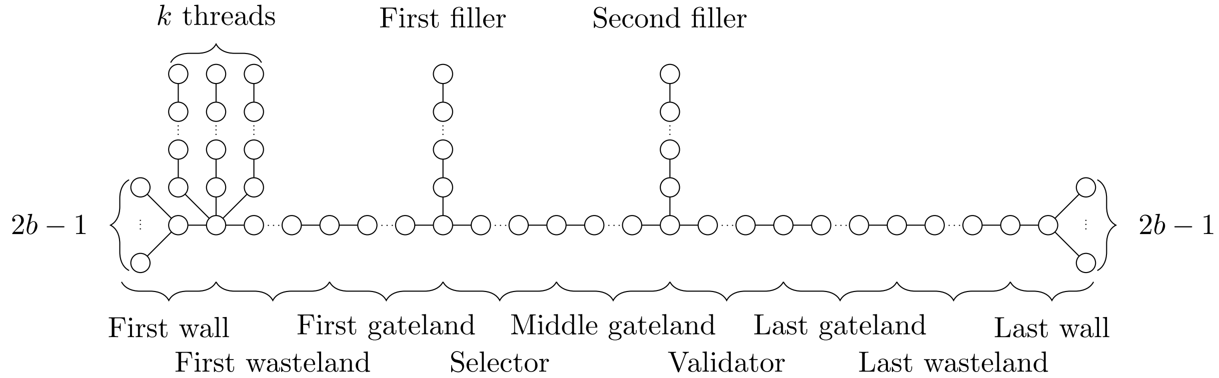

Figure 7: A subgraph of T with the components marked.

When discussing vertices and subgraphs of T we will apply an ordering based on the distance from the center of the first wall, the leftmost wall in Figure 7. We will say that a vertex u comes before a vertex v if u is closer to the center of the first wall than v. For subgraphs, we will compare the minimized distance over all vertices in each subgraph. To complete our construction we need an ordering of the vertices of G, we therefore let $V(G) = \{v_1, \ldots, v_{|V(G)|}\}$.

The First Wall, Wasteland and Gateland

To ensure enough space for the gadgets in the validator we introduce the *pull-factor* p, which will correspond to the distance from the *in* vertex of a hole in the selector to the *in* vertex of the next hole. The pull-factor is $4n + 3$ in our reduction, but will for convenience mostly be referred to as p.

The first sector we will embed is the first wall. This is done by turning u_1 into the center of a wall by attaching leafs to it. Second comes the first wasteland. This is done by attaching nothing to the vertices u_2 until u_{m_1} for $m_1 = pnk + 2$. Note that u_2 is the vertex for which the threads are connected. After this we embed bm_1 consecutive k-gates from u_{m_1} to $u_{(2b+1)m_1}$ to create the first gateland. This is done in such a way that the *in* vertex of the i'th gate is the *out* vertex of the $i-1$'th gate.

The Selector

The selector will control the choices done by the threads. The idea is to let the selector have $|V(G)|$ sparse intervals, namely holes, and let each of the threads have a big knot, which can only be placed within such an interval. The vertex selected by a thread is then decided by which hole its knot is placed within.

The embedding of the selector starts where the first gateland ended, at vertex $u_{(2b+1)m_1}$. Note that this is the vertex where the first filler is attached in Figure 2. We now embed $|V(G)|$ $(k+1)$ holes with $(p-3)/2$ consecutive $(k+1)$-gates in between every consecutive pair of holes on the path $(u_{(2b+1)m_1}, \ldots, u_{(2b+1)m_1+p(n-1)+3})$. After this we embed $b(p(n-1)+3)$ consecutive $(k+1)$-gates. In total, the selector is embedded on the vertices $(u_{(2b+1)m_1}, \ldots, u_{m_2})$ for $m_2 = (2b+1)m_1 + (2b+1)(p(n-1)+3)$.

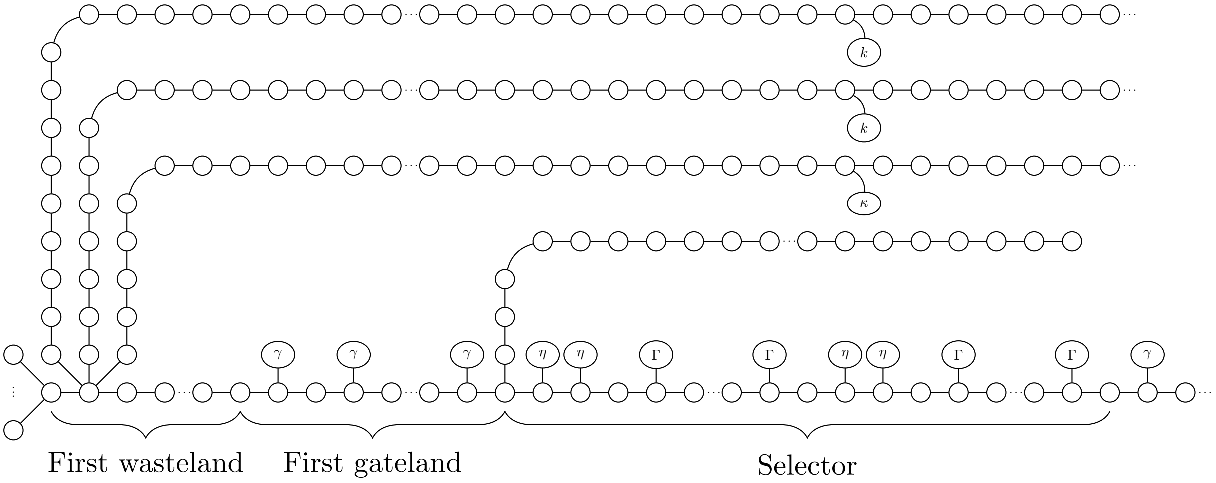

Figure 8: Illustration of the selector where $\gamma = 2(b-k-1)$, $\Gamma = 2(b-k-2)$, $\eta = \frac{3}{4}b - k - 1$ and $\kappa = \frac{3}{2}b - k - 1$.

The Middle Gateland

The middle gateland consist of bm_2 consecutive k-gates, embedded on the main path from vertex u_{m_2} to vertex $u_{(2b+1)m_2}$.

The Validator

We will now give the validator. Its job is to verify that the selected vertices of the threads in fact is a clique. The validator starts with $n-1$ neutral zones, followed by a validation zone and another $n-1$ neutral zones. After this there will be $b(2n-1)(4n+3)$ consecutive $(k+1)$-gates. A neutral zone is a P_{4n+4}. The zones will be joined by sharing endpoints in the same style as the gadgets in the selector. The validation zone consists of a $\hat{P}_{4n+4}$ where there is n $(k+1)$-holes sharing endpoints embedded on the last $3n+1$ vertices. The validator is hence embedded on the vertices $(u_{(2b+1)m_2}, \dots, u_{m_3})$ for $m_3 = (2b+1)m_2 + (2b+1)(2n-1)(4n+3)$.

The Last Gateland, Wasteland and Wall

The last gateland consists of bm_3 consecutive k-gates embedded on the vertices $(u_{m_3}, \dots, u_{(2b+1)m_3})$. After this we embed the last wasteland, which means that we leave the vertices $(u_{(2b+1)m_3}, u_{b^2(2b+1)m_3})$ untouched. Finally we turn the vertex $u_{b^2(2b+1)m_3+1}$ into the center of the last wall by attaching leafs to it.

The Threads and Their Danglements

We will now describe the threads and their danglements. As they are all isomorphic, it is sufficient to describe one of them. Let us name the vertices on the thread by $t_2, \dots$ with $t_2 = u_2$. The leafs neighbouring to the thread will be referred to as its danglements. First we turn $t_{(2b+1)m_1+1}$ into the center of a $(k+1)$-knot by attaching leafs to it. Starting at vertex $t_{(2b+1)m_2+(n-1)(4n+3)}$ we consider n consecutive, disjoint P_{4n+3}. For $\hat{P}_{4n+3}$ number i we do the following. We divide the $\hat{P}_{4n+3}$ into disjoint subpaths, first a $\hat{P}_{n+3}$ followed by n P_3. Consider the j'th P_3. If $i = j$ we turn the middle vertex of the P_3 into the center of a $(k+1)$-knot. Otherwise we attach a single leaf to the middle vertex if $(v_i, v_j) \notin E(G)$. This leaf will be referred to as a non-neighbouring leaf. After this we extend the thread with additional $b(2b+1)m_3$ vertices. We would like to make the reader aware of the fact that a thread is a path and the vertices connected to it, is its dangelments.

The Fillers

A fillers job is to fill up all available room within a component of T to force this part of the main path to be stretched. To accomplish this we let the filler connected to $u_{(2b+1)m_1}$ be of length $(n-k)(\frac{3}{2}b - k - 2) + (2b+1)(p(n-1)+3)$. And the filler attached to $u_{(2b+1)m_2}$ to be of length $(b-1)(4n+3)(2n-1) + 2b(4n+3)(2n-1) - (k(2n-1)(4n+3) + k(n(\frac{3}{2}b - k - 2) + n^2 - n - 2m) + 2n(\frac{3}{4}b - k - 2))$.

3.5 Correctness

With the next lemmas we will prove the correctness of the reduction. After this we will continue by giving the implications of this reduction, which are the main results of this section. Recall that $b = 4k + 16$ and $p = 4n + 3$.

Lemma 6. *Given a yes-instance (G, k) of p-*Even Clique *the reduction instance (T, b) is a yes-instance of p-*Bandwidth.

Proof. We will now give a sparse ordering α of bandwidth $b = 4k + 16$, meaning that the image of α might not be an interval. To obtain a proper bandwidth ordering one can just compress α. During the description of α a *position* is a number in $\mathcal{N}$ that will be in the image of α and a vertex v is said to be *positioned* if the value $\alpha(v)$ has been given. Furthermore, we will say that v is *positioned at* c if $\alpha(v) = c$. By *reserving* a position for a subgraph H of T we guarantee that if a vertex will be positioned at that specific position, it will be a vertex of H. And by a position being *available* we will mean that no vertex has been positioned at that specific position so far. Let $C_k = \{c_1, \ldots, c_k\}$ be a k-clique in G.

For a vertex u_i on the main path let $\alpha(u_i) = bi+1$. We continue by positioning the remainders of the two walls. And let c_f be the center of the first wall and L_f be the neighbouring leaves of c_f. Let $\alpha(L_f) = [\alpha(c_f) - b, \alpha(c_f) + b - 1] \setminus \{\alpha(c_f)\}$ in some arbitrary way. Similarly for the last wall, let $\alpha(L_l) = [\alpha(c_l) - b - 1, \alpha(c_l) + b] \setminus \{\alpha(c_l)\}$. Observe that for every two vertices u and v of T such that both $\alpha(u)$ and $\alpha(v)$ has been described, it holds that $\alpha(u) \neq \alpha(v)$. Furthermore, if uv is an edge in T it is true that $|\alpha(u) - \alpha(v)| \leq b$.

Order the threads of T and name them $\tau_1, \ldots, \tau_k$. Let u and v be two neighbours on the main path such that neither u nor v is the center of a wall and so that $\alpha(u) < \alpha(v)$. Observe that there is $b - 1$ available positions within $I(u, v)$. Reserve the k positions in the middle of $I(u, v)$, one for each of the k threads. If there are two positions equally close to the middle, take the leftmost one. The leftmost is reserved for the first thread, the second to leftmost for the second thread and so forth.

For every i let j_i be such that $c_i = v_{j_i}$. Consider hole number j_i on the main path starting at the first wall, with h_1, h_2, h_3 and h_4 being the vertices on the main path for which the hole is embedded on such that $\alpha(h_1) < \alpha(h_2) < \alpha(h_3) < \alpha(h_4)$. Thus h_1, h_2, h_3 and h_4 are the *in*, *in center*, *out center* and *out* vertices of the hole respectively. Let c be the center of the first knot on τ_i and r the reserved position for τ_i in α within $I(h_2, h_3)$. We then set $\alpha(c) = r$ and complete the following procedure in the left (and right) direction on the thread τ_i. Let P be the path from c to $N(u_2) \cap \tau_i$ (or to the end of the thread). If every vertex of P is positioned we stop. Otherwise, let u be the vertex closest to c on P not yet positioned. Furthermore, let $\hat{P}_2$ be the rightmost (leftmost) P_2 on the main path to the left (right) of the hole such that the position reserved for τ_i is available in $I(\hat{P}_2)$. If $\hat{P}_2$ is not part of any wasteland we set $\alpha(u)$ to this reserved position and continue. Otherwise we consider two cases. If we are right of r we position u at the leftmost position within $I(\hat{P}_2)$ that is either not reserved yet, or reserved for τ_i. If we are left of r we again consider two cases. Either there are exactly as many positions to the left of r reserved for τ_i as there are vertices before c not yet positioned. In that case we position u at the reserved position for τ_i within $I(\hat{P}_2)$. Otherwise, we position u at the rightmost position in $I(\hat{P}_2)$ that is either not reserved yet, or reserved for τ_i. Observe that if uv is an edge of τ_i there are positions reserved for τ_i, x and y, such that $y > x$ and $y - x = b$ and $\alpha(u)$ and $\alpha(v)$ are contained in $[x, y]$. It follows that $|\alpha(u) - \alpha(v)| \leq b$.

Note that the number of vertices on a thread that will be positioned to the left of r is $(2b + 1)m_1$ and that, by construction, $2bm_1$ of these will be within the inclusion interval of the first gateland. Hence it can be observed that there are at most km_1 vertices from the threads within the inclusion interval of the first wasteland. Recall that the distance from u_2 to the first vertex of the first gateland is $m_1 - 2$. Hence there are $(b - 1)(m_1 - 2) > km_1$ available positions within the inclusion interval of the first wasteland, before we position the threads. By the same kind of argument there are $(b - 1)(b^2 - 1)(2b + 1)m_3$ available positions in the inclusion interval of the last wasteland before positioning the thread. Recall that the length of a thread is bounded above by

$$\begin{aligned}&(2b+1)m_2+(n-1)(4n+3)+(4n+3)n+b(2b+1)m_3\\<&(2b+3)m_2+b(2b+1)m_3\\<&2b(2b+3)m_3.\end{aligned}$$

It follows that for every pair of vertices u and v such that both $\alpha(u)$ and $\alpha(v)$ has been described if holds that $\alpha(u) \neq \alpha(v)$.

Recall that every k-gate of T is embedded on the main path. And hence for every k-gate in T there are k paths passing through it with respect to α. Hence there are $2(b-k-1)$ positions available between the left and the right leaf and the rest of the leaves can be positioned in any way within this interval. Clearly, for every pair of vertices u and v of T, such that both $\alpha(u)$ and $\alpha(v)$ are described it holds that $\alpha(u) \neq \alpha(v)$. And furthermore, if uv is an edge of a k-gate it holds that $|\alpha(u)-\alpha(v)| \leq b$. For every $\hat{P}_2$ on the main path such that $\hat{P}_2$ is not in a subgraph of a wasteland and there are available positions in $I(\hat{P}_2)$ we reserve the position to the right of the k positions reserved for the threads, for the fillers. Observe that any $\hat{P}_2$ such that this position is not available either is a subgraph of a wasteland or a k-gate (which has no available positions).

We will now position the leaves of the knots. Let K be a knot in T. The center c of k is a vertex of a thread and hence $\alpha(c)$ has already been described. Let $\hat{P}_2$ be the P_2 of the main path such that $\alpha(c) \in I(\hat{P}_2)$. Position the leaves attached to c as close to the middle of $I(\hat{P}_2)$ as possible by only using available positions, that are not reserved. If there are two such positions equally close to the middle, we take the leftmost one. Let $\hat{P}_4$ be the P_4 of the main path such that $\hat{P}_2$ contains the internal vertices of $\hat{P}_4$. It can be observed, by where the knots are embedded on the thread and where the threads are positioned in α, that $\hat{P}_4$ is either a subgraph of a hole or a neutral zone. Furthermore, if c' is the center of some other knot and $\hat{P}_2'$ is the P_2 of the main main such that $\alpha(c')$ is contained in its inclusion interval, then it can be observed that $\hat{P}_2'$ and $\hat{P}_4$ are disjoint. Hence, we see that there are $2(b-k-2)$ positions available and non-reserved within the inclusion interval of $\hat{P}_4$. Recall that a knot consists of $\frac{3}{2}b-k-2$ leaves and that $b = 4k+16$, and hence $2(b-k-2) \geq \frac{3}{2}b-k-2$. It follows that for every two vertices u and v of T such that both $\alpha(u)$ and $\alpha(v)$ have been described, $\alpha(u) \neq \alpha(v)$. Furthermore, it uv is an edge of T it holds that $|\alpha(u)-\alpha(v)| \leq b$.

Let $\hat{P}_4 = (u_h, u_{h+1}, u_{h+2}, u_{h+3})$ be some subpath of the main path such that a hole is embedded on it. Position the leaves attached to u_{h+1} to the leftmost non-reserved, available positions and the leaves attached to u_{h+2} to the rightmost non-reserved, available positions, within $I(\hat{P}_4)$. Furthermore, for the leaves representing non-neighbours, position it at the position available and not reserved closest to its neighbour. If there are two such positions, any of the two will do. It can be observed, by where the knots are embedded on the threads and where the knots and positioned that no two knots are positioned within the inclusion interval of a hole. And furthermore, that at most k non-adjacency leaves are positioned within the inclusion interval of a hole. At last, since C_k is a clique it holds that no knot and non-neighbour leaf is positioned with the inclusion interval of a hole. Recall that k is even and hence $\frac{3}{2}b-k-2 = 5k+22$ is even. It follows that the leaves of a knot is evenly distributed among the two sides of the center. Recall that the number of leaves in a hole is $\frac{3}{2}b-2k-4$. There are $3k$ vertices from the threads positioned within the inclusion interval of $\hat{P}_4$ and there are $3b-3k-6$ leaves attached to one hole and one knot. Since there are more than k leaves attached to a knot, it can be observed that for any two vertices u and v such that at least u or v is positioned within the inclusion interval of $\hat{P}_4$ it holds that $\alpha(u) \neq \alpha(v)$. And furthermore, if uv is an edge in T it holds that $|\alpha(u)-\alpha(v)| \leq b$.

Consider danglements positioned within the inclusion interval of a $\hat{P}_4$ that is a subgraph of a neutral zone. One can observe that there is at most k non-neighbouring leaves and at most one clique positioned within the inclusion interval of $\hat{P}_4$. And hence the same argument as above can be applied to show that for every two vertices u and v of T such that both $\alpha(u)$ and $\alpha(v)$ has been described, it holds that $\alpha(u) \neq \alpha(v)$. Furthermore, if uv is an edge of T it is true that $|\alpha(u) - \alpha(v)| \leq b$.

It remains to describe the positioning of each of the fillers. Let u be the vertex on the filler closest to the main path not yet positioned and r lowest value bigger than the α-value of the intersection vertex between the filler and the main path that is not taken. Set $\alpha(u) = r$ and continue. Recall that the length of the path where the selector is embedded is $(n-1)p + 3 + 2b(p(n-1)+3)$, and hence there were $(b-1)((n-1)p + 3 + 2b(p(n-1)+3))$ available positions within the inclusion interval of the selector after only the main path had been positioned. Observe that the threads now occupies $k((n-1)p+3+2b(p(n-1)+3))$ of these positions, the $k+1$-gates $((p-3)(n-1)/2 + b(p(n-1)+3))2(b-k-2)$ of the positions, the knots $k(\frac{3}{2}b - k - 2)$ positions, the holes $2n(\frac{3}{4}b - k - 2)$ positions and the filler $(n-k)(\frac{3}{2}b - k - 2) + (2b+1)(p(n-1)+3)$. By substituting p by $4n+3$ and b by $4k+16$ one can verify that the vertices positioned equals the amount of positions available within the inclusion interval of the selector. The expression for the once available positions within the inclusion interval of the selector S and the number of vertices now positioned within it, disregarding the main path, namely X, is given below.

$$\begin{aligned} S &= k((n-1)p + 3 + 2b(p(n-1)+3)) \\ &\quad + ((p-3)(n-1)/2 + b(p(n-1)+3))2(b-k-2) \\ &\quad + k(\frac{3}{2}b - k - 2) + 2n(\frac{3}{4}b - k - 2) \\ &\quad + (n-k)(\frac{3}{2}b - k - 2) + (2b+1)(p(n-1)+3) \\ &= (b-1)((n-1)p + 3 + 2b(p(n-1)+3)) = X. \end{aligned}$$

It follows that for every two vertices u and v such that both $\alpha(u)$ and $\alpha(v)$ have been described, it holds that $\alpha(u) \neq \alpha(v)$. Recall that for every $\hat{P}_2$ that is a subgraph of the main path and the selector there was a position reserved for the fillers. And hence for every edge uv of the first filler, there are positions reserved for the filler, x and y such that $y - x = b$ and $\alpha(u)$ and $\alpha(v)$ is contained within $[x, y]$. It follows directly that $|\alpha(u) - \alpha(v)| \leq b$. For the second filler, we observe that there were $(b-1)(4n+2)(2n-1)$ available positions within the inclusion interval of the validator when only the main path had been positioned. And furthermore, now the n holes occupies $2n(\frac{3}{4}b - k - 2)$ of these positions, the threads $k(2n-1)(4n+2)$ of the positions, the knots $kn(\frac{3}{2}b - k - 2)$ and the non-neighbouring leaves $k(n^2 - n - 2m)$. By a similar argument as for the first filler, one can prove that for every u and v of T it holds that $\alpha(u) \neq \alpha(v)$ and if uv is an edge of T then $|\alpha(u) - \alpha(v)| \leq b$. This completes the description of α and the argument is complete. □

Given a reduced instance (T, b) and a b-bandwidth ordering α we say that a k-gate in T is *blocked* with respect to α if every thread in T pass through the gate.

Lemma 7. *Let (T, b) be the result of the reduction for some instance of p-*Even Clique *and α a b-bandwidth ordering of T. Then every k-gate in T is blocked with respect to α.*

Proof. By Lemma 3 we know that the first wall is either the leftmost or the rightmost elements of α. Observe that every k-gate in T is blocked with respect to α if and only if every k-gate in T is blocked with respect to α reversed. Hence it is sufficient to prove that every k-gate is blocked when the first wall is the leftmost elements of α.

Assume for a contradiction that there is a k-gate Π and a thread τ such that τ is not passing through Π. Let P be the path from u_2 to the *out* vertex of Π and let $X = V(\tau) - u_2$. By Lemma 3 we know that $\alpha(u_2) = \min \alpha(\tau)$ and that $\alpha(u_2) < \min \alpha(\Pi)$. It follows by the definition of passing through that $\max \alpha(\tau) \leq \max \alpha(\Pi)$ and hence $\alpha(X) \subseteq I(P)$. Recall that $|E(P)| \leq (2b+1)m_3 - 2$ and $|X| > b(2b+1)m_3$. It follows directly that $|I(P)| \leq b((2b+1)m_3 - 2) + 1 < b(2b+1)m_3 < |X|$ which is a contradiction. □

Recall that the main path of the reduction instance consist of 9 sectors, namely the first wall, the first wasteland, the first gateland, the selector, the middle gateland, the validator, the last gateland, the last wasteland and the last wall. See Figure 7 for an illustration. The lemma below shows that the sectors will appear in the same order in α as they do in the instance, up to reversion.

Lemma 8. *Let (T, b) be the result of the reduction for some instance of p-*EVEN CLIQUE *and α a b-bandwidth ordering of T such that the first wall is mapped to the leftmost elements of α. If u and v are vertices from two different sectors such that u comes before v in T, then it holds that $\alpha(u) \leq \alpha(v)$.*

Proof. If at least one of the vertices are in one of the walls, the lemma follows directly from Lemma 3. We will now consider two cases. First, we consider the case when there is a k-gate Π with center c embedded on the inner vertices of the path from u to v. We make c adjacent to $\alpha^{-1}([\alpha(c) - b, \alpha(c) + b)$ and observe that c is now the center of a wall and α is still a b-bandwidth ordering of the graph. Apply Lemma 3 on the first wall and the new wall to obtain $\alpha(u) \leq \alpha(c)$ and on the new wall and the last wall to obtain $\alpha(c) \leq \alpha(v)$. It follows immediately that $\alpha(u) \leq \alpha(v)$.

It remains to consider the case when there is no k-gate embedded on the inner vertices of the path from u to v. It follows, by construction, that either u or v is a vertex of a k-gate. First, let us consider the case when u is a vertex of a k-gate. Recall that the vertices the gate is embedded on is named $\mathit{in}, c = \mathit{center}$ and out and let P be the path from out to v. It follows by Lemmata 4 and 7 that $\alpha(P)$ and $[\alpha(c) - b, \alpha(c) + b]$ intersects in only one element, namely $\alpha(\mathit{out})$, and that $\alpha(\mathit{in}) < \alpha(c) < \alpha(\mathit{out})$. Since α is a b-bandwidth ordering it follows that $\alpha(\mathit{out}) = \min \alpha(P)$ and hence $\alpha(u) \leq \alpha(\mathit{out}) \leq \alpha(v)$. The case when v is a vertex of a k-gate follows by a symmetrical argument. □

Let P_F, P_M and P_L be the paths from the center of the first gate to the center of the last gate in the first gateland, the middle gateland and the last gateland respectively.

Lemma 9. *Let (T, b) be the result of the reduction for some instance of p-*EVEN CLIQUE *and α a b-bandwidth ordering of T, then*

- *P_F, P_M and P_L are stretched with respect to α and*
- *for the centers of two k-gates c_1 and c_2 such that c_1 comes before c_2 in T it holds that $\alpha(c_1) < \alpha(c_2)$.*

Proof. This follows directly from Lemmata 4, 7 and 8. □

Let Π_F and Π_L be the first and last k-gate in T, and c_F and c_L their centers respectively. Furthermore, let P_R be the path from c_F to c_L.

Lemma 10. *Let (T, b) be the result of the reduction for some instance of p-*EVEN CLIQUE *and α a b-bandwidth ordering of T. If $u \neq u_2$ is a vertex of a thread, such that the degree of u is at least 3, then $\alpha(u) \in I(P_R)$.*

Proof. By Lemma 3 we know that the first wall is either the leftmost or the rightmost elements of α. Observe that u is mapped within the inclusion interval of P_R by α if and only if u is mapped within the inclusion interval of P_R by α reversed. Hence it is sufficient to prove that $\alpha(u) \in I(P_R)$ when the first wall is the leftmost elements of α.

Assume for a contradiction that there is a vertex $u \neq u_2$ of some thread, such that u has degree at least 3 and $\alpha(u) \notin I(P_R)$. It follows from Lemmata 9 and 8 that either $\alpha(u) < \alpha(c_F)$ or $\alpha(c_L) < \alpha(u)$. First, we consider the case when $\alpha(u) < \alpha(c_F)$. Let P' be the path from u_2 to u except u_2 and let P'' be the path from u to the last vertex of the thread. Furthermore, let P be the path from u_2 to c_F. Assume for a contradiction that there is a k-gate Π such that P'' is not passing through Π. Let P' be the path from u_2 to the *out* vertex of Π. Observe that $\alpha(P'') \subseteq I(P')$. Recall that $|V(P'')| > b(2b+1)m_3$ and that $|E(P')| \leq b((2b+1)m_3 - 2)$. It follows that $|E(P')| \leq b((2b+1)m_3 - 2) < b(2b+1)m_3 < |V(P'')|$ and hence we get our contradiction. Hence P'' is passing through every k-gate. By Lemmata 4 and 7 we get that $\alpha(P') \subseteq I(P)$. Recall that $|V(P')| \geq (2b+1)m_1 - 1$ and that $|E(P)| = m_1$. It follows immediately that $|I(P)| \leq bm_1 + 1 < (2b+1)m_1 - 1 \leq |V(P')|$ and hence we obtain a contradiction.

It remains to consider the case when $\alpha(c_L) < \alpha(u)$. Let P be the path from u_2 to u and P' the path from u_2 to c_L except u_2. By assumption $\alpha(u_2) < \min \alpha(P')$ and hence $\alpha(P') \subseteq I(P)$. Recall that $|E(P)| < m_3$ and that $|V(P')| = (2b+1)m_3 - 3$. It follows that $|I(P)| < bm_3 + 1 < (2b+1)m_3 - 3 = |V(P')|$, which is a contradiction. □

Lemma 11. *Let (T, b) be the result of the reduction for some instance of p-*Even Clique *and α a b-bandwidth ordering of T. Then*

- $|\alpha(\tau_i) \cap I(\hat{P}_2)| = 1$ *for every thread τ_i and every subpath $\hat{P}_2$ of P_R and*
- P_R *is stretched with respect to α.*

Proof. By Lemma 3 we know that the first wall is either the leftmost or the rightmost elements of α. Observe that P_R is stretched with respect to α if and only if P_R is stretched with respect to α reversed. It follows that it is sufficient to prove that the lemma holds when the first wall is the leftmost elements of α.

Let $Z = \alpha^{-1}(I(P_R))$ and observe that there are at most $2b$ vertices in $N(Z)$. Furthermore, observe that every leaf of a gate or a hole is either within $I(P_R)$ or a neighbour of Z. It follows from Lemma 7 that Lemma 4 applies to all k-gates of T. Furthermore, by Lemma 8 it follows that the neighbours of the fillers are positioned after the first gateland and before the last gateland. And hence by Lemma 4 and the fact that α is a b-bandwidth ordering, it follows that both fillers are positioned within $I(P_R)$. By Lemma 10 it holds that for every vertex v that is a danglement, its neighbour is positioned within $I(P_R)$. And hence v is either in $I(P_R)$ or a neighbour of Z. Below you find a table giving an overview of how many vertices not on the main path, each type of gadget contributes with to $N[Z]$.

Type of vertices	Amount
Knots	$k(n+1)(\frac{3}{2}b - k - 2)$
Holes	$4n(\frac{3}{4}b - k - 2)$
First filler	$(n-k)(\frac{3}{2}b - k - 2) + (2b+1)(p(n-1)+3)$
Second filler	$(b-1)(4n+3)(2n-1) + 2b(4n+3)(2n-1) - (k(2n-1)(4n+3) + k(n(\frac{3}{2}b - k - 2) + n^2 - n - 2m) + 2n(\frac{3}{4}b - k - 2))$
k-gates	$2(b-k-1)b(m_1 + m_2 + m_3)$
$(k+1)$-gates	$2(b-k-2)((n-1)(p-3) + b(p(n-1)+3) + b(2n-1)(4n+3))$
non-neighbouring leafs	$k(n^2 - n - 2m)$

It follows from Lemma 3 that there are two vertices of the main path within $N(Z)$. Let X be all leaves in gates, holes and knots and non-neighbouring leaves and all the vertices in the fillers that are positioned within $I(P_R)$. We know that $|X|$ is at least the sum of the numbers in the table above, minus $2b-2$. And hence it can be verified that $|X| \geq (b-k-1)((2b+1)m_3 - m_1 - 2)$. By construction it follows that $|E(P_R)| = (2b+1)m_3 - m_1 - 2$. And by Lemmata 7 and 8 it follows that all threads are passing through P_R and hence we can apply Lemma 2 to complete the proof. □

Name the holes of the selector such that the first hole is called H_1 and the last hole is H_n. Let (T, b) be a resulting instance of the reduction and α a b-bandwidth ordering of T. Furthermore, let H_i be a hole of T embedded on the path (v_1, v_2, v_3, v_4) such that v_1 comes before v_4 in T. We say that a thread τ is *selecting* i, if the center c of the first knot of the thread is positioned so that $\alpha(c) \in I(v_2, v_3)$.

Lemma 12. *Let (T, b) be the result of the reduction for the instance (G, k) of p-Even Clique and α a b-bandwidth ordering of T. Then every thread in T selects a unique integer in $[n]$.*

Proof. By Lemma 3 we know that the first wall is either the leftmost or the rightmost elements of α. Observe that every thread in T selects an unique integer with respect to α if and only if every thread in T selects an unique integer with respect to α reversed. It follows that it is sufficient to prove that the lemma holds when the first wall is the leftmost elements of α.

Let us consider a thread τ with vertices $(u_2 = t_2, t_3, \dots)$, where c is the center of the first knot K of τ. Furthermore, let c_F be the center of the first gate in the first gateland, c_M the center of the last gate in the middle gateland and c_L the center of the last gate in the last gateland. We will now prove that $\alpha(c) \in I(c_F, c_M)$. We know that $\alpha(c) \in I(P_R)$ by Lemma 10 and hence in $I(c_F, c_L)$ by Lemma 11. Assume for a contradiction that $\alpha(c) \notin I(c_F, c_M)$, it follows that $\alpha(c) \in I(c_M, c_L)$. Let P be the path from u_2 to c and P' the path from u_3 to c_M. Observe that $\alpha(P') \subseteq I(P)$. Recall that $|E(P)| = (2b+1)m_1 - 1$ and that $V(P') = (2b+1)m_2 - 3$. A contradiction follows immediately, since $I(P) \leq b((2b+1)m_1 - 1) + 1 < (2b+1)m_2 - 3 \leq V(P')$. And hence we can assume $\alpha(c) \in I(c_F, c_M)$.

We will now prove that there is a hole H_i such that $\alpha(c) \in I(H_i)$. Assume for a contradiction that $\alpha(c) \notin I(H_i)$ for every i. Let $\hat{P}_2 = (p_1, p_2)$ be the P_2 of the main path such that $\alpha(c) \in I(\hat{P}_2)$. It follows by construction, that either p_1 or p_2 is the center of a gate. Observe that the leaves attached to c, p_1 and p_2 must be positioned within a $\hat{P}_4$. And due to Lemma 11 there are $4 + 3k$ vertices from the main path and the threads within $I(\hat{P}_4)$. Recall that there are $\frac{3}{2}b - k - 2$ leaves attached to c and at least $2(b - k - 2)$ leaves attached to $\hat{P}_2$. This adds up to $4 + 3k + \frac{3}{2}b - k - 2 + 2b - 2k - 4 = \frac{7}{2}b - 2 > 3b + 1$ and hence we get a contradiction.

Let H_i be embedded on the path (v_1, v_2, v_3, v_4) such that v_1 comes before v_4 in T. Observe that due to Lemma 11 there is a position within the inclusion interval of the last $(k+1)$-gate of the selector that only the first filler can take. Due to our tight budget when it comes to positions within $I(P_R)$ (see the proof of Lemma 11) it follows that the first filler must take this position. And hence for every hole in the selector, the $(k+1)$-gate immediately before and after will be passed by the first filler. It follows that Lemma 4 is applicable on the $(k+1)$-gates in the selector and hence $\alpha(K) \subseteq I(v_1, v_4)$. Furthermore, due to Lemma 11 we know that $I(H_i) \subseteq I(v_1, v_4)$. And hence we can apply Lemma 5 to obtain that $\alpha(c) \in I(v_2, v_3)$.

It remains to prove that the threads selects unique integers. Assume otherwise for a contradiction and let τ and τ' be two threads selecting the same integer i. Hence there are two knots K and K' such that $\alpha(K) \cup \alpha(K') \subseteq I(H_i)$. Observe that $I(H_i) = 3b + 1 \geq 2(\frac{3}{2}b - k - 2) + 2(\frac{3}{4}b - k - 2) = 6b - 4k - 8 > 5b$ (since there are $\frac{3}{2}b - k - 2$ leaves attached to a knot and $2(\frac{3}{4}b - k - 2)$ leaves attached to a hole) and hence we get our contradiction and the proof is complete. □

Lemma 13. *Let (T, b) be the result of the reduction for the instance (G, k) of p-Even Clique and α a b-bandwidth ordering of T. Then the set $\{v_i \mid$ there is a thread selecting $i\}$ is a clique in G.*

Proof. By Lemma 3 we know that the first wall is either the leftmost or the rightmost elements of α. Observe that the set of integers selected by the threads with respect to α is the same as the one selected with respect to α reversed. It follows that it is sufficient to prove that the lemma holds when the first wall is the leftmost elements of α.

Let A be the set of selected integers and $C = \{v_i \mid i \in A\}$. From Lemma 12 we know that the size of both A and C is k. Assume for a contradiction that there are two vertices v_a and v_b in C such that v_a and v_b are not neighbours in G. Let τ_a be the thread selecting a and τ_b the thread selecting b. One can observe that by construction and Lemma 11 there is a hole H in the validation zone and a knot K_a with center c_a embedded on τ_a such that $\alpha(c_a) \in I(H)$.

Let (v_1, v_2, v_3, v_4) be the path that H is embedded on, such that v_1 comes before v_4 in T. From Lemma 11 one can observe that there is a position within the inclusion interval of the last $(k+1)$-gate in the validator that only the second filler can take. Due to our tight budget when it comes to positions within $I(P_R)$ (see the proof of Lemma 11) it follows that the second filler must take this position. It follows that Lemma 4 is applicable on the $(k+1)$-gates immediately before and after H. Hence it follows by Lemma 5 that $\alpha(K) \cup \alpha(H) \subseteq I(v_1, v_4)$.

From the construction of T and Lemma 11 one can observe that the vertex of τ_b positioned within $I(v_2, v_3)$ has a non-neighbouring leaf attached. It follows that there are $3(k+1)+4$ vertices from the threads, the filler and the main path positioned within $I(v_1, v_4)$. Furthermore, the knot contributes with $\frac{3}{2}b - k - 2$ leaves to $I(v_1, v_4)$ and the hole with $2(\frac{3}{3}b - k - 2)$. And in addition the non-neighbouring leaf must be positioned within $I(v_1, v_4)$. It follows that $3b + 1 = |I(v_1, v_4)| \leq 3(k+1) + 4 + \frac{3}{2}b - k - 2 + 2(\frac{3}{4}b - k - 2) + 1 = 3b + 7 - 2 - 4 + 1 = 3b + 2$ which is a contradiction and the proof is complete. □

Lemma 14. *Given an instance (G, k) of p-*Clique *the reduction instance (T, b) of p-*Bandwidth *has a b-bandwidth ordering if and only if there is a clique of size k in G.*

Proof. This follows immediately by Lemmata 6, 12 and 13. □

3.6 Consequences

We will now present the immediate consequences of our reduction. But first we need to prove that the problem we have been reducing from, namely p-Even Clique is up to the task.

Lemma 15. *p-*Even Clique *is* W[1]*-hard.*

Proof. We give a simple reduction from p-Clique, which was proven to be W[1]-hard by Downey & Fellows [7]. Given an instance (G, k) of p-Clique, if k is even the instance is already a valid instance of p-Even Clique and the correctness is trivial. Otherwise, let G' be G with a universal vertex added and $k' = k + 1$. Clearly, k' is even. So this is a valid instance. If there is a clique of size k in G, then the same clique together with the universal vertex forms a clique of size k' in G'. And the other way around, if there is a clique of size k' in G'. Then there is a subset of this clique of size k not containing the added universal vertex. This is a clique in G of size k and hence our reduction is sound.

Since the reduction is parameter preserving it follows immediately that p-Even Clique is W[1]-hard. □

Lemma 16. *Assuming the Exponential Time Hypothesis p-*Even Clique *does not admit an $O(f(b)n^{o(b)})$ time algorithm.*

Proof. Observe that for the reduction in the proof of Lemma 15 is so that $k' = O(k)$. p-Clique is known to not admit an $O(f(b)n^{o(b)})$ time algorithm by Chen et. al. [3]. The result follows immediately. □

Theorem 2. *p-*Bandwidth *is* W[1]*-hard, even when the input graph is restricted to trees of pathwidth at most* 2*.*

Proof. The result follows directly from Lemmata 14 and 15 and the observations that the graph constructed by the reduction is a tree of pathwidth at most 2 and that $b = f(k)$. □

Theorem 3. *Assuming the Exponential Time Hypothesis p-*Bandwidth *does not admit an $O(f(b)n^{o(b)})$ time algorithm, even when the input graph is restricted to trees of pathwidth at most* 2*.*

Proof. The result follows directly from Lemmata 14 and 16 and the observations that the graph constructed by the reduction is a tree of pathwidth at most 2 and that $b = O(k)$. □

4 Approximation Algorithms

In this section we will provide FPT-approximation algorithms for p-Bandwidth on trees and caterpillars. Given a caterpillar T and a positive integer b, CatAlg either returns a $48b^3$-bandwidth ordering of T or correctly concludes that $bw(T) > b$. To obtain this we define an obstruction for bandwidth on caterpillars inspired by Chung & Seymour [5] and search for these objects. Based on the appearance of these objects in T we construct an interval graph such that either the interval graph has low chromatic number or the bandwidth of T is large. If the interval graph has low chromatic number we use a coloring of this graph to give a low bandwidth layout of T.

Given a tree T and positive integers b and p such that $\text{pw}(T) \leq p$, TreeAlg either returns a $(768b^3)^p$-bandwidth ordering of T or correctly concludes that $bw(T) > b$. The high level outline of the algorithm is as follows. The algorithm first decomposes the tree into several connected components of smaller pathwidth and recurses on these. Then it builds a host graph for T that is a caterpillar, applies CatAlg on the host graph. Finally it combines the result of CatAlg with the results from the recursive calls, to give a $(768b^3)^p$-bandwidth ordering of T. Since the pathwidth of a graph is known to be bounded above by its bandwidth, it follows that TreeAlg is an FPT-approximation.

4.1 An FPT-Approximation for the Bandwidth of Trees

The aim of this section is to give a FPT-approximation for p-Bandwidth on trees, namely an $(768b^3)^b$-approximation. This algorithm crucially uses a $48b^3$-approximation of p-Bandwidth on caterpillars as a subroutine. We provide such an algorithm, namely the algorithm CatAlg, in Section 4.2. In the remainder of this section we give a $(768b^3)^b$-approximation for trees under the assumption that CatAlg is a $48b^3$-approximation of p-Bandwidth on caterpillars with running time $O(bn^3)$.

Recursive Path Decompositions and Other Simplifications

In this section we will present some decomposition results crucial for our algorithm. First we define *recursive path decompositions*, which will allow us to partition our graph into several components of slightly lower complexity. The recursive decomposition is used to call the algorithm recursively on easier instances, and then combine the layouts of these instances to a low bandwidth layout of the input tree.

Definition 3. *Let T be a tree and $P, T^1, \dots, T^t$ induced subgraphs of T such that $V(T) = V(P) \cup \bigcup V(T^i)$. Then we say that $P, T^1, \dots, T^t$ is a p-*recursive path decomposition *of T if P is a path in T and for every i it holds that T^i is a connected component of $T - P$,* $\deg(V(T^i)) = 1$ *and* $\text{pw}(T^i) < p$*.*

Lemma 17. *Given a tree T of pathwidth at most p, a p-recursive path decomposition $P, T^1, \ldots, T^t$ of T can be found in $O(n)$ time.*

Proof. It was proven by Scheffler [28] that given a tree T and an integer p one can find a path decomposition $\mathcal{P}$ of T of width p or correctly conclude that $\mathrm{pw}(T) > p$ in time $O(n)$. Let X and Y be the leaf bags of $\mathcal{P}$. By standard techniques we can assume X and Y to be non-empty. Let u, v be two, not necessarily distinct, vertices such that $u \in X$ and $v \in Y$. Let P be the path in T from u to v. One can easily prove that for every bag Z of $\mathcal{P}$ it is true that $Z \cap P$ is non-empty. Hence, if we remove all the vertices of P from T and $\mathcal{P}$ we obtain a path decomposition of $T - P$ of width $p - 1$. It follows that for every connected component T^i of $T - P$ it holds that $\mathrm{pw}(T^i) \leq p - 1$. Assume for a contradiction that there is a connected component T^i such that $\deg(V(T^i)) \neq 1$. If $\deg(V(T^i)) < 1$ it follows that T was disconnected to begin with, and hence not a tree. And if $\deg(V(T^i)) > 1$ it follows that T^i together with P forms a cycle, and again T is not a tree. To complete the proof, observe that the connected components of $T - P$ can be found in $O(n)$ time by breadth first search.

□

Definition 4. *Let T be a tree and $P, T^1, \ldots, T^t$ a p-recursive path decomposition of T. We construct the* simplified instance *T_S of T with respect to $P, T^1, \ldots, T^t$ as follows. First we add P to T_S. Then, for every T^i we first add a path P^i such that $|V(P^i)| = |V(T^i)|$ and then we add an edge from one endpoint of P^i to $N(T^i)$.*

Observe that the simplified instance T_S is a caterpillar with backbone P.

Lemma 18. *Let T be a tree, $P, T^1, \ldots, T^T$ be a p-recursive path decomposition of T and T_S the corresponding simplified instance, then* $\mathrm{bw}(T_S) \leq 2\mathrm{bw}(T)$

Proof. Let α be an optimal bandwidth ordering of T. We will now give an ordering β of T_S such that $\mathrm{bw}(T_S, \beta) \leq 2\mathrm{bw}(T, \alpha)$. For every $v \in P$, let $\beta(v) = 2\alpha(v)$.

For every T^i we will consider two cases. Let $W = \alpha(T^i)$ and observe that for every $x \in W$ such that y is the smallest element in W larger than x it follows by the connectivity of T^i that $y - x \leq \mathrm{bw}(T)$. First, consider the case when at least half of W is less than $\alpha(N(T^i))$. For every $w \in W$ such that $w < \alpha(N(T^i))$, add $2w$ and $2w + 1$ to the initially empty set Z. Let $P^i = \{p_1, \ldots, p_m\}$ such that $\mathrm{dist}(P, p_j) < \mathrm{dist}(P, p_{j+1})$ for every j. For j from 1 to m, let $\beta(p_j)$ be the largest value in Z and discard $\beta(p_j)$ from Z. Observe that for every j it holds that $|\beta(p_j) - \beta(p_{j+1})|/2 \leq \mathrm{bw}(T)$. And furthermore, $|\beta(p_1) - \beta(N(P^i))| \leq \mathrm{bw}(T)$. If at least half of W is larger than $\alpha(N(T^i))$ apply a symmetric construction.

To conclude the argument we need to prove that β never maps two distinct vertices of T_S on the same position. It is easy to verify that this never happens for two vertices on P or two vertices in the same tree T^i. Consider now a vertex $u \in V(T^i)$ and a vertex $v \in V(T^j)$ for $i \neq j$. It follows that $\lfloor \beta(u)/2 \rfloor \in \alpha(T^i)$ and $\lfloor \beta(v)/2 \rfloor \in \alpha(T^j)$. Since $\alpha(T^i) \cap \alpha(T^j) = \emptyset$ it follows that $\beta(u) \neq \beta(v)$. The argument for one vertex in T^i and one in P is identical. We obtain that $\mathrm{bw}(T_S) \leq \mathrm{bw}(T_S, \beta) \leq 2\mathrm{bw}(T, \alpha) = 2\mathrm{bw}(T)$. □

Let T be a graph, v a vertex of T and α a b-bandwidth ordering of T. Let β' be a sparse ordering such that for every $u \in T$

$$\beta'(u) = \begin{cases} 2[\alpha(v) - \alpha(u)] & \text{if } \alpha(u) \leq \alpha(v) \text{ and} \\ 2[\alpha(u) - \alpha(v)] - 1 & \text{otherwise.} \end{cases}$$

and let β be the bandwidth ordering obtained by compressing β'. We then say that β is α *right folded* around v. Observe that $\mathrm{bw}(T, \beta) \leq 2\mathrm{bw}(T, \alpha)$.

Algorithm and Correctness

We are now ready to describe algorithm TreeAlg and prove its correctness. Pseudocode for TreeAlg is given in Algorithm 1.

```
Input: A tree $T$ and positive integers integers $p$ and $b$ such that $\mathrm{pw}(T) \leq p$.
Output: A $(768b^3)^p$-bandwidth ordering of $T$ or conclusion that $\mathrm{bw}(T) > b$.
if $p = 1$ then
    return CatAlg($T, b$)
end
Find a $p$-recursive path decomposition $P, T^1, \ldots, T^t$ of $T$.
Let $\alpha_1 =$ TreeAlg($T^1, p-1, b$), $\ldots, \alpha_t =$ TreeAlg($T^t, p-1, b$).
if there is an $\alpha_i = \perp$ then
    return $\perp$
end
Let $T_s$ be the simplified instance of $T$ with respect to $P, T^1, \ldots, T^t$.
Let $\alpha_s =$ CatAlg($T_s, 2b$).
if $\alpha_s = \perp$ then
    return $\perp$
end
For every $i$, let $\beta_i$ be $\alpha_i$ right folded around $N(P) \cap T^i$.
For every $v \in P$, let $\alpha(v) = \alpha_s(v)$.
For every $P_i$ of $T_s$ and every $v \in P_i$ of distance $d$ from $P$ in $T_s$, let $\alpha(\beta_i^{-1}(d)) = \alpha_s(v)$.
return $\alpha$
```

Algorithm 1: TreeAlg

Lemma 19. *Given a tree T and two integers p an b such that $\mathrm{pw}(T) \leq p$, TreeAlg terminates in $O(pbn^3)$ time.*

Proof. We start by analyzing the time complexity of the computations done in a specific execution of TreeAlg given T', p', b as input, disregarding the recursive calls. The calls to CatAlg require $O(b|V(T')|^3)$ time. Finding a p-recursive path decomposition can be done in $O(|V(T')|)$ time by Lemma 17. Constructing T'_S can trivially be done in $O(|V(T')|)$ time. And furthermore, constructing all the β's require $\sum_{i=1}^{t} O(|T^i|) = O(|V(T')|)$ time. Last, we observe that constructing α requires $O(|V(T')|)$ time. It follows that the time complexity of the computations done in a specific call to TreeAlg is $O(b|V(T')|^3)$.

Let $n = |V(T)|$ and $T_1, \ldots, T_l$ the trees given as input at a specific recursion level. Observe that $T_1, \ldots, T_l$ are pairwise disjoint and hence it follows that the time complexity of a recursion level is $\sum_{i=1}^{l} O(b|V(T_1)|^3) = O(bn^3)$. Furthermore, as p is decreased by one at each recursion level it follows that TreeAlg runs in time $O(pbn^3)$. □

Lemma 20. *Given a tree T and positive integers b and p such that $\mathrm{pw}(T) \leq p$, TreeAlg either returns a $O((768b^3)^p)$-bandwidth ordering of T or correctly concludes that $\mathrm{bw}(T) > b$ in time $O(pbn^3)$.*

Proof. The running time follows directly from Lemma 19 and hence it remains to prove the correctness of the algorithm. This we will do by induction on p. For $p = 1$ the correctness follows directly from the correctness of CatAlg and hence it remains to prove the induction step. First we consider the case when the algorithm concluded that $\mathrm{bw}(T) > b$. Either there is an α_i such that $\alpha_i = \perp$ or $\alpha_s = \perp$. If $\alpha_i = \perp$ it follows by the induction hypothesis and the fact that bandwidth is preserved on subgraphs that the algorithm concluded correctly. Now we consider the case when $\alpha_s = \perp$. It follows from the correctness of `CatAlg` that $\mathrm{bw}(T_s) > 2b$ and hence by Lemma 18 it follows that $\mathrm{bw}(T) > b$.

It remains to consider the case when the algorithm returns a bandwidth ordering α. Then, by the induction hypothesis α_i is a $(768b^3)^{p-1}$-bandwidth ordering of T^i for every i. Furthermore, α_s is a $384b^3$-bandwidth ordering for T_s, since $48(2b)^3 = 384b^3$. Let u and v be two neighbouring vertices of T. If u and v are vertices in P it follows from $\mathrm{bw}(T_s, \alpha_s) \leq 384b^3$ that $|\alpha(u) - \alpha(v)| \leq 384b^3$. Next, we consider the case when either u or v is a vertex in P. Assume without loss of generality that $u \in P$ and let T^j be such that $v \in T^j$. By the definition of β_j it follows that $\beta_j(v) = 1$. It follows that $|\alpha(u) - \alpha(v)| = |\alpha_s(u) - \alpha_s(w)|$ where $\mathrm{dist}(u, w) = 1$, and hence u and w are neighbours in T_s and it follows directly that $|\alpha(u) - \alpha(v)| \leq 384b^3$. We will now consider the case when u and v are vertices of T^j for some j. Let u' be the vertex in P^j of distance $\beta(u)$ from P and v' the vertex in P^j of distance $\beta(v)$ from P. It follows that

$$\begin{aligned}
|\alpha(u) - \alpha(v)| &= |\alpha(\beta_j^{-1}(\beta_j(u))) - \alpha(\beta_j^{-1}(\beta_j(v)))| \\
&= |\alpha_s(u') - \alpha_s(v')| \\
&\leq \mathrm{dist}(u', v')384b^3 \\
&= |\beta_j(u) - \beta_j(v)|384b^3 \\
&\leq |\alpha_j(u) - \alpha_j(v)|768b^3 \\
&\leq (768b^3)^p
\end{aligned}$$

completing the proof. □

Note that one in the case of $p = 1$ also could solve the instance exactly by Assmann [1]. It would decrease the approximation ratio to $(768b^3)^{p-1}$.

Theorem 4. *There exists an algorithm that given a tree T and a positive integer b either returns a $(768b^3)^b$-bandwidth ordering of T or correctly concludes that $\mathrm{bw}(T) > b$ in time $O(b^2n^3)$.*

Proof. This follows directly from $\mathrm{pw}(T) \leq \mathrm{bw}(T)$ and Lemma 20. □

The proof of Theorem 4 assumed the existence of a $48b^3$-approximation algorithm for caterpillars. In the next section we give such an algorithm.

4.2 An FPT-Approximation for the Bandwidth of Caterpillars

The bandwidth of caterpillars is, somewhat surprisingly, a well-studied problem. Assmann et al. [1] proved that the bandwidth of caterpillars of stray length 1 and 2 is polynomial time computable. Monien [24] completed the story of polynomial time computability by proving that Bandwidth on caterpillars of stray length 3 is NP-hard. Furtermore, Haralambides [19] gave an $O(\log n)$ approximation algorithm, which later was improved to $O(\log n/ \log\log n)$ by Feige & Talwar [12]. We now give the first FPT-approximation of p-Bandwidth on caterpillars, namely a $48b^3$-approximation.

Skewed Cantor Combs

Chung & Seymour [5] defined *Cantor combs*. These are very special caterpillars defined in such a way that they have small local density, but high bandwidth. The definition of Cantor combs is very strict - it precisely defines the length of all the paths in the caterpillars. For our purposes we need a more general definition which captures all caterpillars that are "similar enough" to Cantor combs. We call such caterpillars *skewed Cantor combs*, and we will prove that they also have high bandwidth. Our algorithm will scan for skewed Cantor combs as an obstruction for bandwidth and if none of big enough size are found it will construct a $48b^3$-bandwidth ordering based on the appearance of smaller versions of these objects.

For positive integers $k \leq b$ we now define a *skewed b-Cantor comb* of depth k, denoted $S_{b,k}$ inductively as follows. $S_{b,1}$ is a path of length 1. For the induction step to be well-defined we mark two vertices of every skewed b-Cantor comb as end vertices. For an $S_{b,1}$ the two vertices are the end vertices. For $k > 1$ we start with two skewed b-Cantor combs of depth $k-1$, lets call them S and S' and furthermore let x, y and x', y' be their end vertices respectively. Connect y to x' by a path P of length at least 2. Furthermore, let Q be a stray connected to an internal vertex v of P. Mark x and y' as the end vertices of the construction and let B be the path from x to y'. Let d be the maximum distance from v to any vertex in B. If Q has at least $2(b-1)d$ vertices we say that the graph described is a skewed b-Cantor comb of depth k.

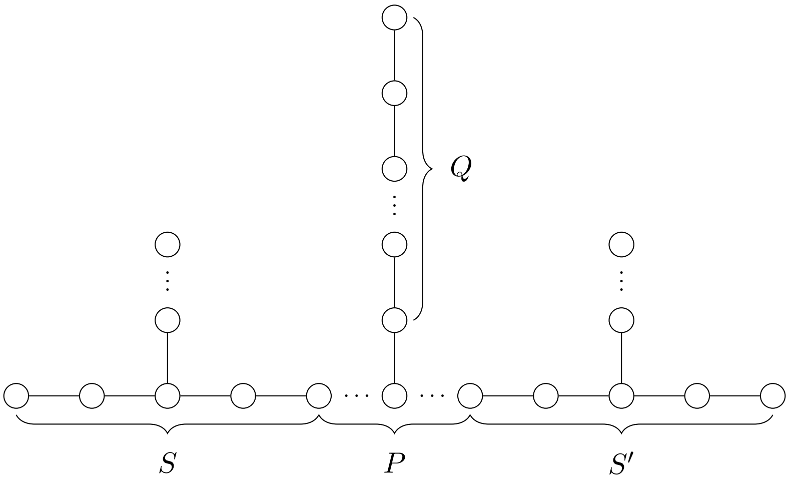

Figure 9: A skewed b-Cantor comb of depth 3 for some b.

Lemma 21. *Let $\hat{S}_{b,k}$ be a skewed b-Cantor comb of depth k and α an optimal bandwidth ordering of $\hat{S}_{b,k}$. Furthermore, let x and y be the end vertices of $\hat{S}_{b,k}$ and B the path from x to y. Then there exists an edge uv of $\hat{S}_{b,k}$ such that $I(u,v) \cap I(B)$ is non-empty and $|\alpha(u) - \alpha(v)| = \mathrm{bw}(\hat{S}_{b,k})$.*

Proof. The graph $\hat{S}_{b,k}$ is a caterpillar with backbone B. Let C_B be the connected component of $\hat{S}_{b,k}[\alpha^{-1}(I(B))]$ that contains B. Observe that $\hat{S}_{b,k} \setminus C_B$ is a collection of paths, with each path being a subpath of a stray and having exactly one neighbor in C_B.

Let L contain every vertex $u \in N(C_B)$ such that $\alpha(u) < \min[I(B)]$ and R contain every vertex $u \in N(C_B)$ such that $\alpha(u) > \max[I(B)]$. By definition we have that $L \cup (N(L) \cap C_B)$ induces a matching of size L, such that each matching edge has one endpoint u with $\alpha(u) < \min[I(B)]$ and the other endpoint v with $\alpha(v) \in \alpha(C_B)$. It follows that for one of the matching edges $|\alpha(u) - \alpha(v)| \geq |L|$. Thus there exists an edge uv of $\hat{S}_{b,k}$ such that $I(u,v) \cap I(B)$ is non-empty and $|\alpha(u) - \alpha(v)| \geq |L|$. An identical argument yields that there exists an edge uv of $\hat{S}_{b,k}$ such that $I(u,v) \cap I(B)$ is non-empty and $|\alpha(u) - \alpha(v)| \geq |R|$. Thus there exists an edge $u'y'$ such that $I(u', v') \cap I(B)$ is non-empty and $|\alpha(u') - \alpha(v')| = \max(|L|, |R|)$.

We now prove that without loss of generality, we can assume that every edge uv such that neither u nor v are in C_B satisfies $|\alpha(u) - \alpha(v)| \leq \max(|L|, |R|)$. Let C_L be the set of vertices connected to L in $G - C_B$ and C_R the set of vertices connected to R in $G - C_B$. Observe that C_B, C_L and C_R form a partition of $V(\hat{S}_{b,k})$. For every $v \in C_B \cup L \cup R$ let $\beta(v) = \alpha(v)$. Let v be a vertex of $C_L \setminus L$ and u the unique vertex of L such that u and v are connected in $G - C_B$. We then let $\beta(v) = \beta(u) - |L| \cdot \mathrm{dist}(u,v)$. Handle the vertices of $C_R \setminus R$ symmetrically and let β' be the compressed β. One can observe that β' is a linear ordering of $\hat{S}_{b,k}$ and that $\mathrm{bw}(\hat{S}_{b,k}, \beta') \leq \mathrm{bw}(\hat{S}_{b,k}, \alpha) = \mathrm{bw}(\hat{S}_{b,k})$. Clearly, for every edge uv such that neither u nor v are in C_B satisfies $|\alpha(u) - \alpha(v)| \leq \max(|L|, |R|)$.

Let uv be an edge of $\hat{S}_{b,k}$ such that $|\alpha(u) - \alpha(v)| = \mathrm{bw}(\hat{S}_{b,k})$. If one endpoint of uv is mapped to $I(B)$ we are done, as uv satisfies the conditions of the lemma. On the other hand, if both endpoints of uv are outside of $I(B)$ then $\mathrm{bw}(\hat{S}_{b,k}) = |\alpha(u) - \alpha(v)| = \max(|L|, |R|)$. In this case the edge $u'v'$ satisfies the conditions of the lemma, completing the proof. □

Lemma 22. *For $b \geq k \geq 1$, the bandwidth of any $S_{b,k}$ is at least k.*

Proof. The proof of this lemma is inspired by the one for Cantor combs given by Chung and Seymour ([6], Lemma 2.1).

Assume for a contradiction that there is a $\hat{S}_{b,k}$ such that $\mathrm{bw}(\hat{S}_{b,k}) < k$. Furthermore, assume without loss of generality that k is the smallest such value with respect to b. Observe that $k > 1$. Let α be an ordering of $\hat{S}_{b,k}$ of bandwidth at most $k-1$. Let S, S', P and Q be as in the definition of skewed Cantor combs. By assumption the bandwidth of both S and S' are $k-1$. Let x and y be the end vertices of S and x' and y' the end vertices of S'. Furthermore, let B be the path from x to y and B' the path from x' to y'. Let Z be the path between the end vertices of $\hat{S}_{b,k}$.

Let β be the compressed version of α when restricted to S. Since α is of bandwidth $k-1$, it follows that β is of bandwidth at most $k-1$ and hence by our assumption β is an optimal bandwidth ordering of S. By Lemma 21 we know that there exists an edge uv in S such that $I_\beta(u,v) \cap I_\beta(B)$ is non-empty and $|\beta(u) - \beta(v)| = k-1$. It follows that $I_\alpha(u,v) \cap I_\alpha(B)$ is non-empty and $|\alpha(u) - \alpha(v)| = k-1$. In the same manner we obtain an edge $u'v'$ from S'. Assume without loss of generality that $\alpha(u) < \alpha(v)$ and that $\alpha(u') < \alpha(v')$.

Observe that $\alpha^{-1}(I_\alpha(u,v)) \subseteq S$ and that $\alpha^{-1}(I_\alpha(u',v')) \subseteq S'$. It follows directly that the inclusion intervals has an empty intersection with P. Let q be the vertex in $N(Q)$. We can assume without loss of generality that $\alpha(v) < \alpha(q)$. There are two cases to consider, either $\alpha(q) < \alpha(u')$ or $\alpha(v') < \alpha(q)$.

First we consider the case when $\alpha(q) < \alpha(u')$. Observe that $|I(Z)| \leq (k-1)|E(Z)| + 1 \leq |V(Q)| + 1$ and $|V(Z)| \geq 5$ since $k > 1$. It follows from $\alpha(Z) \subseteq I(Z)$ that there is a vertex $q' \in Q$ such that $\alpha(q') \notin I(Z)$. Assume without loss of generality that $\alpha(q') < \min I(Z)$. It follows that $\alpha(q') < \alpha(u) < \alpha(v) < \alpha(q)$. Since there is a path from q' to q disjoint from S and $|\alpha(u) - \alpha(v)| = k-1$ it follows that $I(u,v)$ must contain a vertex of Q, which is a contradiction.

It remains to consider the case when $\alpha(v') < \alpha(q)$. Observe that by assumption $I(u,v)$ and $I(u',v')$ are disjoint. And hence, again we consider two cases. First, let $\alpha(v) < \alpha(u')$. We are then in the situation that $\alpha(v) < \alpha(u') < \alpha(v') < \alpha(q)$ and since there is a path from v to q avoiding S' it follows that this path has a non-empty intersection with $I(u',v')$, which is a contradiction. The case $\alpha(v') < \alpha(u)$ follows by a symmetric argument and hence the proof is complete. □

Directions

Given a caterpillar T and a backbone $B = \{b_1, \ldots, b_k\}$ we define *pos*(P) for every stray P in T with respect to B, as the integer i such that P is attached to the vertex b_i. Furthermore, we let $|P|$ denote $|V(P)|$.

Definition 5. *Let T be a caterpillar, $B = \{b_1, \ldots, b_k\}$ a backbone of T and b a positive integer. Furthermore, let* depth *be a function from the strays of T with respect to B to $\mathbb{N}$. For every stray Q we let*

- $X_Q = \left\{P \mid \mathit{pos}(P) + \frac{|P|}{2b} < \mathit{pos}(Q) \text{ and } \mathit{pos}(Q) - \frac{|Q|}{2b} \leq \mathit{pos}(P) - \frac{|P|}{2b}\right\}$ *and*

- $Y_Q = \left\{ P \mid pos(Q) < pos(P) - \frac{|P|}{2b} \text{ and } pos(P) + \frac{|P|}{2b} \leq pos(Q) + \frac{|Q|}{2b} \right\}$.

Let $x_Q = \max(\text{depth}(X_Q))$ *and* $y_Q = \max(\text{depth}(Y_Q))$. *We say that* Q *is* pushed east *if* $x_Q > y_Q$, pushed west *if* $x_Q < y_Q$ *and* lifted *if* $x_Q = y_Q$.

We say that a skewed b-Cantor comb of depth k is centered around the stray Q, where Q is as in the definition of $S_{b,k}$. For a caterpillar T we say that a *backbone* B *is maximized* if for every other backbone B' it holds that $|B'| \leq |B|$.

We will now describe an algorithm `FindSCC` that given a caterpillar T, a maximized backbone B of T and a positive integer b searches for skewed Cantor combs in T. Let *depth* be a function from the strays of T with respect to B into $\mathbb{N}$. As an invariant, *depth* promises there to be a skewed $(b+1)$-Cantor comb centered around Q of depth *depth*(Q). The exception is if *depth*(Q) is 0, then the stray is so short that we ignore it and we hence make no promises with respect to skewed $(b+1)$-Cantor combs. Initially, for every stray Q let *depth*(Q) be 2 if $|Q| \geq 4b$ and 0 otherwise. Observe that the invariant is true due to B being a maximized backbone.

Now we search for a stray Q that is lifted such that both x_Q and y_Q are at least *depth*(Q). It such a Q is found, increase *depth*(Q) by one. Observe that there is in fact a skewed $(b+1)$-Cantor comb centered around Q of this depth (*depth*(Q) after the incrementing). Run this procedure until such a stray Q can not be found or until *depth*(Q) reaches $b+1$ for some stray. Observe that we can for every stray evaluate x_Q and y_Q in $O(n^2)$. And since this is done at most $O(bn)$ times, the running time of `FindSCC` is bounded by $O(bn^3)$.

The reader should note that `FindSCC` does not detect all skewed b-Cantor combs. In fact, it searches only for a stricter version and might overlook the deep skewed $(b+1)$-Cantor combs in a caterpillar. But, as it turns out, these stricter versions are sufficient for our purposes. From now on, we will assume that the function applied when evaluation whether a stray is pushed west or east, is the depth function calculated by running `FindSCC`.

Definition 6. *For a caterpillar* T, *a maximized backbone* $B = \{b_1, \ldots, b_l\}$ *of* T *and a positive integer* b *we define the* directional stray graph *as the following interval graph: for every stray* P *add the interval*

- $[pos(P)48b^3 - 12b^2|P|, pos(P)48b^3]$ *if* P *is pushed west and*
- $[pos(P)48b^3, pos(P)48b^3 + 12b^2|P|]$ *otherwise.*

We say that an interval originating from a stray pushed west is west oriented *and visa versa.*

Lemma 23. *Let* T *be a caterpillar,* b *a positive integer,* G_I *some directional stray graph of* T *and* x *and* y *two natural numbers such that* $x < y$. *Then either there are at most* $2b$ *intervals of length at least* $y - x$ *in* G_I *starting within* $[x, y]$, *or* $\text{bw}(T) > b$.

Proof. Assume otherwise for a contradiction and let $\text{bw}(T) \leq b$ and K be a set of $2b+1$ intervals of length at least $y - x$ starting within $[x, y]$. Let x' be the smallest number such that $x \leq x'$ and x' is divisible by $48b^3$ and y' the largest number such that $y' \leq y$ and y' is divisible by $48b^3$. Observe that all intervals in K has their starting point within $[x', y']$ by construction. Consider the minimum connected, induced subgraph H of T containing the vertices of the strays corresponding to the intervals in K. We will consider H with respect to the backbone such that the strays of H are exactly the ones corresponding to intervals in K. Let $z = y' - x'$ and observe that every stray in H contains at least $q = z/12b^2$ vertices and that the backbone of H is of length $r = z/48b^3$. It follows that

$$\begin{aligned} D(G) &\geq \frac{|V(H)|-1}{\mathrm{diam}(H)} \\ &\geq \frac{(2b+1)q+r-1}{2q+r} \\ &> \frac{2bq+r}{2q+r} \\ &\geq b\frac{2q+r/b}{2q/b+r/b} \\ &\geq b \end{aligned}$$

which contradicts $D(G) \leq b$ and hence we know that there are at most $2b$ such intervals. Note that we used the fact that $q > 1$. This follows from the fact that $x' < y'$ due to the local density bound and hence $q \geq 48b^3/12b^2 \geq 4$.

□

Lemma 24. *Let T be a caterpillar, b a positive integer and G_I some directional stray graph of T. Then either $\chi(G_I) < 12b^2$ or $\mathrm{bw}(T) > b$.*

Proof. Assume for a contradiction that $\chi(G_I) \geq 12b^2$ and that $\mathrm{bw}(T) \leq b$. Then there is a number w such that at least $12b^2$ of the intervals of G_I contains w. This follows from the well-known result that $\chi(G_I)$ equals the size of the maximum clique of G_I, since G_I is an interval graph. Let I be the set of all east oriented intervals containing w and assume without loss of generality that I is of size at least $6b^2$. Discard the elements of I with the highest starting value and let $[x', y']$ be a discarded element. Observe that at most $2b$ elements were discarded due to the local density bound. Hence we now have at least $6b^2 - 2b$ elements left. We will start by giving a lower bound on the length of the intervals in I. Consider an element $[x, y]$ of shortest length in I. By definition $x < x' \leq y$ and by construction $x' - x \geq 48b^3$, hence $y - x \geq 48b^3$ and it follows that all elements of I are of length at least $48b^3$.

Let $[x_2, y_2]$ be a shortest interval in I and recall that the stray P^2 corresponding to the interval is attached to the backbone vertex b_{c_2} for $c_2 = x_2/48b^3$. Furthermore, $|P^2| = (y_2 - x_2)/12b^2 \geq 48b^3/12b^2 = 4b$. Since the backbone used when constructing G_I is maximized it follows that the distance from b_{c_2} to any endpoint of the backbone is at least 4 and hence there is an $S_{b+1,2}$ centered around b_{c_2}.

Discard all intervals with their starting point within $[x_2 - 2(y_2 - x_2), y_2]$ in I. We know that at most $6b$ elements are discarded by Lemma 23. Now let $[x_3, y_3]$ be a shortest interval in I and recall that the stray P^3 corresponding to the interval is attached to the backbone vertex b_{c_3} for $c_3 = x_3/(12b^2)$. Observe that $|y_3 - x_3| > |x_3 - x_2|$ and that $|y_2 - x_2| < \frac{1}{2}|x_3 - x_2|$ and hence

$$\begin{gathered} |y_3 - x_3| > |x_3 - x_2| > \frac{1}{2}|x_3 - x_2| + |y_2 - x_2| \\ \Longrightarrow \\ \frac{|y_3 - x_3|}{2b(12b^2)} > \frac{|x_3 - x_2|}{48b^3} + \frac{|y_2 - x_2|}{2b(12b^2)} \\ \Longrightarrow \\ \frac{|V(P^3)|}{2b} > |c_3 - c_2| + \frac{|V(P^2)|}{2b}. \end{gathered}$$

Let S be the $S_{b+1,2}$ centered around b_{c_2} and recall that by definition the distance from b_{c_2} to any backbone vertex of S is bounded from above by $\frac{|V(P^2)|}{2b}$. It follows that the distance from b_{c_3}

to any backbone vertex of S is bounded by $\frac{|V(P^3)|}{2b}$. Since $[x_3, y_3]$ is east oriented there is another $S_{b+1,i}$ centered around a stray $\bar{P}^2$ such that $\mathrm{pos}(\bar{P}^2) + \frac{|\bar{P}^2|}{2b} < c_3$ and $\mathrm{pos}(\bar{P}^2) - \frac{|\bar{P}^2|}{2b} \geq c_3 - \frac{|P^3|}{2b}$ for some $i \geq 2$. By definition, the $S_{b+1,i}$ contains an $S_{b+1,2}$ as a subgraph in such a way that there is an $S_{b+1,3}$ centered around c_3. Discard all intervals with starting points within $[x_3 - 2(y_3 - x_3), x_3]$ and repeat the argument to obtain a $S_{b+1,4}$. We keep repeating the argument until we obtain a $S_{b+1,b+1}$

Notice that we can do this as we are discarding at most $6b$ vertices each time, repeating the procedure $b-1$ times and I contains at least $6b^2 - 2b > 6b(b-1)$ intervals. This completes the proof, as we know from Lemma 22 that $\mathrm{bw}(S_{b+1,b+1}) \geq b + 1$. □

Algorithm and Correctness

```
Input: A caterpillar $T$ and a positive integer $b$.
Output: A $48b^3$-bandwidth ordering of $T$ or conclusion that $\mathrm{bw}(T) > b$.
Let $B = \{b_1, \ldots, b_k\}$ be a maximized backbone of $T$.
Construct the directional stray graph $G_I$ of $T$ with respect to $B$.
Find a minimum coloring of $G_I$.
if $\chi(G_I) \geq 12b^2$ then
    return $\perp$.
end
Let $\alpha(b_i) = 48b^3(n + i)$.
Let $\mathcal{P}$ be the collection of strays in $T$ with respect to $B$.
For every stray $P$ in $\mathcal{P}$ let $C(P)$ be the color of the interval representing the stray.
for every $P \in \mathcal{P}$ do
    Let $p_1, \ldots, p_k$ be the vertices of $P$ such that $\mathrm{dist}(B, p_i) < \mathrm{dist}(B, p_{i+1})$ for every $i$.
    Let $\{u\} = N(P)$.
    if $P$ is pushed west then
        Let $\alpha(p_i) = \alpha(u) + C(P) - i12b^2$ for every $i$.
    end
    else
        Let $\alpha(p_i) = \alpha(u) + C(P) + (i - 1)12b^2$ for every $i$.
    end
end
return Compressed version of $\alpha$.
```

Algorithm 2: `CatAlg`

Theorem 5. *There exists an algorithm that given a caterpillar T and a positive integer b either returns a $48b^3$-bandwidth ordering of T or correctly concludes that* $\mathrm{bw}(T) > b$ *in time $O(bn^3)$.*

Proof. Recall that `FindSCC` runs in $O(bn^3)$ time. Furthermore, a coloring of G_I can be found in $O(n)$ time by Golumbic [17]. Observe that every other step of the algorithm trivially runs in $O(n)$ time. And hence the algorithm runs in $O(bn^3)$ time. If `CatAlg` returns $\perp$, then $\chi(G_I) \geq 12b^2$. It follows from Lemma 24 that $\mathrm{bw}(T) > b$ and hence the conclusion is correct. We will now prove that α is a sparse ordering of $V(T)$ of bandwidth at most $48b^3$. It is clear that for any edge $uv \in E(T)$ it holds that $|\alpha(u) - \alpha(v)| \leq 48b^3$. It remains to prove that α is an injective function. Assume for a contradiction that there are two vertices u, v such that $\alpha(u) = \alpha(v)$. Observe that $\alpha(u) \equiv 0 \bmod (48b^3)$ if and only if u is a backbone vertex of T. This comes from the fact that $\chi(G_I) < 12b^2$. And since it is clear from the algorithm that no two vertices of the backbone are given the same position we can assume that neither u nor v is a backbone vertex. It follows that $\alpha(u) \equiv c(P) \bmod (12b^2)$ where P is the stray containing u. Observe that the algorithm gives unique positions to all vertices from the same stray and hence u and v must belong to two different strays given the same color. Let P_u be the stray containing u and P_v the strain containing v. Furthermore, let $[x_u, y_u]$ and $[x_v, y_v]$ be the corresponding intervals in G_I. Observe that $I(P_u) \subseteq [x_u, y_u]$ and $I(P_v) \subseteq [x_v, y_v]$ and hence $[x_u, y_u] \cap [x_y, y_v] \neq \emptyset$, which is a contradiction, completing the proof. □

5 Concluding Remarks

We have shown that the classical $2^{O(b)}n^{b+1}$ time algorithm of Saxe [27] for the Bandwidth problem is essentially optimal, even on trees of pathwidth at most 2. On trees of pathwidth 1, namely caterpillars with hair length 1, the problem is known to be polynomial time solvable. On the positive side, we gave the first approximation algorithm for Bandwidth on trees with approximation ratio being a function of b and independent of n. Our approximation algorithm is based on pathwidth, local density and a new obstruction to bounded bandwidth called skewed Cantor combs. We conclude with a few open problems.

- Does Bandwidth admit a parameterized approximation algorithm on general graphs?
- Does Bandwidth admit an approximation algorithm on trees with approximation ratio polynomial in b? What if one allows the algorithm to have running time $f(b)n^{O(1)}$?
- Does there exist a function f such that any graph G with pathwidth at most c_1, local density at most c_2, and containing no S_{c_3,c_3} as a subgraph has bandwidth at most $f(c_1, c_2, c_3)$?

References

[1] S. Assmann, G. Peck, M. Syslo, and J. Zak, *The bandwidth of caterpillars with hairs of length 1 and 2*, SIAM Journal on Algebraic Discrete Methods, 2 (1981), pp. 387–393.

[2] H. L. Bodlaender, M. R. Fellows, and M. T. Hallett, *Beyond np-completeness for problems of bounded width: hardness for the w hierarchy*, in STOC, 1994, pp. 449–458.

[3] J. Chen, X. Huang, I. A. Kanj, and G. Xia, *Strong computational lower bounds via parameterized complexity*, Journal of Computer and System Sciences, 72 (2006), pp. 1346–1367.

[4] P. Z. Chinn, J. Chvtalov, A. K. Dewdney, and N. E. Gibbs, *The bandwidth problem for graphs and matricesa survey*, Journal of Graph Theory, 6 (1982), pp. 223–254.

[5] F. R. Chung and P. D. Seymour, *Graphs with small bandwidth and cutwidth*, Discrete Mathematics, 75 (1989), pp. 113–119.

[6] F. R. K. Chung and P. D. Seymour, *Graphs with small bandwidth and cutwidth*, Discrete Mathematics, 75 (1989), pp. 113–119.

[7] R. G. Downey and M. R. Fellows, *Fixed-parameter tractability and completeness ii: On completeness for w[1]*, Theoretical Computer Science, 141 (1995), pp. 109–131.

[8] R. G. Downey and M. R. Fellows, *Parameterized complexity*, vol. 3, Springer, 1999.

[9] C. Dubey, U. Feige, and W. Unger, *Hardness results for approximating the bandwidth*, Journal of Computer and System Sciences, 77 (2011), pp. 62–90.

[10] J. Dunagan and S. Vempala, *On euclidean embeddings and bandwidth minimization*, in RANDOM-APPROX, 2001, pp. 229–240.

[11] U. Feige, *Approximating the bandwidth via volume respecting embeddings*, J. Comput. Syst. Sci., 60 (2000), pp. 510–539.

[12] U. Feige and K. Talwar, *Approximating the bandwidth of caterpillars*, Algorithmica, 55 (2009), pp. 190–204.

[13] J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag New York, Inc., 2006.

[14] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.

[15] A. George and J. W. Liu, *Computer Solution of Large Sparse Positive Definite*, Prentice Hall Professional Technical Reference, 1981.

[16] P. A. Golovach, P. Heggernes, D. Kratsch, D. Lokshtanov, D. Meister, and S. Saurabh, *Bandwidth on at-free graphs*, Theor. Comput. Sci., 412 (2011), pp. 7001–7008.

[17] M. C. Golumbic, *Algorithmic graph theory and perfect graphs*, vol. 57, Elsevier, 2004.

[18] A. Gupta, *Improved bandwidth approximation for trees*, in SODA, 2000, pp. 788–793.

[19] J. Haralambides, F. Makedon, and B. Monien, *Bandwidth minimization: an approximation algorithm for caterpillars*, Mathematical Systems Theory, 24 (1991), pp. 169–177.

[20] P. Heggernes, D. Kratsch, and D. Meister, *Bandwidth of bipartite permutation graphs in polynomial time*, J. Discrete Algorithms, 7 (2009), pp. 533–544.

[21] R. Impagliazzo, R. Paturi, and F. Zane, *Which problems have strongly exponential complexity?*, J. Comput. Syst. Sci., 63 (2001), pp. 512–530.

[22] D. J. Kleitman and R. V. Vohra, *Computing the bandwidth of interval graphs*, SIAM Journal on Discrete Mathematics, 3 (1990), pp. 373–375.

[23] D. Marx, *Parameterized complexity and approximation algorithms*, Comput. J., 51 (2008), pp. 60–78.

[24] B. Monien, *The bandwidth minimization problem for caterpillars with hair length 3 is np-complete*, SIAM Journal on Algebraic Discrete Methods, 7 (1986), pp. 505–512.

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

[26] C. H. Papadimitriou, *The np-completeness of the bandwidth minimization problem*, Computing, 16 (1976), pp. 263–270.

[27] J. B. Saxe, *Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time*, SIAM Journal on Algebraic Discrete Methods, 1 (1980), pp. 363–369.

[28] P. Scheffler, *A linear algorithm for the pathwidth of trees*, in Topics in combinatorics and graph theory, Springer, 1990, pp. 613–620.

[29] J.-H. Yan, *The bandwidth problem in cographs*, Tamsui Oxford Journal of Mathematical Sciences, (1997), pp. 31–36.