Bandwidth on AT-free graphs*

Petr Golovach† Pinar Heggernes† Dieter Kratsch‡ Daniel Lokshtanov†
Daniel Meister† Saket Saurabh†

Abstract

We study the classical Bandwidth problem from the viewpoint of parametrised algorithms. Given a graph $G = (V, E)$ and a positive integer k, the Bandwidth problem asks whether there exists a bijective function $\beta : \{1, \ldots, |V|\} \to V$ such that for every edge $uv \in E$, $|\beta^{-1}(u) - \beta^{-1}(v)| \leq k$. It is known that under standard complexity assumptions, no algorithm for Bandwidth with running time of the form $f(k)n^{\mathcal{O}(1)}$ exists, even when the input is restricted to trees. We initiate the search for classes of graphs where such algorithms do exist. We present an algorithm with running time $n \cdot 2^{\mathcal{O}(k \log k)}$ for Bandwidth on AT-free graphs, a well-studied graph class that contains interval, permutation, and cocomparability graphs. Our result is the first non-trivial algorithm that shows fixed-parameter tractability of Bandwidth on a graph class on which the problem remains NP-complete.

Keywords: algorithm, fixed-parameter tractable, bandwidth, AT-free graph.

1 Introduction

A layout for a graph G is a bijective function $\beta : \{1, \ldots, |V(G)|\} \to V(G)$. The *bandwidth* of G is the smallest integer b such that G has a layout β where for every edge $uv \in E(G)$, $|\beta^{-1}(u) - \beta^{-1}(v)| \leq b$. Given a graph G and an integer k, the decision problem Bandwidth asks whether the bandwidth of G is at most k. 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 like 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 [10].

Bandwidth is one of the most well-known and most extensively studied graph layout problems [9]. It is NP-complete [22] on general graphs and remains NP-complete even on very restricted subclasses of trees, like caterpillars of hair length at most 3 [19]. Furthermore,

*This work is supported by the Research Council of Norway. A preliminary version of the paper was presented at the 20th International Symposium on Algorithms and Computation (ISAAC 2009).

†Department of Informatics, University of Bergen, 5020 Bergen, Norway. Emails: {petr.golovach, pinar.heggernes, daniel.lokshtanov, daniel.meister, saket.saurabh}@ii.uib.no

‡Laboratoire d'Informatique Théorique et Appliquée, Université Paul Verlaine – Metz, 57045 Metz Cedex 01, France. Email: kratsch@univ-metz.fr

the bandwidth of a graph is NP-hard to approximate within a constant factor for trees [2]. Polynomial-time algorithms for the exact computation of bandwidth are known for a few graph classes including caterpillars of hair length at most 2 [1], cographs [24], interval graphs [14] and bipartite permutation graphs [12]. The best known algorithm, deciding for a constant k whether an input graph G has bandwidth at most k, is the celebrated dynamic programming algorithm by Saxe [23], which runs in time $2^{\mathcal{O}(k)}n^{k+1}$. However, as the value of k grows, the exponent of the polynomial in the running time grows with it. A natural question is whether BANDWIDTH can be solved in time $f(k)n^c$ where c is a constant independent of k. This amounts to asking whether BANDWIDTH is *fixed-parameter tractable.*

Parametrised complexity is a two-dimensional generalisation of "P vs. NP" where, in addition to the overall input size n, one studies how a secondary measurement that captures additional relevant information affects the computational complexity of the problem in question. Parametrised decision problems are defined by specifying the triple *input/parameter/question to be answered.* The two-dimensional analogue of P is solvability within a time bound of $f(k)n^c$, where n is the total input size, k is the parameter, f is some computable function, and c is a constant that does not depend on k or n. A parametrised problem that can be solved in such time is termed *fixed-parameter tractable*, *FPT*. There is a hierarchy of intractable parametrised problem classes above the class FPT, the main ones being:

$$\mathrm{FPT} \subseteq \mathrm{W}[1] \subseteq \mathrm{W}[2] \subseteq \cdots \subseteq \mathrm{W}[\mathrm{P}] \subseteq \mathrm{XP}\,.$$

This hierarchy is called the W*-hierarchy.* The principal analogue of the classical intractability class NP is W[1], which is a strong analogue, because a fundamental problem complete for W[1] is the k-STEP HALTING PROBLEM FOR NONDETERMINISTIC TURING MACHINES with unlimited nondeterminism and alphabet size – this completeness result provides an analogue of Cook's Theorem in classical complexity. Thus, a parametrised problem that is hard for W[1] is unlikely to be fixed-parameter tractable. For general background on the theory, see the textbooks by Downey and Fellows [7], Flum and Grohe [11], and Niedermeier [20].

In a seminal paper, Bodlaender, Fellows, and Hallet showed that BANDWIDTH on trees is hard for W[t] for every $t \geq 1$ [3]. This rules out the existence of an FPT-algorithm for BANDWIDTH unless the entire W-hierarchy collapses. The hardness result in [3] indicates that the tractable cases for BANDWIDTH seem to be few and far between. Here, we initiate the search for classes of graphs where BANDWIDTH is fixed-parameter tractable.

For the graph classes for which polynomial-time algorithms are known, it has been proved that BANDWIDTH becomes NP-complete (or its complexity remains unknown) on slightly larger graph classes. Therefore, it is natural to investigate the *parametrised* complexity of BANDWIDTH on these larger classes of graphs. In this paper, we present an algorithm with running time $n \cdot 2^{\mathcal{O}(k \log k)}$ for BANDWIDTH on AT-free graphs. A graph is *AT-free* if for every triple of pairwise non-adjacent vertices, the neighbourhood of one of them separates the two others. The class of AT-free graphs contains various well-known graph classes, like interval, permutation, trapezoid, and cocomparability graphs [4]. While BANDWIDTH is polynomial-time solvable on interval graphs [14] and well-studied subclasses of permutation graphs [24, 12], it is NP-complete on cocomparability graphs and hence on AT-free graphs [16]. For permutation graphs, the complexity of BANDWIDTH is a well-known open problem. Most natural superclasses of AT-free graphs contain trees, and thus, the hardness result in [3] rules out an FPT-algorithm

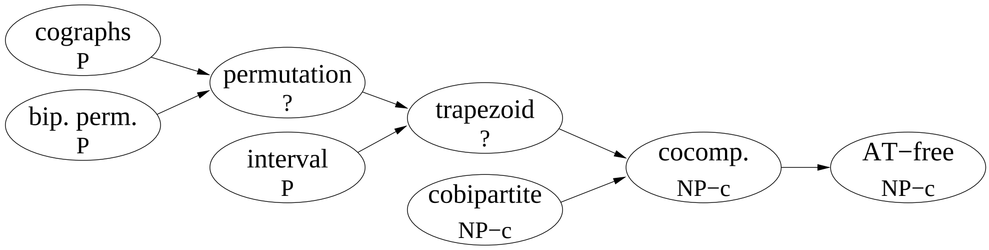

Figure 1: A graph class inclusion diagram with the classical complexity of BANDWIDTH on these graph classes.

for BANDWIDTH on these classes. Thus, our FPT-algorithm on AT-free graphs essentially settles the parametrised complexity of BANDWIDTH on the chain of natural graph classes above the polynomial cases (Figure 1).

Our algorithms might be seen as modifications of Saxe's dynamic programming algorithm aiming at smaller search space by exploiting the structure of graph classes. They are based on two results. In Section 3, we introduce the notion of (k,l)-*layout*, where k and l are non-negative integers, that correlates distances in the layout and in the graph. Informally spoken, adjacent vertices are not too far apart in the layout, and vertices that are close in the layout are not too distant in the graph. We show that the existence problem of such layouts is fixed-parameter tractable when parametrised with k and l. The main algorithmic idea is to construct a digraph, that we call the *auxiliary digraph*, whose arcs represent small portions of feasible layouts, that can be combined along directed paths. For obtaining our FPT-algorithm for AT-free graphs, that is parametrised with k only, we show in Section 4 that the size of l can be bounded from above by a small function in k. For this result, we rely on structural properties of AT-free graphs and their relationship to interval graphs. In the conclusions section of the paper, we briefly discuss applications of our approach to other graph classes and list some possible emerging research directions.

2 Definitions and notation

In this paper, we consider simple finite directed and undirected graphs. Directed graphs are shortly called *digraphs*, and by "graph", we always mean undirected graph.

For a digraph $G=(V,A)$, V denotes the vertex set of G and A denotes the arc set of G. If (u,v) is an arc of G then u is an *in-neighbour* of v and v is an *out-neighbour* of u. For a vertex pair u,v of G and an integer $t\geq 0$, a *directed* u,v*-path of* G *of length* t is a sequence $(x_0,\ldots,x_t)$ of pairwise different vertices of G such that $x_0=u$ and $x_t=v$ and $(x_i,x_{i+1})\in A$ for every $0\leq i<t$. A *shortest* directed u,v-path of G is a directed u,v-path of G of smallest length.

For a graph $G=(V,E)$, $V=V(G)$ denotes the vertex set and $E=E(G)$ denotes the edge set of G. Edges of G are denoted as uv, and if uv is an edge of G, we call u and v *adjacent*, and v is a *neighbour* of u. The *neighbourhood* of a vertex u, denoted as $N_G(u)$, is the set of vertices that are adjacent to u. A graph H is called *subgraph* of G, denoted as $H\subseteq G$, if $V(H)\subseteq V(G)$ and $E(H)\subseteq E(G)$. For a set $U\subseteq V(G)$, the subgraph of G *induced* by U, denoted as $G[U]$, is the subgraph of G on vertex set U and with edge set $\{uv \in E(G) : u, v \in U\}$. For a vertex pair u, v of G, a *u, v-path of G of length r* is a sequence $(u_0, \ldots, u_r)$ of pairwise different vertices where $u_0 = u$, $u_r = v$ and $u_i u_{i+1} \in E(G)$ for every $0 \leq i < r$. The *distance of u and v in G*, denoted as $\mathrm{d}_G(u, v)$, is the smallest length of a u, v-path in G; if G has no u, v-path then the distance of u and v in G is ∞. Graph G is *connected* if there is a u, v-path in G for every vertex pair u, v of G. A *connected component* of G is a connected subgraph H of G where $V(H)$ and $E(H)$ are inclusion-maximal. A set $C \subseteq V(G)$ is called *clique* of G if the vertices in C are pairwise adjacent. The *size* of C is $|C|$.

Let G be a graph. A *layout* (or *vertex ordering*) is a bijective function from $\{1, \ldots, |V(G)|\}$ to $V(G)$. For β a layout, we also write β as $\langle \beta(1), \ldots, \beta(n) \rangle$. As a special case, we allow the *empty layout* $\langle \rangle$. For a vertex pair u, v of G, the *distance* between u and v in β, $\mathrm{d}_\beta(u, v)$, is $|\beta^{-1}(u) - \beta^{-1}(v)|$. We write $u \preccurlyeq_\beta v$ if $\beta^{-1}(u) \leq \beta^{-1}(v)$ and $u \prec_\beta v$ if $\beta^{-1}(u) < \beta^{-1}(v)$. For an integer $k \geq 0$, we call β a *k-layout* for G if for every edge uv of G, $d_\beta(u, v) \leq k$. The *bandwidth* of G, denoted as $\mathrm{bw}(G)$, is the smallest integer $b \geq 0$ such that G has a b-layout. The bandwidth of G is equal to the maximum bandwidth of its connected components. If H is a subgraph of G then $\mathrm{bw}(H) \leq \mathrm{bw}(G)$. For two layouts $\beta = \langle \beta(1), \ldots, \beta(r) \rangle$ and $\gamma = \langle \gamma(1), \ldots, \gamma(s) \rangle$, $\beta \circ \gamma$ is the layout resulting from appending γ to β, which means, $\beta \circ \gamma = \langle \beta(1), \ldots, \beta(r), \gamma(1), \ldots, \gamma(s) \rangle$. Note that the $\circ$ operator satisfies the associativity law.

In this paper, we study AT-free graphs and subclasses of AT-free graphs. Let G be a graph. A set $\{u, v, w\}$ of three pairwise different and non-adjacent vertices of G is called an *asteroidal triple*, *AT* for short, if for every pair x, y from $\{u, v, w\}$, there is an x, y-path in G that contains no neighbour of the third vertex. A graph that has no asteroidal triple is called *AT-free*. Every connected component of an AT-free graph is AT-free. For more information on structural properties of AT-free graphs, we refer to [4, 5].

3 Restricted layouts and the auxiliary digraph

In this section, we consider a special type of k-layouts. In addition to the constraint that adjacent vertices be close in the layout, we also require that vertices which are close in the layout be close in the graph. We define "being close" as a distance condition. We characterise the existence of such layouts, and we show that the existence is efficiently decidable for fixed distance bounds.

Let G be a graph and let β be a layout for G. For $k, l \geq 0$, we say that β is a *(k, l)-layout* if β is a k-layout for G and for every vertex pair u, v of G, if $\mathrm{d}_\beta(u, v) \leq k$ then $\mathrm{d}_G(u, v) \leq l$. Informally spoken, the vertices that are at small distance in β are at bounded distance in G. It is not difficult to see that for G connected, G has a k-layout if and only if G has a (k, n)-layout. Therefore, every connected graph of bandwidth at most k has a (k, l)-layout for some integer l. We aim at characterising the existence of (k, l)-layouts and thereby providing a tool for an efficient decision algorithm.

Let G be a connected graph, and let $k, l \geq 0$ such that G has at least $k + 1$ vertices. For a vertex u of G and an integer $l \geq 1$, the *ball around u of radius l*, $B_G(u, l)$, is the set of vertices of G that are at distance at most l to u in G. Formally, $B_G(u, l) =_{\mathrm{def}} \{x \in V(G) : \mathrm{d}_G(u, x) \leq l\}$. Analogously, for β a layout for G, $B_\beta(u, l) =_{\mathrm{def}} \{x \in V(G) : \mathrm{d}_\beta(u, x) \leq l\}$. With these definitions, β is a (k, l)-layout for G if and only if $B_\beta(x, k) \subseteq B_G(x, l)$ for every vertex x of G.

We define the *auxiliary digraph* $\mathrm{aux}(G,k,l)$. The digraph has four types of vertices, three of which are labelled with layout pairs:

- *start vertices*
  let a be a vertex of G and let $U \subseteq B_G(a,l)$ where $|U| = k$ and $a \in U$; let α be a layout for $G[U]$ such that $\alpha(1) = a$;
  $\mathrm{aux}(G,k,l)$ has a vertex with label $(\langle\rangle, \langle\alpha(1),\ldots,\alpha(k)\rangle)$

- *middle vertices*
  let a be a vertex of G and let $U \subseteq B_G(a,l)$ where $|U| = 2k$ and $a \in U$; let α be a layout for $G[U]$ such that $\alpha(k+1) = a$;
  $\mathrm{aux}(G,k,l)$ has a vertex with label $(\langle\alpha(1),\ldots,\alpha(k)\rangle, \langle\alpha(k+1),\ldots,\alpha(2k)\rangle)$

- *end vertices*
  let a be a vertex of G and let $U \subseteq B_G(a,l)$ where $k \leq |U| < 2k$ and $r \cdot k + |U| = |V(G)|$ for some integer r and $a \in U$; if $|U| = k$ then let α be a layout for $G[U]$ such that $\alpha(1) = a$, if $|U| \geq k+1$ then let α be a layout for $G[U]$ such that $\alpha(k+1) = a$;
  $\mathrm{aux}(G,k,l)$ has a vertex with label $(\langle\alpha(1),\ldots,\alpha(k)\rangle, \langle\alpha(k+1),\ldots,\alpha(|U|)\rangle)$

- the two special vertices S and T.

We call vertex a the *anchor* of α or the defined vertex of $\mathrm{aux}(G,k,l)$. We define the arcs of the auxiliary digraph. Let u,v be a vertex pair of $\mathrm{aux}(G,k,l)$ where $u \neq v$ and $u,v \notin \{S,T\}$. Let (β,β') and (γ,γ') be the layout pair label of respectively u and v, and let $U_\beta, U_{\beta'}, U_\gamma, U_{\gamma'}$ be the set of vertices contained in respectively $\beta,\beta',\gamma,\gamma'$. The auxiliary digraph contains arc (u,v) if the following conditions are satisfied:

- u is a start or middle vertex, and v is a middle or end vertex
- $\beta' = \gamma$
- $\beta \circ \beta' \circ \gamma'$ is a k-layout for $G[U_\beta \cup U_{\beta'} \cup U_{\gamma'}]$
- for every vertex $w \in U_{\beta'}$, $N_G(w) \subseteq U_\beta \cup U_{\beta'} \cup U_{\gamma'}$;
  if v is an end vertex then for every vertex $w \in U_{\gamma'}$, $N_G(w) \subseteq U_\beta \cup U_{\beta'} \cup U_{\gamma'}$
- for every vertex $w \in U_\beta \cup U_{\beta'} \cup U_{\gamma'}$, $B_{\beta\circ\beta'\circ\gamma'}(w,k) \subseteq B_G(w,l)$.

Finally, add all arcs of the forms (S,u) and (v,T) where u is a start vertex and v is an end vertex. This completes the definition of the auxiliary digraph. In particular, $\mathrm{aux}(G,k,l)$ has no further vertices or arcs. Below, we will refer to the two last conditions on the existence of arcs as the *neighbourhood* and *distance condition.* We only remark here that $\mathrm{aux}(G,k,l)$ may have directed cycles.

Lemma 3.1 *Let G be a connected graph on n vertices. Let $k,l \geq 1$ be integers where $k < n$. Let r be smallest such that $rk > n$.*

1) If $\mathrm{aux}(G,k,l)$ has a directed S,T-path then G has a k-layout.

2) G has a (k,l)-layout if and only if $\mathrm{aux}(G,k,l)$ *has a directed S,T-path of length at most $r+1$.*

Proof. For the first statement of the lemma, we show that every directed S,T-path of $\mathrm{aux}(G,k,l)$ defines a k-layout for G. Let $(u^{(0)}, u^{(1)}, \ldots, u^{(t+1)})$ be a directed S,T-path of $\mathrm{aux}(G,k,l)$. Note that $t \geq 2$ and $u^{(0)} = S$ and $u^{(t+1)} = T$. For every $1 \leq i \leq t$, let (β_i, γ_i) be the layout pair label of $u^{(i)}$. Let $U_1, \ldots, U_t$ be the sets of vertices that appear in respectively $\gamma_1, \ldots, \gamma_t$. Note that $\gamma_t = \langle\rangle$ and thus $U_t = \emptyset$ may happen. For convenience reasons, let $\gamma_0 =_{\mathrm{def}} \langle\rangle$ and $U_0 =_{\mathrm{def}} \emptyset$. We show that $\gamma =_{\mathrm{def}} \gamma_0 \circ \cdots \circ \gamma_t$ contains a k-layout for G. First, we show that every vertex of G appears at least once in γ. Clearly, $\gamma(1)$ is a vertex of G and appears in γ. We show the claim by induction through a breadth-first-search strategy. Here, it is important that G is connected. Let y be a vertex of G that has an already observed neighbour x in γ. Let $1 \leq p \leq t$ be such that $x \in U_p$. If $p \leq t-1$ then $u^{(p)}$ is not an end vertex of $\mathrm{aux}(G,k,l)$ and $y \in U_{p-1} \cup U_p \cup U_{p+1}$ according to the neighbourhood condition on $(u^{(p)}, u^{(p+1)})$. So, y appears in $\gamma_{p-1} \circ \gamma_p \circ \gamma_{p+1}$. If $p = t$ then $u^{(p)} = u^{(t)}$ is an end vertex of $\mathrm{aux}(G,k,l)$, and $y \in U_{t-2} \cup U_{t-1} \cup U_t$ due to the neighbourhood condition on $(u^{(t-1)}, u^{(t)})$. Thus, y appears in $\gamma_{t-2} \circ \gamma_{t-1} \circ \gamma_t$.

As the next step, we extract a layout for G from γ. For every vertex x of G, let $\pi(x) \geq 1$ be the smallest integer such that $\gamma(\pi(x)) = x$. The above proved result shows that π is well-defined. Let δ be the layout for G such that for every integer pair i,j, if $i < j$ then $\pi(\delta(i)) < \pi(\delta(j))$. Informally, δ is the "leftmost occurrence" sub-layout of γ. We show that δ is a k-layout for G. Let xy be an edge of G. Without loss of generality, we can assume $\pi(x) < \pi(y)$. Let $1 \leq p \leq t$ be such that position $\pi(x)$ in γ is in γ_p. Assume that $p < t$. Then, x appears in $\gamma_{p-1} \circ \gamma_p \circ \gamma_{p+1} = \beta_p \circ \gamma_p \circ \gamma_{p+1}$ due to the definition of arc $(u^{(p)}, u^{(p+1)})$ of $\mathrm{aux}(G,k,l)$. Due to the neighbourhood condition, it follows that $y \in U_{p-1} \cup U_p \cup U_{p+1}$, which means that y appears in $\beta_p \circ \gamma_p \circ \gamma_{p+1}$. Since $\beta_p \circ \gamma_p \circ \gamma_{p+1}$ is a k-layout for $G[U_{p-1} \cup U_p \cup U_{p+1}]$, we conclude that y appears exactly once in $\beta_p \circ \gamma_p \circ \gamma_{p+1}$ and $\mathrm{d}_{\beta_p \circ \gamma_p \circ \gamma_{p+1}}(x,y) \leq k$. Now, observe that position $\pi(y)$ in γ is in $\gamma_p \circ \gamma_{p+1}$, since $\pi(x) < \pi(y)$ and y appears in $\gamma_p \circ \gamma_{p+1}$. It follows from the definition of δ that $\mathrm{d}_\delta(x,y) \leq \mathrm{d}_{\beta_p \circ \gamma_p \circ \gamma_{p+1}}(x,y) = \pi(y) - \pi(x) \leq k$, which completes the case when $p < t$. Now, assume $p = t$, which means that x appears in $\gamma_{t-1} \circ \gamma_t = \beta_t \circ \gamma_t$. According to the neighbourhood condition, $y \in U_{t-2} \cup U_{t-1} \cup U_t$, and therefore, y appears in $\beta_{t-1} \circ \gamma_{t-1} \circ \gamma_t$. Analogous to the previous case, $\beta_{t-1} \circ \gamma_{t-1} \circ \gamma_t$ is a k-layout for $G[U_{t-2} \cup U_{t-1} \cup U_t]$, thus, x and y appear exactly once in $\beta_{t-1} \circ \gamma_{t-1} \circ \gamma_t$, $\mathrm{d}_{\beta_{t-1} \circ \gamma_{t-1} \circ \gamma_t}(x,y) \leq k$, and so, $\mathrm{d}_\delta(x,y) \leq \pi(y) - \pi(x) \leq k$. We conclude that pairs of adjacent vertices appear at distance at most k in δ, so that δ is a k-layout for G.

For the second statement of the lemma, let β be a (k,l)-layout for G. We partition β into portions. Note that $r \geq 2$. For $0 \leq i \leq r-2$, let $\gamma_{i+1} =_{\mathrm{def}} \langle \beta(ik+1), \ldots, \beta(ik+k) \rangle$ and $\gamma_r =_{\mathrm{def}} \langle \beta(rk-k+1), \ldots, \beta(n) \rangle$ and $\gamma_0 =_{\mathrm{def}} \langle\rangle$. Note that in case $rk = n$, our definition means $\gamma_r = \langle\rangle$. For every $0 \leq i \leq r$, let U_i be the set of vertices that appear in γ_i. We show that $\beta = \gamma_0 \circ \cdots \circ \gamma_r$ corresponds to a directed S,T-path of $\mathrm{aux}(G,k,l)$, that we explicitly construct. To show the existence of the path, we have to check a series of properties. First, observe that $U_1 \cup \{\gamma_2(1)\} = B_\beta(\gamma_1(1), k)$, and $U_{i-1} \cup U_i \cup \{\gamma_{i+1}(1)\} = B_\beta(\gamma_i(1), k)$ for every $2 \leq i \leq r-2$, and $U_{r-2} \cup U_{r-1} \cup \{\gamma_r(1)\} = B_\beta(\gamma_{r-1}(1), k)$ and $U_{r-1} \cup U_r = B_\beta(\gamma_r(1), k)$ in case $U_r \neq \emptyset$, and $U_{r-2} \cup U_{r-1} = B_\beta(\gamma_{r-1}(1), k)$ in case $U_r = \emptyset$. Thus, according to the definition of (k,l)-layouts, it follows for every $1 \leq i \leq r$ where $U_i \neq \emptyset$ that $B_\beta(\gamma_i(1), k) \subseteq B_G(\gamma_i(1), l)$. It follows that $\mathrm{aux}(G,k,l)$ has the following vertices:

- a start vertex $u^{(1)}$ with label (γ_0, γ_1)
- for every $2 \leq i \leq r-1$, a middle vertex $u^{(i)}$ with label (γ_{i-1}, γ_i)
- an end vertex $u^{(r)}$ with label (γ_{r-1}, γ_r).

We show that $(S, u^{(1)}, u^{(2)}, \ldots, u^{(r)}, T)$ is a directed path of $\mathrm{aux}(G,k,l)$. Since $u^{(1)}$ is a start vertex and $u^{(r)}$ is an end vertex, it follows that $(S, u^{(1)})$ and $(u^{(r)}, T)$ are arcs of $\mathrm{aux}(G,k,l)$. Let $1 \leq i < r$. We consider the vertex pair $u^{(i)}, u^{(i+1)}$. The labels of $u^{(i)}$ and $u^{(i+1)}$ are (γ_{i-1}, γ_i) and (γ_i, γ_{i+1}), respectively. Then, the following holds:

- $u^{(i)}$ is not end vertex and $u^{(i+1)}$ is not start vertex
- $\gamma_i = \gamma_i$
- $\gamma_{i-1} \circ \gamma_i \circ \gamma_{i+1}$ is a k-layout for $G[U_{i-1} \cup U_i \cup U_{i+1}]$, since it is a sub-layout of a k-layout for G.

It remains to consider the neighbourhood and distance conditions. Let $w \in U_i$ and let $y \in N_G(w)$. Since β is a k-layout for G, $\mathrm{d}_\beta(w,y) \leq k$, which means that $y \in U_{i-1} \cup U_i \cup U_{i+1}$. If $u^{(i+1)}$ is an end vertex, i.e., $i+1 = r$, and if $\gamma_r \neq \langle\rangle$ then for every $w \in U_r$ and every $y \in N_G(w)$, it analogously follows that $\mathrm{d}_\beta(w,y) \leq k$ implies $y \in U_{r-1} \cup U_r$. This proves the neighbourhood condition. For the distance condition, let $w \in U_{i-1} \cup U_i \cup U_{i+1}$. It holds that $B_{\gamma_{i-1}\circ\gamma_i\circ\gamma_{i+1}}(w,k) \subseteq B_\gamma(w,k) \subseteq B_G(w,l)$, where the latter inclusion holds due to the properties of (k,l)-layouts. We conclude that $(u^{(i)}, u^{(i+1)})$ is an arc of $\mathrm{aux}(G,k,l)$. Hence, $\mathrm{aux}(G,k,l)$ contains a directed S,T-path of length at most $r+1$.

For the converse, let $(S, u^{(1)}, \ldots, u^{(t)}, T)$ be a directed S,T-path of $\mathrm{aux}(G,k,l)$ of length at most $r+1$. Let (β_i, γ_i) be the layout pair label of $u^{(i)}$ for every $1 \leq i \leq t$. Let $\gamma_0 =_{\mathrm{def}} \gamma_{t+1} =_{\mathrm{def}} \langle\rangle$. Note that $\gamma_{i-1} = \beta_i$ for every $1 \leq i \leq t$. We show that $\delta =_{\mathrm{def}} \gamma_0 \circ \cdots \circ \gamma_{t+1}$ is a (k,l)-layout for G. Let U_i be the set of vertices that appear in γ_i for every $0 \leq i \leq t+1$. It is important to note that $|U_0| = |U_{t+1}| = 0$ and $|U_t| \leq k-1$ and $|U_i| = k$ for every $1 \leq i \leq t-1$. Then, $|U_1| + \cdots + |U_t| < tk \leq rk$. According to the proof of the first statement, every vertex of G appears at least once in δ. Thus, $(r-1)k \leq n \leq |U_1| + \cdots + |U_t| < rk$. We show that every vertex of G appear exactly once in δ. Remember that for all $1 \leq i \leq t$, $\gamma_{i-1} \circ \gamma_i \circ \gamma_{i+1}$ is a layout for $G[U_{i-1} \cup U_i \cup U_{i+1}]$. It follows that for every choice of i,j such that $i < j$ and $\delta(i) = \delta(j)$, $j - i > 2k$ must hold. Assume that there is a vertex x of G such that $i < j$ with $\delta(i) = \delta(j) = x$ exist. Let $1 \leq p \leq t$ be such that position j in δ is in γ_p. Due to the neighbourhood condition, it holds that all neighbours of x appear in $\gamma_{p-1} \circ \gamma_p \circ \gamma_{p+1}$ and, for the occurrence of x in position i, all neighbours of x appear in $\gamma_1 \circ \cdots \circ \gamma_{p-2}$. If there is a neighbour y of x and a position i' in δ with $\delta(i') = y$ and $|i - i'| \leq k$ and $|j - i'| \leq k$ then $j - i \leq 2k$, a contradiction. Thus, every neighbour of x appears at least two times in δ. Since G is connected, it directly follows that every vertex of G appears at least two times in δ, which yields a contradiction to the length of δ of less than $n + k < 2n$. We conclude that δ is a layout for G, and due to the proof of the first statement, δ is a k-layout for G.

To complete the proof, let x, y be a vertex pair of G such that $\mathrm{d}_\delta(x,y) \leq k$. Then, there is $1 \leq p \leq t$ such that x appears in γ_p and y appears in $\gamma_{p-1} \circ \gamma_p \circ \gamma_{p+1}$. Due to the distance condition, it directly follows that $y \in B_G(x,l)$. We conclude that δ is a (k,l)-layout for G. ∎

We only remark here that the proof of Lemma 3.1 shows even stronger results. For the first statement, the proof shows how to extract a k-layout from an arbitrary directed S,T-path, and for the second statement, the proof actually shows a 1-to-1 correspondence between the (k,l)-layouts of G and the short directed S,T-paths of $\mathrm{aux}(G,k,l)$.

In the second part of the section, we consider running-time aspects. We will show that (the existence of) a short directed S,T-path of the auxiliary digraph can be determined very efficiently under reasonable assumptions. The next lemma establishes the "reasonable assumptions".

Lemma 3.2 *Let G be a graph, let $k,d \geq 0$ be integers and let x be a vertex of G. If* $\mathrm{bw}(G) \leq k$ *then $|B_G(x,d)| \leq 2kd+1$.*

Proof. Let β be a k-layout for G, that exists due to the assumption $\mathrm{bw}(G) \leq k$. Let $1 \leq i \leq j \leq i' \leq |V(G)|$ be such that $\beta(j) = x$ and i and i' are smallest and largest, respectively, satisfying $\beta(i), \beta(i') \in B_G(x,d)$. Since $d \geq \mathrm{d}_G(x,\beta(i)) \geq \frac{1}{k}(j-i)$, it holds that $\mathrm{d}_\beta(x,\beta(i)) \leq kd$. Analogously, $\mathrm{d}_\beta(x,\beta(i')) \leq kd$. With the choice of i and i', it follows for every $1 \leq j' \leq |V(G)|$, if $\beta(j') \in B_G(x,d)$ then $i \leq j' \leq i'$. Since $i'-i \leq 2kd$, we conclude the cardinality bound of the lemma. ■

Hence, the result of Lemma 3.2 provides a sufficient condition for a graph to have large bandwidth. Motivated by this result, for integers $k,l \geq 0$, we say that a vertex x of a graph G *has k-few l-close neighbours* if $|B_G(x,d)| \leq 2kd+1$ for every $0 \leq d \leq l$. Thus, if a graph has a k-layout then each vertex can have only k-few l-close neighbours.

Lemma 3.3 *Let $k,l \geq 1$. For every connected graph G on at least $k+1$ vertices whose vertices have only k-few l-close neighbours, the vertices and arcs of* $\mathrm{aux}(G,k,l)$ *can be listed in $n \cdot 2^{\mathcal{O}(k \log kl)}$ time, where n is the number of vertices of G.*

Proof. Let G be a connected graph on n vertices, where $n \geq k+1$. Let each vertex of G have only k-few l-close neighbours. We list the vertices and arcs of $\mathrm{aux}(G,k,l)$ separately. A vertex of $\mathrm{aux}(G,k,l)$ is uniquely determined by its label, so it suffices to list the corresponding layout pairs. Let a be a vertex of G. We determine the number of layout pairs with anchor a and distinguish between the three different vertex types. Remember that $|B_G(a,l)| \leq 2kl+1$ due to the assumptions about the number of neighbours. Let r be smallest such that $rk > n$, and let $d =_{\mathrm{def}} n-(r-2)k$. Then, the numbers of the different vertex types are:

- *start vertices:* at most $k! \cdot \binom{2kl+1}{k} \leq (2k)! \cdot (2kl+1)^k$
- *middle vertices:* at most $(2k)! \cdot \binom{2kl+1}{2k} \leq (2k)! \cdot (2kl+1)^{2k}$
- *end vertices:* at most $d! \cdot \binom{2kl+1}{d} \leq (2k)! \cdot (2kl+1)^{2k}$.

Including S and T, the total number of vertices of $\mathrm{aux}(G,k,l)$ is bounded above by $n \cdot 2^{\mathcal{O}(k \log kl)}$. And since a layout pair requires at most $2k$ representation space by simply listing the vertices according to the layouts, the vertices of $\mathrm{aux}(G,k,l)$ can be listed in time $n \cdot 2^{\mathcal{O}(k \log kl)}$.

Next, we determine the arcs of $\mathrm{aux}(G,k,l)$. It suffices to list the out-neighbours of each vertex. Since the out-neighbours of S are the start vertices, and since the unique out-neighbour of an end vertex is T, it suffices to restrict to out-neighbours of start and middle vertices. It even suffices to determine a very loose upper bound. Let x be a start or middle vertex with anchor a. Let y be a vertex of $\mathrm{aux}(G,k,l)$ with anchor b. If (x,y) is an arc of $\mathrm{aux}(G,k,l)$ then $b \in B_G(a,l)$ due to the distance condition. Since a has only k-few l-close neighbours, it follows that at most $(2kl+1) \cdot 2^{\mathcal{O}(k \log kl)} \leq 2^{\mathcal{O}(k \log kl)}$ vertices of $\mathrm{aux}(G,k,l)$ are potential out-neighbours of x. These candidate vertices (including their layout pair labels) can be listed in $2^{\mathcal{O}(k \log kl)}$ time. Checking whether a vertex pair satisfies the requested properties can be done in $\mathcal{O}(k^3l^2)$ time. This mainly relies on the fact that each vertex of G can have at most $2k$ neighbours, since $N_G(x) \cup \{x\} = B_G(x,1)$, and thus, checking the neighbourhood condition sums up to a total $\mathcal{O}(k^2)$ time, and every vertex has only k-few l-close neighbours, that can be listed in $\mathcal{O}(k^2l^2)$ time, and checking the distance condition therefore requires a total $\mathcal{O}(k^3l^2)$ time. Hence, every vertex of $\mathrm{aux}(G,k,l)$ has at most $2^{\mathcal{O}(k \log kl)}$ out-neighbours, that can be listed in time $2^{\mathcal{O}(k \log kl)}$.

We summarise: $\mathrm{aux}(G,k,l)$ has at most $n \cdot 2^{\mathcal{O}(k \log kl)}$ vertices and at most $n \cdot 2^{\mathcal{O}(k \log kl)} \cdot 2^{\mathcal{O}(k \log kl)}$ arcs, and the vertices and arcs can be listed in total $n \cdot 2^{\mathcal{O}(k \log kl)}$ time. ∎

Corollary 3.4 *Let $k,l \geq 1$. For every connected graph G on at least $k+1$ vertices whose vertices have only k-few l-close neighbours, the length of a shortest directed S,T-path of $\mathrm{aux}(G,k,l)$ can be determined in $n \cdot 2^{\mathcal{O}(k \log kl)}$ time, where n is the number of vertices of G.*

Proof. Let G be a connected graph on n vertices of the required type. Due to Lemma 3.3, the vertices and arcs of $\mathrm{aux}(G,k,l)$ can be listed in $n \cdot 2^{\mathcal{O}(k \log kl)}$ time. Constructing a shortest directed S,T-path can be done in time linear in the numbers of vertices and arcs of $\mathrm{aux}(G,k,l)$ by a straightforward breadth first search. ∎

Combining the results of Lemma 3.1 and Corollary 3.4, the bandwidth problem is fixed-parameter tractable on arbitrary graphs when parametrised by k and l.

4 An FPT-algorithm for AT-free graphs

The results of Section 3 combine into an algorithm for deciding Bandwidth efficiently for graphs of bounded structure. To be precise, deciding the existence of (k,l)-layouts can be done efficiently when parametrised by k and l. In this section, we will consider the case of AT-free graphs and show that l can be bounded by a simple (linear) function in k. This will establish an FPT-algorithm for deciding Bandwidth on AT-free graphs when parametrised by the bandwidth k. Our bound on l and thus our algorithm relies on structural results about minimal triangulations of AT-free graphs.

A graph H is an *interval graph* if its vertices can be assigned closed intervals of the real line such that two vertices of H are adjacent if and only if the assigned intervals have a non-empty intersection. Such a family of closed intervals is called *interval model* for the graph. An interval graph is a *proper interval graph* if it has an interval model such that no interval completely contains another interval. Let G be a graph. An *interval completion* of G is an interval graph H with vertex set $V(G)$ and with $E(G) \subseteq E(H)$. If there is no interval completion H' of G with $E(H') \subset E(H)$ then H is a *minimal* interval completion of G. Analogously, a *proper interval completion* of G is a proper interval graph H with vertex set $V(G)$ and with $E(G) \subseteq E(H)$.

Lemma 4.1 ([13]) *For any integer $k \geq 0$, a graph G has a proper interval completion H that does not contain a clique of size $k+2$ if and only if $bw(G) \leq k$.*

Let β be a layout for a graph G. From β and G, we obtain the *fill graph* of (G, β), $\text{fi}(G, \beta)$, by adding edges to G. Formally, $\text{fi}(G, \beta)$ has vertex set $V(G)$, and xy is an edge of $\text{fi}(G, \beta)$ if and only if there is an edge u, v of G such that $u \preccurlyeq_\beta x \prec_\beta y \preccurlyeq_\beta v$. It is clear that G is a subgraph of $\text{fi}(G, \beta)$, and it follows from a layout characterisation of proper interval graphs in [17] that $\text{fi}(G, \beta)$ is a proper interval graph. Hence, $\text{fi}(G, \beta)$ is a proper interval completion of G.

Lemma 4.2 *For every graph G, there is a minimal interval completion H of G with* $\text{bw}(G) = \text{bw}(H)$.

Proof. Let β be a k-layout for G. Observe that β is a k-layout also for $H' =_{\text{def}} \text{fi}(G, \beta)$, which means that $\text{bw}(H') \leq k$. Since H' is a proper interval graph and therefore an interval completion of G, there is a minimal interval completion H of G such that $E(H) \subseteq E(H')$. It follows that $\text{bw}(G) \leq \text{bw}(H) \leq \text{bw}(H') \leq k$. By choosing k smallest possible, the claimed result follows. ■

Interval graphs can be characterised by admitting a special vertex ordering. A graph G is an interval graph if and only if G has a vertex ordering σ such that for every vertex triple u, v, w of G with $u \prec_\sigma v \prec_\sigma w$, $uw \in E(G)$ implies $vw \in E(G)$ [21]. Such a vertex ordering is called *interval ordering.*

Theorem 4.3 ([8]) *Let H be an interval graph with interval ordering σ. Let $b =_{\text{def}} \text{bw}(H)$. There is a b-layout β for H such that for every pair u, v of non-adjacent vertices of H, $u \prec_\sigma v$ implies $u \prec_\beta v$.*

Lemma 4.4 *Let H be a connected interval graph. Let $b =_{\text{def}} \text{bw}(H)$. There is a b-layout β for H with the property: for every vertex pair u, v of H, $\text{d}_H(u, v) \leq \text{d}_\beta(u, v) + 2$.*

Proof. Let σ be an interval ordering for H. Let β be a b-layout for H with the property of Theorem 4.3 with respect to σ. We show that β satisfies the claim of the lemma. Let u, v be a vertex pair of H. If $\text{d}_H(u, v) \leq 3$ then the lemma trivially holds. Let $\text{d}_H(u, v) \geq 4$. Without loss of generality, we can assume $u \prec_\sigma v$. Let $P = (x_0, \ldots, x_r)$ be a u, v-path of H that has smallest length. If there is $1 \leq i \leq r-2$ with $x_{i-1} \prec_\sigma v \prec_\sigma x_i$ then v is adjacent to x_i by the properties of interval orderings, showing that $(u, x_1, \ldots, x_i, v)$ is a u, v-path of length smaller than r, a contradiction to the choice of P. If there is $2 \leq i \leq r$ with $x_{i-1} \prec_\sigma u \prec_\sigma x_i$ then u and x_i are adjacent, which again yields a contradiction to the choice of P. Thus, $u \prec_\sigma x_i \prec_\sigma v$ for all $1 \leq i \leq r-2$. Since $x_2, \ldots, x_{r-2}$ are non-adjacent to u and v by the choice of P as of smallest length possible, Theorem 4.3 for β implies that $u \prec_\beta x_i \prec_\beta v$ for all $2 \leq i \leq r-2$. Hence, $\text{d}_\beta(u, v) \geq r - 2 = \text{d}_H(u, v) - 2$. ■

The *square* of a graph G, denoted as G^2, is the graph on vertex set $V(G)$ and with edge set $\{uv : 1 \leq \text{d}_G(u, v) \leq 2\}$. Combining the results of [16] and [18], we obtain the following:

Theorem 4.5 ([16, 18]) *Let G be an AT-free graph. For every minimal interval completion H of G, $H \subseteq G^2$.*

We combine the results of this section and finally prove the main lemma about the structure of balls of bounded radius in connected AT-free graphs.

Lemma 4.6 *Let G be a connected AT-free graph and let $k \geq \text{bw}(G)$. There is a k-layout β for G such that for every vertex x of G, $B_\beta(x,k) \subseteq B_G(x, 2k+4)$.*

Proof. Let H be a minimal interval completion of G such that $\text{bw}(H) = \text{bw}(G)$; H exists due to Lemma 4.2. As a corollary of Theorem 4.5, $\text{d}_H(u,v) \geq \frac{1}{2} \cdot \text{d}_G(u,v)$ for every vertex pair u,v of G. Let $b =_{\text{def}} \text{bw}(G)$ and let β be a b-layout for H with the property of Lemma 4.4. Then, for every vertex pair u,v of G, $\text{d}_\beta(u,v) + 2 \geq \text{d}_H(u,v) \geq \frac{1}{2} \cdot \text{d}_G(u,v)$, i.e., $\text{d}_\beta(u,v) \geq \frac{1}{2} \cdot \text{d}_G(u,v) - 2$. Now, let x,y be vertices of G. If $y \in B_\beta(x,k)$ then $\text{d}_G(x,y) \leq 2k+4$, which means that $y \in B_G(x, 2k+4)$. It remains to observe that β is a b-layout for G and so a k-layout for G. ∎

Corollary 4.7 *Let $k \geq 1$. A connected AT-free graph G has a k-layout if and only if G has a $(k, 2k+4)$-layout.*

Proof. Clearly, if β is a $(k, 2k+4)$-layout for G then β is a k-layout for G. Conversely, if G has a k-layout then $k \geq \text{bw}(G)$ and G has a $(k, 2k+4)$-layout due to Lemma 4.6. ∎

With the main results of this and the previous section, we conclude our FTP-algorithm for Bandwidth on AT-free graphs.

Theorem 4.8 *Given an AT-free graph G and an integer $k \geq 1$, it can be decided in time $n \cdot 2^{\mathcal{O}(k \log k)}$ whether $\text{bw}(G) \leq k$, where n is the number of vertices of G.*

Proof. It is first to observe that $\text{bw}(G) \leq k$ implies that $|E(G)| \leq kn$. This follows from the fact that no vertex of G can have more than $2k$ neighbours according to Lemma 3.2. As a first step, determine the connected components of G. This can be done in linear time, i.e., in time $\mathcal{O}(kn)$. We apply Corollary 4.7 and see that $\text{bw}(G) \leq k$ if and only if every connected component of G has a $(k, 2k+4)$-layout. If a vertex of G does not have only k-few $(2k+4)$-close neighbours then $\text{bw}(G) > k$ due to Lemma 3.2, and the algorithm rejects. Otherwise, we apply the algorithm of Corollary 3.4 to each connected component of G and decide the existence of a $(k, 2k+4)$-layout by applying Lemma 3.1. Since $\log k(2k+4) \leq 2 \log k + c$ for some constant c, we conclude the claim of the theorem. ∎

5 Concluding remarks

The *diameter* of a graph G, denoted as $\text{diam}(G)$, is the smallest value d such that $\text{d}_G(u,v) \leq d$ for every vertex pair u,v of G. If the diameter is bounded for a graph class then a result of the type of Corollary 4.7 holds, and we can obtain an analogue of Theorem 4.8. Graphs of bounded diameter are dense graphs. However, for such classes of dense graphs, the problem becomes trivial, as we show in the following.

Lemma 5.1 *Let $G = (V,E)$ be a connected graph. Then, $|V| \leq 1 + \text{diam}(G) \cdot \text{bw}(G)$.*

Proof. Let β be a k-layout for G for $k \geq 0$. Let $a =_{\text{def}} \beta(1)$ and $b =_{\text{def}} \beta(|V|)$. Then, $|V| - 1 = \mathrm{d}_\beta(a, b) \leq \mathrm{d}_G(a, b) \cdot k \leq \mathrm{diam}(G) \cdot k$. ■

The result of Lemma 5.1 implies for a graph class of bounded diameter that there is only a finite number of graphs of bounded bandwidth. Thus, for such graph classes, deciding whether $\mathrm{bw}(G) \leq k$ for given graph G is trivial when k is fixed. If k is part of the input, we can apply the currently best known exact algorithm for computing the bandwidth by Cygan and Pilipczuk, with running time $\mathcal{O}(4.473^n)$ [6], and obtain a $\mathcal{O}(4.473^{dk})$-time algorithm for deciding whether $\mathrm{bw}(G) \leq k$ for a given (connected) graph G with $\mathrm{diam}(G) \leq d$ and integer k. Examples of graph classes of bounded diameter are P_{d+1}-free graphs, in particular split graphs and cobipartite graphs, which are well-studied classes of P_5-free graphs. Note that Bandwidth is NP-complete when restricted to split and cobipartite graphs [15, 16].

An interesting corollary follows from the result of Lemma 3.1. Even though a graph G may not have a (k, l)-layout, it may still have a k-layout that is witnessed by a directed S, T-path of $\mathrm{aux}(G, k, l)$. Is there a graph class for which the size of l can be bounded from above such that the auxiliary digraph definitely has a directed S, T-path, without necessarily having such a path of short length?

We conclude with a few more open problems.

- Does there exist an FPT-algorithm for Bandwidth on AT-free graphs that has a running time of the form $2^{\mathcal{O}(k)} n^{\mathcal{O}(1)}$? An algorithm with this running time would be interesting even for cocomparability graphs.
- Can the bandwidth be FPT-approximated on trees? That is, is there an algorithm that given a tree T and an integer k, runs in time $f(k) n^{\mathcal{O}(1)}$ and either correctly answers that $\mathrm{bw}(T) > k$ or outputs a $g(k)$-layout for T for some function g?
- What is the parametrised complexity of Bandwidth on caterpillars with hairlength c, for fixed constant $c \geq 3$?

Acknowledgement

We would like to thank Fedor Fomin for insightful discussions around bandwidth.

References

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

[2] G. Blache, M. Karpinski, J. Wirtgen. On approximation intractability of the bandwidth problem. Technical report TR98-014, University of Bonn, 1997.

[3] H. L. Bodlaender, M. R. Fellows, M. T. Hallet. Beyond NP-completeness for problems of bounded width (extended abstract): hardness for the W hierarchy. *Proceedings of STOC 1994*, pages 449–458, ACM, 1994.

[4] A. Brandstädt, V. B. Le, J. Spinrad. *Graph Classes: A Survey*. SIAM, Philadelphia, 1999.

[5] D. G. Corneil, S. Olariu, L. Stewart. Asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics*, 10:399–430, 1997.

[6] M. Cygan and M. Pilipczuk. Exact and approximate bandwidth. *Proceedings of ICALP 2009*, LNCS, 5555:304–315, 2009.

[7] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, New York, 1999.

[8] P. Fishburn, P. Tanenbaum, A. Trenk. Linear discrepancy and bandwidth. *Order*, 18:237–245, 2001.

[9] M. R. Garey and D. S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[10] J. A. George and J. W. H. Liu. *Computer Solution of Large Sparse Positive Definite Systems*. Prentice-Hall, New Jersey, 1981.

[11] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer-Verlag, 2006.

[12] P. Heggernes, D. Kratsch, D. Meister. Bandwidth of bipartite permutation graphs in polynomial time. *Journal of Discrete Algorithms*, 7:533–544, 2009.

[13] H. Kaplan and R. Shamir. Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques. *SIAM Journal on Computing*, 25:540–561, 1996.

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

[15] T. Kloks, D. Kratsch, Y. Le Borgne, H. Müller. Bandwidth of split and circular permutation graphs. *Proceedings of WG 2000*, LNCS, 1928:243–254, 2000.

[16] T. Kloks, D. Kratsch, H. Müller. Approximating the bandwidth for AT-free graphs. *Journal of Algorithms*, 32:41–57, 1999.

[17] P. J. Looges and S. Olariu. Optimal greedy algorithms for indifference graphs. *Computers & Mathematics with Applications*, 25:15–25, 1993.

[18] R. Möhring. Triangulating graphs without asteroidal triples. *Discrete Applied Mathematics*, 64:281–287, 1996.

[19] B. Monien. The bandwidth minimization problem for caterpillars with hair length 3 is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 7:505–512, 1986.

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

[21] S. Olariu. An optimal greedy heuristic to color interval graphs. *Information Processing Letters*, 37:21–25, 1991.

[22] C. Papadimitriou. The NP-completeness of the bandwidth minimization problem. *Computing*, 16:263–270, 1976.

[23] J. B. Saxe. Dynamic programming algorithms for recognizing small bandwidth graphs in polynomial time. *SIAM Journal on Algebraic and Discrete Methods*, 1:363–369, 1980.

[24] J. H. Yan. The bandwidth problem in cographs. *Tamsui Oxford Journal of Mathematical Sciences*, 13:31–36, 1997.