The Complexity of the Approximation of the Bandwidth Problem (Extended Abstract)

Walter Unger
Lehrstuhl für Informatik I
RWTH Aachen
Ahornstraße 55
D-52074 Aachen
Germany
quacks@i1.Informatik.RWTH-Aachen.de

Abstract

The bandwidth problem has a long history and a number of important applications. It is the problem of enumerating the vertices of a given graph G such that the maximum difference between the numbers of adjacent vertices is minimal. We will show for any constant $k \in \mathbb{N}$ that there is no polynomial time approximation algorithm with an approximation factor of k. Furthermore, we will show that this result holds also for caterpillars, a class of restricted trees. We construct for any $x, \varepsilon \in \mathbb{R}$ with $x > 1$ and $\varepsilon > 0$ a graph class for which an approximation algorithm with an approximation factor of $x + \varepsilon$ exists, but the approximation of the bandwidth problem within a factor of $x - \varepsilon$ is NP-complete. The best previously known approximation factors for the intractability of the bandwidth approximation problem were 1.5 for general graphs and $4/3$ for trees.

1. Introduction

The bandwidth problem has a long history. It was first considered by mathematicians as the bandwidth problem of a matrix M. To operate efficiently with a sparse, symmetric matrix it is important to permutate the columns and lines of the matrix with the same permutation such that the non-zero entries are near the diagonal of the matrix. These two problems are equivalent. The graph G is modeled by an adjacency matrix M', where the 'ones' coincide with the non-zero entries of the given matrix M.

An other equivalent problem is the embedding of one graph G with n vertices into the path with n vertices with minimal dilation and load 1. For an introduction to the embedding problems see [12]. We note that the bandwidth problem on graphs has a number of practical applications [2].

The bandwidth problem — the problem to decide for a given graph G and $b \in \mathbb{N}$ if the bandwidth of G is less or equal b — is NP-complete [13]. It remains NP-complete if it is restricted to trees of degree three [5] and even if it is restricted to caterpillars of hairlength three or to caterpillars of degree three [11]. A caterpillar is a tree where all vertices of degree 3 and larger lie on one path, called the backbone. The hairlength of a caterpillar is the maximum over all minimal distances between the non-backbone vertices and the vertices of the backbone. For more than ten years no further intractability results about the bandwidth problem were presented.

There are just a few cases where the bandwidth of a graph is constructable in polynomial time [5, 14, 3, 16]. A polynomial time approximation algorithm for caterpillars with an approximation factor of $\log n$ is presented in [6]. For general graphs [4] presented a polynomial time approximation algorithm with a polylogarithmic approximation factor. An approximation algorithm for δ-dense graphs with an approximation factor of three is given in [9]. A graph G with n vertices is called δ-dense iff the minimum degree of G is at least δn.

In [1] the intractability of the bandwidth approximation problem with an approximation factor of 1.5 for general graphs and $4/3$ for trees is presented. Thus there exists no

PTAS (polynomial time approximation schemes) in these cases (see also [8]).

We will show that the bandwidth approximation problem is intractable (NP-complete) for any factor $k \in I\!N$. This result holds even for the case where the input is restricted to caterpillars. We construct for any $x, \varepsilon \in I\!R$ with $x > 1$ and $\varepsilon > 0$ a graph class for which an approximation algorithm with an approximation factor of $x + \varepsilon$ exists, but the approximation of the bandwidth problem within a factor of $x - \varepsilon$ is NP-complete. The reduction is done by component design from the exact 3-satisfiability problem [15].

This considerable improvement is based on two facts. In a first step we introduce 'blockers' as the base component of our reduction. Two blockers will 'block' each other in a labeling with bandwidth between b and $2b - \varepsilon$. I.e. there is no labeling with a bandwidth within the range b and $2b - \varepsilon$ where the minimal and maximal label are put on vertices of the same blocker. These blockers enable us to show that the bandwidth approximation problem is NP-complete for an approximation factor of $2 - \varepsilon$. In a second step we use a 'recursive' reduction to prove that the bandwidth is intractable for any approximation factor $k \in I\!N$. This recursive reduction contains components which on their own simulate the boolean formula from the exact 3-satisfiability problem.

This paper is structured as follows. Section 2 presents the used formal definitions. The basic components called 'blocker' and 'breaker' used in the reduction are constructed in Section 3. The intractability results with an approximation factor of less than two are presented in Section 4. Results with arbitrary approximation factors are given in Section 5.

2. Definitions

We start with the formal definitions of the bandwidth of a graph. Let $G = (V, E)$ be a graph, $W \subset V$ and let $v, v' \in V$. A *labeling* of G is a mapping $e : V \to I\!N$ with $e(v) = e(v') \Rightarrow v = v'$. We denote the distance between v and v' in the labeling e by $\mathrm{dist}_e(v, v') = |e(v) - e(v')|$. The *bandwidth* $\mathrm{bw}_e(G)$ of a labeling e of G is $\max\{\mathrm{dist}_e(v, v') \mid \{v, v'\} \in E\}$. The *bandwidth* $\mathrm{bw}(G)$ of a graph G is $\min_{e \text{ is a labeling of } G}\{\mathrm{bw}_e(G)\}$.

The graphs we are considering are connected, not directed, without self-loops and multiple edges. We may assume this wlog. when considering the bandwidth of graphs. For the standard graph theoretic definitions see also [7].

We are using the following notations. Let $G = (V, E)$ and $G' = (V', E')$ be graphs. Furthermore let $v, w \in V$. The set of neighbors of v is given by $\mathrm{N}_G(v) = \{w \in V \mid \{v, w\} \in E\}$. The degree of vertice v is $\deg_G(v) = |\mathrm{N}_G(v)|$ and the degree of G is $\deg(G) = \max_{w \in V}\{\deg_G(w) \quad |\}$. If the graph G is determined uniquely by the context we use the abbreviations $\mathrm{N}(v) = \mathrm{N}_G(v)$ and $\deg(v) = \deg_G(v)$.

G restricted to a set W of vertices is denoted by $G|W = (W, \{e \in E \mid e \subset W\})$. We denote the deletion of vertice v in G by $G \setminus v = G|(V \setminus \{v\})$. With $G + G' = (V \cup V', E \cup E')$ we denote the union of two graphs. In this case we always assume that $V \cap V' = \varnothing$ holds. In the next notation the vertices v and w with $v \neq w$ are joined to become one vertice in the resulting graph. $\mathcal{M}_v^w(G) = (V \setminus \{w\}, \{\{v, v'\} \mid \{v, v'\} \in E \wedge v' \neq w\} \cup \{\{v, v'\} \mid \{w, v'\} \in E \wedge v \neq v'\})$.

Two graphs are considered to be 'equal' if they are isomorphic. We say that G and G' are *isomorphic* — denoted by $G \cong G'$ — iff there exists a one-to-one function $e : V \to V'$ with $\{v, w\} \in E \iff \{e(v), e(w)\} \in E'$. G' is a vertex induced subgraph of G iff $V' \subset V$ and $G' = G|V'$. In this case we call G' the subgraph of G induced by V'. We will use in the following the short term 'subgraph'.

The reduction is done by component design. The components (resp. subgraphs of the construction) are joined in linear (resp. cyclic) fashion (see Section 1). Each component G has two special vertices called $\mathrm{Left}(G)$ and $\mathrm{Right}(G)$. Two components G, G' form a new component $H = \mathcal{M}_{\mathrm{Right}(G)}^{\mathrm{Left}(G')}(G + G')$ with $\mathrm{Left}(H) = \mathrm{Left}(G)$ and $\mathrm{Right}(H) = \mathrm{Right}(G')$. To ease the description we use the following notations for components G and G_i. $G^1 = G$ and $G^{i+1} = \mathcal{M}_{\mathrm{Right}(G)}^{\mathrm{Left}(G^i)}(G + G^i)$ for $i \geqslant 1$. By $G_1 G_2 G_3 \ldots G_i$ we denote $\mathcal{M}_{\mathrm{Right}(G_1)}^{\mathrm{Left}(G_2 G_3 \ldots G_i)}(G_1 + G_2 G_3 \ldots G_i)$ and by ${G_1 G_2 G_3 \ldots G_i}^{\mathrm{R}}$ we denote $G_i \ldots G_3 G_2 G_1$.

For these components we introduce a special type of labeling and bandwidth, called the stretched labeling (resp. bandwidth). A labeling e of G is called stretched iff $\forall w \in V : \min(e(\mathrm{Left}(G)), e(\mathrm{Right}(G))) \leqslant e(w) \leqslant \max(e(\mathrm{Left}(G)), e(\mathrm{Right}(G)))$. The *stretched-bandwidth* of G — denoted by $\mathrm{sbw}(G)$ — is $\min\{\mathrm{bw}_e(G) \mid e \text{ is a stretched labeling of } G\}$.

Based on the next definition we will construct in the following the first components of our reduction. By $\mathrm{Path}(G)$ we denote all vertices $\mathrm{Left}(G')$, $\mathrm{Right}(G')$, where G' is a subcomponent in G. We say G *spans* H *in the labeling* e of $G + H$ iff $\forall w \in \{\mathrm{Left}(H), \mathrm{Right}(H)\} : e(\mathrm{Left}(G)) < e(w) < e(\mathrm{Right}(G))$. We say G *spans* H *with bandwidth* b iff there exists a labeling e with $\mathrm{bw}_e(G + H) = b$ and

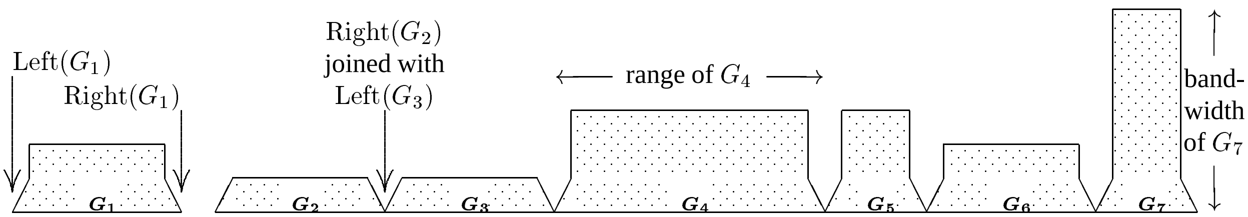

Figure 1. Combining components using special vertices

G spans H in the labeling e. Let G be a component of our reductions with $|V(G)| = n$. G is *blocking* with bandwidth b, iff for all stretched labelings e with $\mathrm{bw}_e(G) = b$ holds $\mathrm{dist}_e(\mathrm{Left}(G), \mathrm{Right}(G)) < 2 \cdot (n-1)$. G has *stretch-factor* of s — denoted by $\mathrm{stretch}(G)$ —, iff $\forall b$ ($\mathrm{sbw}(G) \leqslant b \leqslant s \cdot \mathrm{sbw}(G)$) G is blocking with bandwidth b. Let $\mathrm{Path}(G) = (v_1, v_2, v_3, ..., v_m)$. We say G is *not folded* in labeling e iff $(e(v_1), e(v_2), ..., e(v_m))$ is monotonous, otherwise G is called *folded*.

Finally we give a formal definition of the bandwidth approximation problem and exact 3-satisfiability problem. A boolean formula $\mathbb{F}$ with variables $v_1, v_2, ..., v_n$ is in conjunctive normal form (CNF) with clauses of length 3 iff the following properties hold: $\mathbb{F} = \bigwedge_{i=1}^{m}(l_{1,i} \vee l_{2,i} \vee l_{3,i})$ and $\forall i, j, 1 \leqslant i \leqslant m, 1 \leqslant j \leqslant 3 : \exists k : l_{j,i} = v_k$ or $l_{j,i} = \neg v_k$. The terms of the form $(l_{1,i} \vee l_{2,i} \vee l_{3,i})$ are called clauses and the terms of the form $l_{j,i}$ are called literals. We say that $\mathbb{F}$ consists of m clauses and has n variables. The *exact 3-satisfiability problem* is the following: Does for a given instance of a boolean formula in CNF an assignment to the variables exist such that each clause contains exactly one variable assigned the value *true*. The above problem is known to be NP-complete [15] and remains NP-complete if each clause of $\mathbb{F}$ contains exactly three different non-negated variables. The *ε-bandwidth problem* for $\varepsilon \in I\!R$ with $\varepsilon \geqslant 1$ is the following: Compute for a given graph $G = (V, E)$ a labeling e of V with $\mathrm{bw}_e(G) \leqslant \varepsilon \cdot \mathrm{bw}(G)$.

3. Blockers and Breakers

The components in our constructions are called "blocker" and "breaker". Here are the formal definitions of these components.

A component G is called a *(x, y)-blocker,* iff for all $i \in I\!N$: $\mathrm{bw}(G^i) \leqslant x$ and in any labeling e with bandwidth b ($x \leqslant b \leqslant y$) G^i cannot span another copy of G. To ease the description of our construction we also define pairs of blockers. G, H are called a pair of *(x, y)-blockers,* iff for all $i \in I\!N$: G, H are (x, y)-blockers and in any labeling e with bandwidth b ($x \leqslant b \leqslant y$) G^i (resp. H^i) cannot span another copy of H (resp. G). Note that a (x, y)-blocker is also a $(x, y-1)$-blocker.

A graph H is called a *breaker* for a (x, y)-blocker G iff for each b with $x \leqslant b \leqslant y$ there exists a labeling with bandwidth b such that GHG spans at least one copy of G. A breaker H is called a *z-breaker* for G iff for each b with $x \leqslant b \leqslant y$ exists no labeling with bandwidth b such that GHG spans G^{z+1}.

Based on the stretch factor of a component G we will construct a (x, y)-blocker. For the first step we introduce a restricted type of blocker for $x, y \in I\!N$ with $y < 2x$. G is called a *very simple (x, y, z)-blocker,* iff for all b with $x \leqslant b \leqslant y$ and for all i with $i \geqslant z$ and $\mathrm{Path}(G^i) = (v_0, v_1, ..., v_i)$ holds: $\mathrm{sbw}(G) = \mathrm{bw}(G^i) = x$ and G^i is not folded by bandwidth b. In any labeling e with $\mathrm{bw}(e, G^i) = b$ and G^i is not folded in e holds: the number of vertices mapped between $e(v_0)$ and $e(v_i)$ (including v_0 and excluding v_i) is at least $\mathrm{dist}_e(v_0, v_i)/2 + 1$, i.e. $\{v \mid e(v_0) \leqslant e(v) < e(v_i)\}| \geqslant \mathrm{dist}_e(v_0, v_i)/2 + 1$. When skipping the condition that G^i is not folded by bandwidth b we get the definition of a *simple (x, y, z)-blocker*.

Lemma 3.1 Let G be a graph which has a stretch factor of s and $x = \mathrm{sbw}(G)$. Then there exists $z \in I\!N$ with G is a very simple $(x, \lfloor s \cdot x \rfloor, z)$-blocker. Lem. 3.1 □

Proof of Lem. 3.1: Let $y = \lfloor s \cdot x \rfloor$ and $n = |\mathrm{V}(G)|$. We proof that there exists a $z \in I\!N$ such that G is a very simple (x, y, z)-blocker. Thus we have to check for $l \in I\!N$ if G^l makes G a very simple (x, y, l)-blocker. To check G^l we identify the different copies of G in G^l. Let $G^l = G_1 G_2 G_3 ... G_l$ with $\forall i, 1 \leqslant i \leqslant l : G_i \cong G$ and $\mathrm{Path}(G^l) = (v_0, v_1, v_2, ..., v_l)$. Assume that G^l is labeled by e such that G^l is not folded. The proof would be easy if all vertices of all G_i are embedded between v_{i-1} and v_i. This is not true in general. But we are able to estimate the number of vertices of G^l which are placed to the left of $e(v_0)$.

At most $y - 1$ edges of G^l may 'span' v_0. I.e. at most y edges $\{a, b\}$ may be labeled such that $e(a) < e(v_0) \leqslant e(b)$

holds. Thus the number of vertices placed to the left of $e(v_0)$ is at most $y \cdot (n-1)$. The total number of vertices not placed between $e(v_0)$ and $e(v_l)$ is at most $2 \cdot y \cdot (n-1)$ and at least $l \cdot (n-1) + 1 - 2 \cdot y \cdot (n-1)$ vertices are placed between $e(v_0)$ and $e(v_l)$.

The distance between v_i and v_{i+1} ($0 \leqslant i < l$) is strictly less than $2 \cdot (n-1)$ because the stretch factor of G is s. We conclude that $\mathrm{dist}_e(v_0, v_l) \leqslant 2 \cdot l \cdot (n-1) - l$. For the final step of the proof we set $l = 4 \cdot y \cdot (n-1)$ and compare the distance between $e(v_0)$ and $e(v_l)$ with the number of vertices embedded between them: $\mathrm{dist}_e(v_0, v_l)/2 + 1 \leqslant (2 \cdot l \cdot (n-1) - l)/2 + 1 = l \cdot (n-1) + 1 - 4 \cdot y \cdot (n-1)/2 = l \cdot (n-1) + 1 - 2 \cdot y \cdot (n-1)$. Which completes the proof. Note that the lemma holds for all $l \geqslant 4 \cdot y \cdot (n-1)$.

Lem. 3.1 ■

A copy of a very simple blocker constructed in Lemma 3.1 cannot span another copy when both are not folded. But when folding is allowed this is not true anymore.

Lemma 3.2 Let G be a graph with a stretch factor of s and $x = \mathrm{sbw}(G)$. Then there exists $z \in I\!N$ with G is a simple $(x, \lfloor s \cdot x \rfloor, z)$-blocker. Lem. 3.2 □

Proof of Lem. 3.2: Due to Lemma 3.1 G is a very simple $(x, \lfloor s \cdot x \rfloor, z')$-blocker G. We will now show that G is also a simple $(x, \lfloor s \cdot x \rfloor, z)$-blocker with $z = 5 \cdot z' + 2$. Let G' and G'' two node disjoint subgraphs with $G', G'' \cong G^{z'}$. If G' and G'' are not folded, then G' cannot span G''. This holds because the number of places not used by G' is strictly less than the number of vertices of G' and G'' with the same number of vertices as G' cannot be labeled by just using these free places. Thus in any labeling e of G^z is at least one copy G'''isomorphic to $G^{z'}$ which is labeled with number between $e(\mathrm{Left}(G'''))$ and $e(\mathrm{Right}(G'''))$. In the same way as in the proof of Lemma 3.1 this proof is accomplished.

Lem. 3.2 ■

Lemma 3.3 Let G be a graph with a stretch factor of s and $x = \mathrm{sbw}(G)$. Then there exist graphs B and B' such that B is a $(x, \lfloor s \cdot x \rfloor - 1)$-blocker and B' is a corresponding 1-breaker. Lem. 3.3 □

Proof of Lem. 3.3: Due to Lemma 3.2 G is a simple (x, y, z)-blocker for some z. We will show that $B = G^z$ is a $(x, \lfloor s \cdot x \rfloor)$-blocker. In any labeling of B the number of not used positions between $e(\mathrm{Left}(B))$ and $e(\mathrm{Right}(B))$ is strictly less than the number of used positions. Thus B cannot span an other graph isomorphic to B.

Let l be the distance between $\mathrm{Left}(B)$ and $\mathrm{Right}(B)$ in B. Thus $BB'B$ can span B in any labeling with bandwidth b ($x < b \leqslant \lfloor s \cdot x \rfloor$). But a graph of the form $BB'B$ may not span two copies of G, because there are not enough free places between $\mathrm{Left}(B')$ and $\mathrm{Right}(B')$. Thus H is a 1-breaker for G. Lem. 3.3 ■

4. Approximation with a factor of up to 2

In this section we will give the first results about the approximation of the bandwidth problem. We will show that for several graph classes the bandwidth cannot be approximated in polynomial time within a constant factor of $2 - \varepsilon$ for any $\varepsilon > 0$. For the first part of the construction we will define elements called *keyholes* and *keys* using the blockers and breakers of the previous sections. The aim is to have a finite set of pairs of keyholes and keys such that any two keyholes cannot span each other and for any keyhole only the corresponding key — or a general key — may span it. We start with a set of three pairs of keys and keyholes.

Lemma 4.1 Let G be a graph with a stretch factor of s, $x = \mathrm{sbw}(G)$ and $y = \lfloor s \cdot x \rfloor - 1)$. There exist graphs H_α,K_α with $\alpha \in \{0, 1, \aleph\}$ with:

a: For $\alpha, \beta \in \{0, 1, \aleph\}$ H_α and H_β form a pair of (x, y)-blockers.

b: For $\alpha \in \{0, 1\}$ H_α is a (x, y)-blocker and K_α is a corresponding 1-breaker.

c: For $\alpha \in \{0, 1, \aleph\}$ H_α is a (x, y)-blocker and $K_\aleph$ is a corresponding 2-breaker.

d: For $\alpha, \beta \in \{0, 1, \aleph\}$ if K_α spans H_β, then $\alpha \in \{\beta, \aleph\}$ holds. Lem. 4.1 □

Proof of Lem. 4.1: Let B and B' such that B is a $(x, \lfloor s \cdot x \rfloor - 1)$-blocker and B' is a corresponding 1-breaker given by Lemma 3.3. The construction of H_α,K_α with $\alpha \in \{0, 1, \aleph\}$ is as follows (see also an example in Section 2):

$$\begin{aligned}
H_0 &= B^5 S B^8 S^2 B^8 S B^5 \\
H_1 &= B^5 S^2 B^6 S B^2 S B^6 S^2 B^5 \\
K_0 &= S^5 B S^8 B^2 S^8 B S^5 \\
K_1 &= S^5 B^2 S^6 B S^2 B S^6 B^2 S^5 \\
K_\aleph &= S^{30} \\
H_\aleph &= B^{30}
\end{aligned}$$

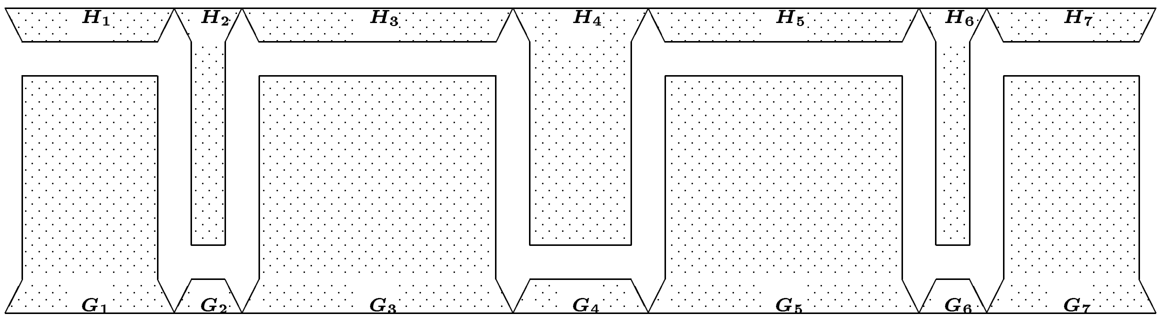

Figure 2. Example of keyhole H_0 and a fitting key K_0

Every H_α with $\alpha \in \{0, 1, \aleph\}$ contains a block of the form B^8 but no block of the form S^3, thus H_α cannot span an other H_β with $\beta \in \{0, 1, \aleph\}$ and property 4.1.a holds.

By construction K_α is a breaker for H_α for $\alpha \in \{0, 1\}$. By checking from left to right is is also easy to see that K_α spans the corresponding part of H_α. Thus property 4.1.b holds.

Each H_α with $\alpha \in \{0, 1, \aleph\}$ contains at least 26 Blockers B, thus $K_\aleph$ may span at most 2 copies of H_α and property 4.1.c holds.

To prove property 4.1.d we have to check that K_1 (resp. K_0) cannot span H_0 (resp. H_1). The first part S^5B^2 of K_1 cannot span the first part $B^5SB^8S^2$ of H_0. Thus K_1 cannot span H_0. The first part S^5B of K_0 cannot span the first part $B^5S^2B^6$ of H_1. As a consequence S^5B just spans B^5. We check the next part. The part $S^5BS^8B^2$ of K_0 cannot span $B^5S^2B^6SB^2SB^6$, which proves property 4.1.d.

Lem. 4.1 ■

The next lemma will present a larger set of keys and keyholes.

Lemma 4.2 Let $\overline{k} \in I\!N$, G be a graph with a stretch factor of s, $x = \mathrm{sbw}(G)$ and $y = \lfloor s \cdot x \rfloor - 1)$. Then there exist graphs H_i,K_i with $i \in \{\aleph, 0, 1, 2, ..., \overline{k}\}$ with:

a: For $\alpha, \beta \in \{\aleph, 0, 1, 2, ..., \overline{k}\}$ H_α and H_β form a pair of (x, y)-blockers.

b: For $\alpha \in \{0, 1, 2, ..., \overline{k}\}$ H_α is a (x, y)-blocker and K_α is a corresponding 1-breaker.

c: For $\alpha \in \{\aleph, 0, 1, 2, ..., \overline{k}\}$ H_α is a (x, y)-blocker and $K_\aleph$ is a corresponding 2-breaker.

d: For $\alpha, \beta \in \{\aleph, 0, 1, 2, ..., \overline{k}\}$ if K_α spans H_β, then $\alpha \in \{\beta, \aleph\}$ holds.

Lem. 4.2 □

Proof of Lem. 4.2: For $\alpha \in \{0, 1, \aleph\}$ let $\hat{B}_\alpha$ and $\hat{S}_\alpha$ be the blockers and breaker from Lemma 4.1. Let $l = \lceil \log(\overline{k}+1) \rceil$ and for $i \in I\!N_0$ with $i \leqslant \overline{k}$ let $\delta^n_1, \delta^n_2, ..., \delta^n_l$ be the binary representation of i. The construction of H_i,K_i with $i \in \{0, 1, 2, ..., \overline{k}\}$ is as follows:

$$\begin{aligned} H_i &= \hat{B}_\aleph \hat{B}_{\delta^i_1} \hat{B}_{\delta^i_2} ... \hat{B}_{\delta^i_l} \hat{B}_{\delta^i_l} ... \hat{B}_{\delta^i_2} \hat{B}_{\delta^i_1} \hat{B}_\aleph \\ K_i &= \hat{S}_\aleph S_{\delta^i_1} \hat{S}_{\delta^i_2} ... \hat{S}_{\delta^i_l} \hat{S}_{\delta^i_l} ... \hat{S}_{\delta^i_3} \hat{S}_{\delta^i_2} \hat{S}_{\delta^i_1} S_\aleph \\ H_\aleph &= \hat{B}^{2l+2}_\aleph \\ K_\aleph &= \hat{S}^{2l+2}_\aleph \end{aligned}$$

All properties are by construction a direct consequence of Lemma 4.1.

Lem. 4.2 ■

Lemma 4.3 Let $\mathbb{F}$ be an instance of the exact 3-satisfiability problem with n variables and m clauses. Let G be a graph with a stretch factor of s and $x = \mathrm{sbw}(G)$. If $\lfloor sx \rfloor - x > 4n$ holds, then there exist graphs B, S such that S spans B with bandwidth b ($x + 4n < b \leqslant \lfloor sx \rfloor$) iff $\mathbb{F}$ is satisfiable.

Lem. 4.3 □

Proof of Lemma 4.3: The construction is done by component design using the keys and keyholes defined in Lemma 4.2.

Let $\overline{k} = m + 2n + 2$ and $k \geqslant 10\overline{k}$. $\mathbb{F} = c_1 \wedge c_2 \wedge c_2 \wedge ... \wedge c_m$ contains the variables $v_1, v_2, ..., v_n$. In the following construction $x \in \{1, ..., \overline{k}\}$ will be used to represent variables, clauses and some constants.

Let V be the set of variables and C be the set of clauses of $\mathbb{F}$. We will use in the following the function $f(U) \rightarrow \{0, 1, 2, ..., \overline{k}\}$ with $U = \{0, 1, ..., \overline{k}\} \dot\cup C \dot\cup V \dot\cup \{true, false\}$ to ease the notation of the components.

$$x \text{ represents} \begin{cases} x & 0 \leqslant x \leqslant n \\ l_x & n+1 \leqslant x \leqslant 2n \\ c_{x-n} & 2n < x \leqslant n+m \\ true & x = 2n+m+1 \\ false & x = 2n+m+2 \end{cases}$$

$$f(x) = \begin{cases} x & x \in \{0, ..., n+m+4\} \\ i+n & x \in V \text{ and } x = v_i \\ i+2n & x \in C \text{ and } x = c_i \\ 1+2n+m & x = true \\ 2+2n+m & x = false \end{cases}$$

In the following a block H_x $(x \in U)$ will be the block $H_{f(x)}$ and a block K_x $(x \in U)$ will be the block $K_{f(x)}$. A clause $c_j = (v_{j_1} \vee v_{j_2} \vee v_{j_3})$ will be represented by the following block C_j. In block C_j we will have keyholes for c_j, each variable in c_j, twice $false$ and once $true$. The block D_j is defined nearly like C_j, except that $true$ occurs twice and $false$ once. Variable v_i is encoded in block V_i. For each variable v_i let a_i denote the number of clauses which include v_i and let $c_{h_i^1}, c_{h_i^2}, ..., c_{h_i^{a_i}}$ be the set of those clauses with $h_i^1 < h_i^2 <, ..., < h_i^{a_i}$. Furthermore let $h_i^0 = 1$. The whole construction is given by the blocks B and S.

In any labeling with bandwidth b $(x + 4n < b \leqslant y)$ where S spans B the following holds:

1. From a formula $\mathbb{F}$ the construction can be done in polynomial time.

2. We may assume wlog. that $\overline{B}$ is not folded.

3. Only the blocks $K_\aleph^{100m}$ may span the blocks $H_\aleph^{100m}$. Thus B spans all L_i. Furthermore all the keys of L_i have to fit the keyholes of F_C.

4. For any component X we denote by $\#(X)$ the number of component of type X in the above construction. Then the following holds:

$$\begin{array}{rcll} \#(H_i) & = & \#(K_i) & 0 \leqslant j \leqslant n \\ \#(H_{c_j}) & = & \#(K_{c_j}) & 1 \leqslant j \leqslant m \\ \#(H_{v_i}) & = & \#(K_{v_i}) & 1 \leqslant j \leqslant n \\ \#(H_x) & = & \#(K_x) & x \in \{true, false\} \end{array}$$

5. A block $X \in \{\overline{V}_{i,j}, \overline{W}_{i,j}\}$ spans a block $Y \in \{C_j, D_j\}$ such that the left part of X spans the left part of C_j resp. the right part of D_j.

6. A block $\overline{L}_i$ spans the block $\overline{F}_C$ in both directions.

7. The baseline of $\overline{F}_L$ moves from the left of F_C to the center of F_C where the key K_0 of $\overline{F}_L$ fits into a H_0 of F_C.

8. In any labeling the path of L_i is "embedded within" H_C as follows:

   (a) Starting from the center of H_C — from some H_0 — it moves left (right) to H_i using the key K_i.

   (b) Using the part V_i the baseline moves back to the center of H_C. On this way each $V_{i,j}$ spans the corresponding C_j (D_j). We say the variable v_i is set to $true$ $(false)$ in c_j.

   (c) Using the part W_i the baseline continues to moves to the right (left) part of H_C i.e. to to H_i using the key K_i. On this way each $W_{i,j}$ spans the corresponding D_j (C_j).

   (d) Finally the baseline moves back to the center of H_C where the key K_0 fits some H_0.

9. If $\mathbb{F}$ is satisfiable then using the assignment of the variables we may embed the blocks L_i as described above. In this case all keys fit the corresponding keyholes. I.e. in each C_i exactly one key K_{true} is placed. In this labeling e each keyhole H_x is spanned by at most $4n$ keys $K_\aleph$. Thus we may construct the labeling e with any bandwidth between $k + 4n$ and $b(k)$.

10. If there is a labeling e of F with $k + 4n \leqslant \mathrm{bw}_e(F) \leqslant b(k)$ then all keys fit the corresponding keyholes and form the embedding of L_i we may read the assignment to the variables which satisfy $\mathbb{F}$.

11. If $\mathbb{F}$ is not satisfiable then the bandwidth of B and S is at least $b(k) + 1$.

12. If $\mathbb{F}$ is satisfiable then the bandwidth of B and S is $\leqslant k + 4n$. Lemma 4.3 ■

For the final results on the complexity of the approximation problem for the bandwidth we need one more Lemma.

Lemma 4.4 Let $n \in I\!\!N$, $y, \varepsilon \in I\!\!R$ with $0 < \varepsilon < y \leqslant 2$. For $k > \frac{y4n+1}{\varepsilon} - 4n$ holds: $\frac{yk-1}{k+4n} > y - \varepsilon$ Lem. 4.4 □

$$
\begin{array}{lcll}
C_j & = & H_\aleph^5 H_{false} H_{false} H_{true} H_{v_{j_1}} H_{c_j} H_{v_{j_2}} H_{c_j} H_{v_{j_3}} H_{c_j} H_\aleph^6 & 1 \leqslant j \leqslant m \\
D_j & = & H_\aleph^6 H_{c_j} H_{v_{j_3}} H_{c_j} H_{v_{j_2}} H_{c_j} H_{v_{j_1}} H_{true} H_{true} H_{false} H_\aleph^5 & 1 \leqslant j \leqslant m \\
F_C & = & H_n ... H_2, H_1 \underbrace{C_{1,2,\dots,m} H_0^{n+1} D_{m,\dots,2,1}}_{\overline{F}_C} H_1 H_2 ... H_n & \\
V_{i,j} & = & K_\aleph^{20(h_i^j - h_i^{j-1})} \underbrace{K_\aleph^7 K_{true} K_\aleph^5 K_{v_i} K_{c_j} K_\aleph^{10}}_{\overline{V}_{i,j}} & 1 \leqslant i \leqslant n \text{ and } 1 \leqslant j \leqslant a_j \\
W_{i,j} & = & K_\aleph^{20(h_i^j - h_i^{j-1})} \underbrace{K_\aleph^6 K_{false} K_\aleph^6 K_{v_i} K_{c_j} K_\aleph^{10}}_{\overline{W}_{i,j}} & 1 \leqslant i \leqslant n \text{ and } 1 \leqslant j \leqslant a_j \\
V_i & = & K_\aleph^{20(m - h_i^{a_i})} V_{i,h_i^1} V_{i,h_i^2} V_{i,h_i^3} ... V_{i,h_i^{a_i}} & 1 \leqslant i \leqslant n \\
W_i & = & W_{i,h_i^{a_i}} ... W_{i,h_i^3} W_{i,h_i^2} W_{i,h_i^1} K_\aleph^{20(m - h_i^{a_i})} & 1 \leqslant i \leqslant n \\
L_i & = & K_\aleph^{20m} K_i \underbrace{V_i K_\aleph^{n+1} W_i}_{\overline{L}_i} K_i K_\aleph^{20m} K_0 & 1 \leqslant i \leqslant n \\
F_L & = & \underbrace{K_\aleph^{20m} K_0}_{\overline{F}_L} L_1 L_2 L_3 ... L_n & \\
B & = & H_\aleph^{100m} F_C H_\aleph^{100m} & \\
S & = & K_\aleph^{100m} F_L K_\aleph^{100m} &
\end{array}
$$

Figure 3. The main contruction

Proof of Lemma 4.4:

$$
\begin{array}{rcll}
\frac{y4n+1}{\varepsilon} - 4n & < & k & \Rightarrow \\
\frac{y4n}{\varepsilon} & < & (k+4n) - \frac{1}{\varepsilon} & \Rightarrow \\
4n & < & \frac{\varepsilon}{y}(k+4n) - \frac{1}{y} & \Rightarrow \\
k + 4n - k\frac{\varepsilon}{y} - 4n\frac{\varepsilon}{y} & < & k - \frac{1}{y} & \Rightarrow \\
(1 - \frac{\varepsilon}{y})(k+4n) & < & k - \frac{1}{y} & \Rightarrow \\
1 - \frac{\varepsilon}{y} & < & \frac{k-1/y}{k+4n} & \Rightarrow \\
y - \varepsilon & < & \frac{yk-1}{k+4n} &
\end{array}
$$

Lemma 4.4 ■

The next theorem of this section combines the breaking problem and the stretch factor of a path constructable graph class.

Theorem 4.5 If for every $\delta > 0$ ($\delta \in I\!R$) and $b \in I\!N$ a graph G exists with $\mathrm{sbw}(G) \geqslant b$ and G has a stretch-factor of $s - \varepsilon$ ($\varepsilon \in I\!R$, $0 \leqslant \varepsilon \leqslant \delta$) with $s \leqslant 2$, $s - \varepsilon > 1$ and $\varepsilon \geqslant 0$, then the ε'-bandwidth approximation problem is NP-complete for all $\varepsilon' \in I\!R$ with $\varepsilon' > 0$ and $s - \varepsilon > 1$. The. 4.5 □

Proof of The. 4.5: Let $\mathbb{F}$ be a input of the exact 3-satisfiability problem that contains n variables. We set $s' = s - \varepsilon$ and $k = \frac{s'4n+1}{\varepsilon'} - 4n + 1$. Using Lemma 4.4 we get $\frac{s'k-1}{k+4n} > s' - \varepsilon$. By Lemma 4.3 there exist graphs B, S such that S spans B with bandwidth b ($x + 4n < b \leqslant y$) iff $\mathbb{F}$ is satisfiable. We construct now a new Graph $G = \mathcal{M}^{\mathrm{Left}(G')}_{\mathrm{Right}(G')}(G')$ with $G' = \mathcal{M}^{\mathrm{Left}(B)}_{\mathrm{Right}(S)}(S+B)$. This graph has bandwidth $x + 4n$ iff $\mathbb{F}$ is satisfiable, otherwise the bandwidth of G is y. Thus the $(s-\varepsilon')$-breaking problem is NP-complete. The. 4.5 ■

We summarize some approximation results in the following theorems:

Theorem 4.6 The $(2-\varepsilon)$-approximation of the bandwidth is NP-complete for every small $\varepsilon \in I\!R, \varepsilon > 0$. The. 4.6 □

Proof of The. 4.6: By using Theorem 4.5 it remains to construct for each $b \in I\!N$ a graph G with $\mathrm{sbw}(G) \geqslant b$ and a stretch factor of $2 - \varepsilon'$ for every small $\varepsilon' \in I\!R, \varepsilon' > 0$. This technical part of the proof is skipped. The. 4.6 ■

A unit circular-arc graph is the intersection graph of the set of unit arcs on a cycle.

Theorem 4.7 For the class of unit circular-arc graphs and for every small $\varepsilon \in I\!R, \varepsilon > 0$ the $(2-\varepsilon)$-approximation of the bandwidth is NP-complete. For a unit circular-arc graph the 2-approximation of the bandwidth is possible in polynomial time. The. 4.7 □

Proof of The. 4.7: In the same way as in the proof of Theorem 4.6 we may construct a unit circular-arc graph with a stretch factor of $2 - \varepsilon'$ for every small $\varepsilon' \in I\!R, \varepsilon' > 0$. We note that the final construction will then also be a unit circular-arc graph. The second part of the proof is presented in [10]. The. 4.7 ■

A ringed caterpillar is a caterpillar where one edge is added, such that the backbone becomes a cycle. Note that due to the limited space we omit the complex proves for the following theorems

Theorem 4.8 For every small $\varepsilon \in I\!R, \varepsilon > 0$ the $(2-\varepsilon)$-approximation of the bandwidth for the class of ringed caterpillars with hair-length 1 is NP-complete. For ringed caterpillars with hair-length 1 the 2-approximation of the bandwidth is possible in polynomial time. The. 4.8 □

Theorem 4.9 For the class of ringed caterpillars with degree 3 and for every small $\varepsilon \in I\!R, \varepsilon > 0$ the $(2-\varepsilon)$-approximation of the bandwidth is NP-complete. For ringed caterpillars with degree 3 the 2-approximation of the bandwidth is possible in polynomial time. The. 4.9 □

It is also possible to define a graph glass by joining graphs with a stretch factor of x with $1 < x < 2$.

Theorem 4.10 For any $x, \varepsilon \in I\!R$ with $2 > x > 1$ and $\varepsilon > 0$ exists a graph class for which an approximation algorithm with an approximation factor of $x + \varepsilon$ exists, but the approximation of the bandwidth problem within a factor of $x - \varepsilon$ is NP-complete. The. 4.10 □

5. Approximation Factors $k \in I\!N$

Due to the limited space we present just the basic idea for the following theorems. We present here just the basic idea. Assume that the blockers we are using in the previous construction have a bandwidth of b. Depending on the boolean formula $\mathbb{F}$ we have two cases. If there is a satisfying assignment to the variables of $\mathbb{F}$ then the construction has a smooth fitting. In this case the bandwidth will be $b + b'$ for some $b \in I\!N$ with $b \gg b'$. If there is no satisfying assignment to the variables of $\mathbb{F}$ then the construction will have no smooth fitting and the bandwidth is of the construction is $2 \cdot b$.

The graphs G_1 and G_4 in Figure Section 4 represent one construction simulating a boolean formula for which a satisfying assignment of the variables exists. The graphs G_2 and G_5 simulate boolean formula with no satisfying assignment of the variables exist. Thus G_1 and G_4 have a bandwidth of $b + b'$ and G_2 and G_5 have a bandwidth of $2b$.

We will now use this construction several times to build a new more powerful blocker. The bandwidth of this blocker is either $2 \cdot b$ or $b + b'$. This new blocker is now used in a similar way as before. In the whole construction — simulation $\mathbb{F}$ has either a smooth fitting (see Section 5) or no smooth fitting (see Section 6).

It depends on $\mathbb{F}$ whether the whole construction and each of its blockers may have a smooth fitting or not. In the first case the bandwidth is $b + 2 \cdot b'$. In the other case (there is no satisfying assignment to the variables of $\mathbb{F}$) neither the whole construction nor any of the blockers have smooth fitting and the bandwidth will be $3 \cdot b$. With this single step of recursive construction we will show that the bandwidth of a graph can not be approximated within a factor of $3 - \varepsilon$ (unless $P = NP$ holds). By using the construction with multiple recursive steps we are able to show the following theorem.

Theorem 5.1 The k-approximation of the bandwidth is NP-complete for every $k \in I\!N$. The. 5.1 □

So far we used a cyclic construction to force the interaction between keyholes and keys. But it is also possible to introduce a barrier, which force the labeling of two parts of the construction on the same side of this barrier. By using this barrier in a recursive way we get the same type of results as before.

Theorem 5.2 The k-approximation of the bandwidth for caterpillars of degree three is NP-complete for every $k \in I\!N$. The. 5.2 □

Finally we note that Theorem 4.8 may also be generalized.

Theorem 5.3 For any $x, \varepsilon \in I\!R$ with $x > 1$ and $\varepsilon > 0$ exists a graph class for which an approximation algorithm with an approximation factor of $x + \varepsilon$ exists, but the approximation of the bandwidth problem within a factor of $x - \varepsilon$ is NP-complete. The. 5.3 □

6. Conclusions

The important open question given by [1] was to improve the upper and lower approximation bounds for the general

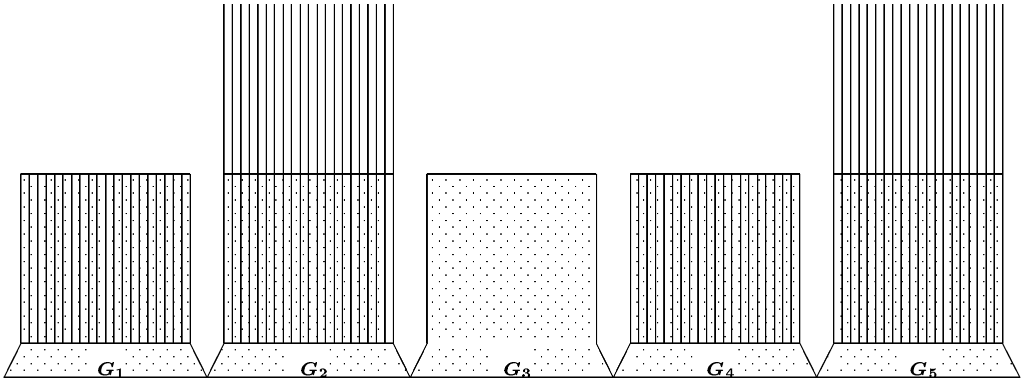

Figure 4. Symbols to be used in the recursive construction

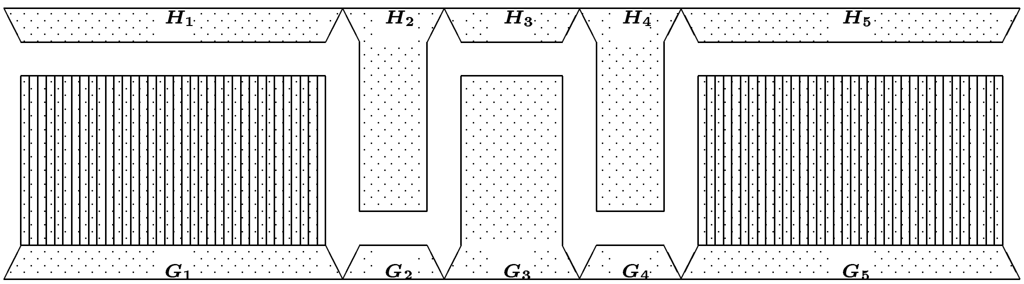

Figure 5. Recursive construction with small bandwidth

bandwidth problem. The huge gap presented by [1] was between 1.5 and $O(\log^{11/2} n)$ (cf. [4]). We have closed this gap considerably. The bandwidth approximation problem is NP-complete for any approximation factor $k \in I\!N$. It remains to close this new 'small' gap. A close look to the techniques may already give intractability results for the bandwidth problem with an approximation factor $f(n)$.

References

[1] G. Blache M. Karpinski, J. Wirthgen, *On Approximation Intractability of the Bandwidth Problem*, Electronic Colloquium on Computational Complexity, Rep. No 14 (1998) Technical Report ECCC TR 97-041, 1997

[2] P. Chinn, J. Chvatalova, A. Dewdney, N. Gibbs, *The Bandwidth Problem for Graphs and Matrices – A Survey*, Journal of Graph Theory (1982), pp. 223-254

[3] F. Chung, *Labelings of Graphs*, L. Beineke, R. Wilson ed, Selected Topics in Graphs Theory, pp. 151-168, Academic Press, 1988

[4] U. Feige, *Approximating the Bandwidth via Volume Respectiin Embeddings*, ms. 1997.

[5] M. Garey, R. Graham, D. Johnson, D. Knuth, *Complexity Results for Bandwidth Minization*, SIAM Journal of Appl. Math. 34 (1996), pp. 339-348

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

[7] F. Harary, *Graph theory*, Addison-Wesley Publ. Company, 1969.

[8] M. Karpinski, J. Wirthgen, *On Approximation Hardness of the Bandwidth Problem*, Technical Report ECCC TR 97-041, 1997

[9] M. Karpinski, J. Wirthgen, A. Zelikovsky, *On Approximation Algorithm for the Bandwidth Problem on Dense Graphs*, Technical Report ECCC TR 97-017, 1997

[10] T. Kloks, D. Kratsch, H. Müller, *Approximating the Bandwidth for Asteroidal Triple-free Graphs*, 3^{rd} Annual European Symposium on Algorithms (1995)

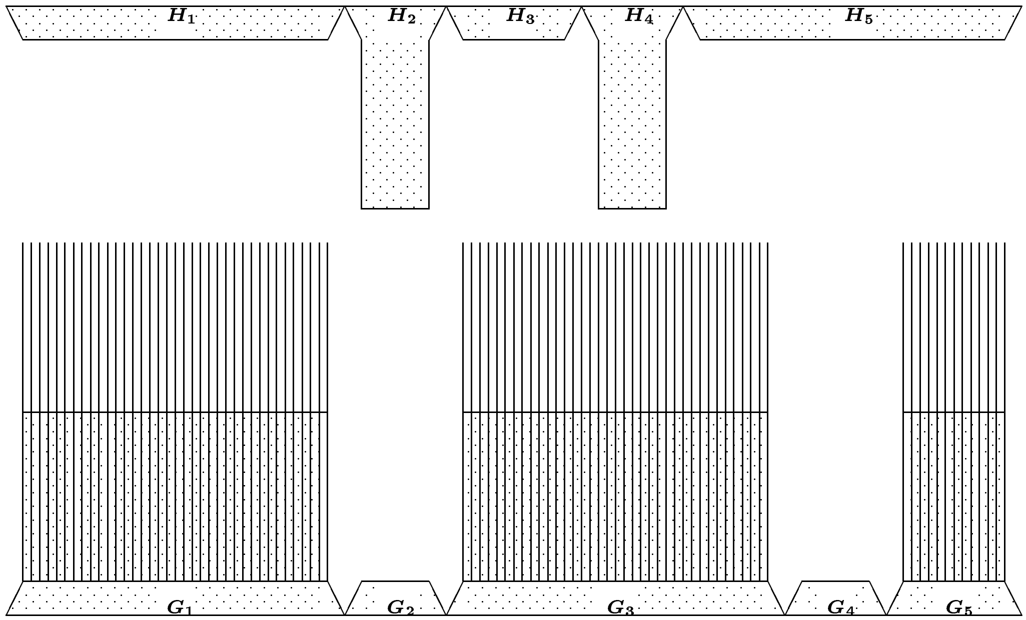

Figure 6. Recursive construction with large bandwidth

[11] B. Monien, *The Bandwidth Minimization Problem for Caterpillars with Hair Length 3 is NP-Complete* 1986 7 SIAM Journal on Algebraic Discrete Methods 7 (1986), pp. 505-512.

[12] B. Monien, I.H. Sudborough, *Embedding one Interconnection Network in Another*, Computing Suppl. **7** (1990), 257–282.

[13] C. Papadimitriou, *The NP-Completness of the Bandwidth Minimization Problem*, Computing 16 (1976), pp. 263-270.

[14] J. Saxe, *Dynamic Programming Algorithms for Recognizing Small-Bandwidth Graphs*. SIAM Journal on Algebraic Methods 1 (1980), pp. 3363-369

[15] T.J. Schäfer, *The complexity of satisfiability problems*, Proc. 10^{th} Ann. ACM Symp. on Theory of Computing, pp. 216-266

[16] I. Smithline, *Bandwidth of the Complete k-ary Tree*, Discrete Mathematics 142 (1995), pp 203-212.