Structural Parameterizations for Boxicity

Henning Bruhn[1(✉)], Morgan Chopin[1], Felix Joos[1], and Oliver Schaudt[2]

[1] Institut Für Optimierung Und Operations Research, Universität Ulm, HelmholtzstraßE 18, 89081 Ulm, Germany
{henning.bruhn,morgan.chopin,felix.joos}@uni-ulm.de
[2] Institut Für Informatik, Universität Zu Köln, Weyertal 80, 50931 Köln, Germany
schaudto@uni-koeln.de

Abstract. The boxicity of a graph G is the least integer d such that G has an intersection model of axis-aligned d-dimensional boxes. Boxicity, the problem of deciding whether a given graph G has boxicity at most d, is NP-complete for every fixed $d \geq 2$. We show that Boxicity is fixed-parameter tractable when parameterized by the cluster vertex deletion number of the input graph. This generalizes the result of Adiga et al. [4], that Boxicity is fixed-parameter tractable in the vertex cover number. Moreover, we show that Boxicity admits an additive 1-approximation when parameterized by the pathwidth of the input graph.

Finally, we provide evidence in favor of a conjecture of Adiga et al. [4] that Boxicity remains NP-complete even on graphs of constant treewidth.

1 Introduction

Every graph G can be represented as an intersection graph of axis-aligned boxes in $\mathbb{R}^d$, provided d is large enough. The *boxicity* of G, denoted by $\text{box}(G)$, introduced by Roberts [21], is the smallest dimension d for which this is possible. We denote the corresponding decision problem by Boxicity: given G and $d \in \mathbb{N}$, determine whether G has boxicity at most d.

Boxicity has received a fair amount of attention. This is partially due to the wider context of graph representations, but also because graphs of low boxicity are interesting from an algorithmic point of view. While many hard problems remain so for graphs of bounded boxicity, some become solvable in polynomial time, notably max-weighted clique (as observed by Spinrad [23, p. 36]).

Cozzens [13] showed that Boxicity is NP-complete. To cope with this hardness result, several authors [1,4,18] studied the parameterized complexity of Boxicity. Since the problem remains NP-complete for constant $d \geq 2$ (Yannakakis [25] and Kratochvíl [20]), boxicity itself is ruled out as parameter. Instead more structural parameters have been considered. Our work follows this line. We prove:

Theorem 1. Boxicity *is fixed-parameter tractable when parameterized by cluster vertex deletion number.*

D. Kratsch and I. Todinca (Eds.): WG 2014, LNCS 8747, pp. 117–128, 2014.
DOI: 10.1007/978-3-319-12340-0_10

The *cluster vertex deletion* number is the minimum number of vertices that have to be deleted to get a disjoint union of complete graphs or *cluster graph*. As discussed by Doucha and Kratochvíl [15] cluster vertex deletion is an intermediate parameterization between vertex cover and cliquewidth. A *d-box representation* of a graph G is a representation of G as intersection graph of axis-aligned boxes in $\mathbb{R}^d$.

Theorem 2. *Finding a d-box representation of G such that* $d \leq \text{box}(G) + 1$ *can be done in* $f(\text{pw}(G)) \cdot |V(G)|$ *time where* $\text{pw}(G)$ *is the pathwidth of G.*

A natural parameter for Boxicity is the treewidth $\text{tw}(G)$ of a graph G, in particular as Chandran and Sivadasan [11] proved that $\text{box}(G) \leq \text{tw}(G) + 2$. However, Adiga, Chitnis and Saurabh [4] conjecture that Boxicity is NP-complete on graphs of bounded treewidth. Our last result provides evidence in favor of this conjecture. For this, we mention the observation of Roberts [21] that a graph G has boxicity d if and only if G can be expressed as the intersection of d interval graphs.

Theorem 3. *There is an infinite family of graphs* $\mathcal{G}$ *of boxicity* 2 *and bandwidth* $\mathcal{O}(1)$ *such that, among any pair of interval graphs whose intersection is* $G \in \mathcal{G}$*, at least one has treewidth* $\Omega(|V(G)|)$*.*

Why do we see the result as evidence? An algorithm solving Boxicity on graphs of bounded treewidth (or even stronger, of bounded bandwidth) is likely to exploit the local structure of the graph in order to make dynamic programming work. Yet, Theorem 3 implies that this locality may be lost in some dimensions, which constitutes a serious obstacle for any dynamic programming based approach. We discuss this in more detail in Sect. 5.

Figure 1 summarizes previously known parameterized complexity results on boxicity along with those obtained in this article. Adiga et al. [4] initiated this line of research when they parameterized Boxicity by the minimal size k of a *vertex cover* in order to give an $2^{O(2^k k^2)} \cdot n$-time algorithm, where n denotes the number of vertices of the input graph, as usual. This result had already been observed earlier by Fellows et al. [17] in the context of well-quasi orders of certain graph classes. Adiga et al. [4] also described an approximation algorithm that, in time $2^{O(k^2 \log k)} \cdot n$, returns a box representation of at most $\text{box}(G) + 1$ dimensions. Both results were extended by Ganian [18] to the less restrictive parameter *twin cover*. Our Theorem 1 includes Ganian's.

Other structural parameters that were considered by Adiga et al. [4] for parameterized approximation algorithms are the size of a *feedback vertex set* – the minimum number of vertices that need to be deleted to obtain a forest – and *maximum leaf number* – the maximum number of leaves in a spanning tree of the graph. They proved that finding a d-box representation of a graph G such that $d \leq 2\text{box}(G)+2$ (resp. $d \leq \text{box}(G)+2$) can be done in $f(k) \cdot |V(G)|^{O(1)}$ time (resp. $2^{O(k^3 \log k)} \cdot |V(G)|^{O(1)}$ time) where k is the size of a feedback vertex set (resp. maximum leaf number). In [1], Adiga, Babu, and Chandran generalized these approximation algorithms to parameters of the type "distance to $\mathcal{C}$", where

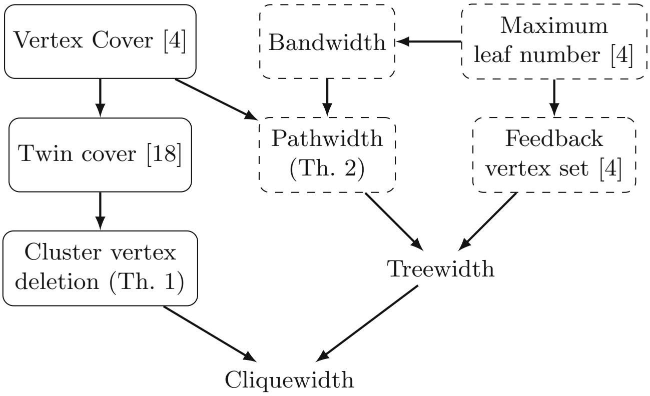

Fig. 1. Navigation map through our parameterized complexity results for BOXICITY. An arc from a parameter k_2 to a parameter k_1 means that there exists some function h such that $k_1 \leq h(k_2)$. A rectangle means fixed-parameter tractability for this parameter and a dashed rectangle means an approximation algorithm with running time $f(k) \cdot n^{O(1)}$ is known.

$\mathcal{C}$ is any graph class of bounded boxicity. More precisely, the parameter measures the minimum number of vertices whose deletion results in a graph that belongs $\mathcal{C}$.

The algorithm of Theorem 2 generalizes the approximation algorithm for the parameter vertex cover number, and improves the guarantee bound of the approximation algorithm for the parameter maximum leaf number.

There is merit in studying approximation algorithms from a parameterized perspective: not only is BOXICITY NP-complete, but the associated minimization problem cannot be approximated in polynomial time within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$ even when the input is restricted to bipartite, co-bipartite or split graphs (provided NP$\neq$ZPP). This is a result due to Chalermsook et al. [10] using the hardness reduction of Adiga, Bhowmick and Chandran [2]. There is, however, an approximation algorithm with factor $o(n)$ for general graphs; see Adiga et al. [1].

While Roberts [21] was the first to study the boxicity parameter, he was hardly the first to consider box representations of graphs. Already in 1948 Bielecki [6] asked, here phrased in modern terminology, whether triangle-free graphs of boxicity ≤ 2 had bounded chromatic number. This was answered affirmatively by Asplund and Grünbaum in [5]. Kostochka [19] treats this question in a much more general setting.

Following Roberts who proved that $\text{box}(G) \leq \frac{n}{2}$, other authors obtained bounds for boxicity. Adiga et al. [3], for instance, showed that $\text{box}(G) \leq \Delta(G) \log^2 \Delta(G)$, while Scheinerman [22] established that every outerplanar graph has boxicity at most two. This, in turn, was extended by Thomassen [24], who showed that planar graphs have boxicity at most three.

In the next section, we will give formal definitions of the necessary concepts for this article. We prove our main results in Sects. 3–5. Finally, we discuss the impact and limitations of our results in Sect. 6, where we also outline some future directions for research. Due to space limitation, some proofs are deferred to a full version [8].

2 Preliminaries

Graph terminology. We follow the notation of Diestel [14], where also all basic definitions concerning graphs may be found.

Let X be some finite set. With a slight abuse of notation, we consider a collection $I = ([\ell_v, r_v])_{v \in X}$ of closed intervals in the real line to be an *interval graph*: I has vertex set X, and two of its vertices u and v are adjacent if and only if the corresponding intervals $[\ell_u, r_u]$ and $[\ell_v, r_v]$ intersect. By perturbing the endpoints of the intervals we can ensure that no two intervals have a common endpoint, and that for every interval the left endpoint is distinct from the right endpoint. We always tacitly assume the intervals to be of that form.

The *bandwidth* of a graph G, say with vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$, is the least number k for which the vertices of G can be labeled with distinct integers $\ell(v_i)$ such that $k = \max\{|\ell(v_i) - \ell(v_j)| : v_i v_j \in E\}$. Equivalently, it is the least integer k for which the vertices of G can be placed at distinct integer points on the real line such that the length of the longest edge is at most k. We denote the bandwidth of a graph G by $\mathrm{bw}(G)$.

The *pathwidth* of a graph G, denoted $\mathrm{pw}(G)$, is the minimum size of the largest clique of any interval supergraph of G, minus 1.

The *treewidth* of a graph G, denoted $\mathrm{tw}(G)$, is the minimum size of the largest clique of any chordal supergraph of G, minus 1.

For the purpose of our paper it is important to remark that for every graph G we have $\mathrm{tw}(G) \leq \mathrm{pw}(G) \leq \mathrm{bw}(G)$.

Parameterized complexity. A decision problem parameterized by a problem-specific parameter k is called *fixed-parameter tractable* if there exists an algorithm that solves it in time $f(k) \cdot n^{O(1)}$, where n is the instance size. The function f is typically super-polynomial and only depends on k. One of the main tools to design such algorithms is the *kernelization* technique. A kernelization algorithm transforms in polynomial time an instance I of a given problem parameterized by k into an equivalent instance I' of the same problem parameterized by $k' \leq k$ such that the size of I' is bounded by $g(k)$ for some computable function g. The instance I' is called a *kernel* of size $g(k)$. The following folklore result is well known.

Theorem 4. *A parameterized problem P is fixed-parameter tractable if and only if P has a kernel.*

In the remainder of this paper, the kernel size is expressed in terms of the number of vertices.

For more background on parameterized complexity the reader is referred to Downey and Fellows [16].

Problem definition. We call an *axis-aligned d-dimensional box* (or *d-box*) a Cartesian product of d closed real intervals. A *d-box representation* of a graph G is a mapping that maps every vertex $v \in V(G)$ to a d-box B_v such that two vertices $u, v \in V(G)$ are adjacent if and only if their associated boxes have a non-empty intersection. The *boxicity* of G, denoted by $\text{box}(G)$, is the minimum integer d such that G admits a d-box representation. We consider the following problem.

BOXICITY **Input:** A graph G and an integer d. **Question:** Is $\text{box}(G) \leq d$?

Given a d-box representation of G, we denote by $[\ell_i(v), r_i(v)]$ the interval representing v in the i-th dimension.

Throughout the article, we make frequent use of the reformulation of boxicity in terms of interval graphs:

Theorem 5 (Roberts [21]). *The boxicity of a graph G is equal to the smallest integer d so that G can be expressed as the intersection of d interval graphs.*

3 Cluster Vertex Deletion

Theorem 1 follows immediately from the following lemma:

Lemma 1. BOXICITY *admits a kernel of at most $k^{2^{O(k)}}$ vertices, where k is the cluster vertex deletion number of the input graph.*

In the course of this section, we present the sequence of lemmas that are needed to prove the above kernelization result.

Two adjacent vertices u, v in a graph G are *true twins* if u and v have the same neighbourhoods in $G - \{u, v\}$. As observed by Ganian [18], deleting one of two true twins does not change the boxicity.

Lemma 2. *Let u, v be true twins of a graph G. Then* $\text{box}(G) = \text{box}(G - u)$.

We remark, without proof, that there is also a reduction for false twins (those that are non-adjacent): if there are at least three of them, then one may be deleted without changing the boxicity. We will not, however, make use of this observation.

Recall that a *cluster graph* is the disjoint union of complete graphs, called *clusters*. In what follows, we implicitly identify a cluster with its vertex set.

Let $G - X$ be a cluster graph for some $X \subseteq V(G)$. We call two clusters C, C' of $G - X$ *equivalent* if there is a bijection $C \to C'$, $v \mapsto v'$, such that $N_G(v) \cap X = N_G(v') \cap X$. Observe that, if $G - X$ has no true twins, then two clusters C and C' are equivalent if and only if $\{N_G(u) \cap X : u \in C\} = \{N_G(v) \cap X : v \in C'\}$.

Lemma 3. *Let G be a graph without true twins, and let X be a set of k vertices so that $G - X$ is a cluster graph. Then every cluster in $G - X$ contains at most 2^k vertices.*

We also need the following result.

Theorem 6 (Chandran and Sivadasan [11]). *It holds that* $\mathrm{box}(G) \le \mathrm{tw}(G) + 2$ *for any graph* G.

In particular, $\mathrm{box}(G) \le \mathrm{pw}(G) + 2$ for any graph G.

Lemma 4. *Let G be a graph without true twins, and let X be a set of k vertices so that $G - X$ is a cluster graph. Moreover, let $\mathcal{D}$ be an equivalence class of clusters with $|\mathcal{D}| \ge 2(2k+2)^{2^{k+1}(2^k+k+1)}$. For every $C^* \in \mathcal{D}$, $\mathrm{box}(G) = \mathrm{box}(G - C^*)$.*

Proof. As deleting vertices may only decrease the boxicity, it suffices to prove that $\mathrm{box}(G) \le \mathrm{box}(G - C^*)$.

Set $H = G - C^*$, $d = \mathrm{box}(H)$, $k = |X|$ and $\mathcal{C} = \mathcal{D} \setminus \{C^*\}$. We claim that

$$d = \mathrm{box}(H) \le 2^k + k + 1. \tag{1}$$

Indeed, define a path decomposition with a bag W_C for every cluster C of $H - X$ such that $W_C = X \cup C$. This gives a path decomposition of H with width at most $k + 2^k - 1$, by Lemma 3. Theorem 6 now implies (1).

For the sake of simplicity, let us introduce the following notions. Fix a d-box representation of H. The set of *corners* of a box of a vertex is the Cartesian product $\times_{i=1}^{d}\{\ell_i(v), r_i(v)\}$. By rescaling every dimension, we can ensure that every endpoint of an interval of a vertex in X lies in $\{1, 2, \dots, 2k\}$. Thus every corner of a box of X lies in the grid $\{1, 2, \dots, 2k\}^d$. We may moreover assume that every other box of H is contained in $[0, 2k+1]^d$. Points of $\{0, 1, \dots, 2k+1\}^d$ are called *grid points*, and any set $[z_1, z_1+1] \times \ldots \times [z_d, z_d+1]$, where $z_i \in \{0, \dots, 2k\}$, is a *grid cell*. In each dimension i we say that the grid induces the *grid intervals* $[0, 1]$, $[1, 2]$, $\dots$, $[2k, 2k+1]$. A box of a vertex in $H - X$ is a *cluster box*.

By perturbing the boxes slightly we may always assume that

if s is a corner of a cluster box of a cluster C of $H - X$, and if t is a corner of the box of any vertex $z \in V(H - C)$ then $s_i \ne t_i$ for all dimensions $i = 1, \dots, d$. (2)

Moreover, we may assume that any corner of a cluster box lies in the interior of a grid cell. A cluster box that does not contain any grid point is called a *thin box*.

We concentrate on *thin* clusters, that is, clusters that consist of thin boxes only. We claim that

$$\textit{at least } (2k+2)^{2^{k+1}(2^k+k+1)} \textit{ clusters in } \mathcal{C} \textit{ are thin.} \tag{3}$$

To prove this claim, observe that no grid point lies in a cluster box of two different clusters as then two vertices in distinct clusters would be adjacent. Thus, there is at most one cluster per grid point so that one of its cluster boxes contains the grid point. As, by (1), there are $(2k+2)^d \le (2k+2)^{2^k+2k+1}$ grid points, it follows that $\mathcal{C}$ has at least $|\mathcal{C}| - (2k+2)^{2^k+2k+1} \ge (2k+2)^{2^{k+1}(2^k+k+1)}$ thin clusters.

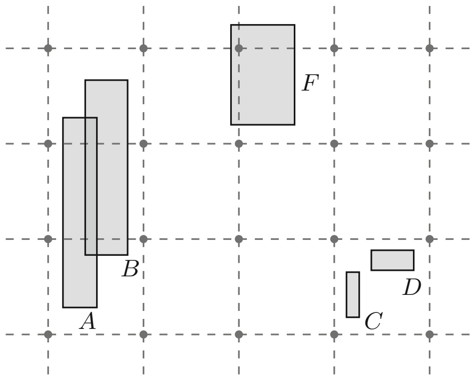

Fig. 2. Boxes A, B are in the same position, as are C and D; F is not thin.

We say that two cluster boxes B and B' are in the *same position* if every grid cell containing a corner of B also contains a corner of B' and vice versa (see Fig. 2). Note that if two vertices $v, v' \in V(H) - X$ have boxes in the same position then $N_H(v) \cap X = N_H(v') \cap X$. (Here we use the fact that cluster boxes have their corner strictly in the interior of grid cells).

For every cluster $C \in \mathcal{C}$ we fix a point $p(C)$ that lies in every cluster box of C: such a point exists by the Helly property for boxes in $\mathbb{R}^d$. We claim that, using this Helly point, we can modify our box representation of H so that

for all thin clusters $C \in \mathcal{C}$ and for each dimension $i \in \{1, \ldots, d\}$ holds the following: if $p(C)$ and a corner t of a box of C lie in the same grid interval in dimension i, that is, if there is a j so that $p_i(C), t_i \in [j, j+1]$, then $t_i = p_i(C)$. (4)

To achieve (4), we proceed as follows. Let v be a vertex of any thin cluster $C \in \mathcal{C}$. Consider a dimension i where $\ell_i(v)$ or $r_i(v)$ lie in the same grid interval as $p_i(C)$. Note that $\ell_i(v) \leq p_i(C) \leq r_i(v)$. In dimension i, we shrink the box of v in the following way: if $\ell_i(v)$ lies in the same grid interval as $p_i(C)$, we replace $\ell_i(v)$ by $p_i(C)$. Similarly, if $r_i(v)$ lies in the same grid interval as $p_i(C)$, we replace $\ell_i(v)$ by $p_i(C)$. This procedure is illustrated in Fig. 3.

Since by shrinking a box we may only lose edges of the corresponding graph, it suffices to show that every edge is still present. Since the new box of v still contains $p(C)$, the vertex v is still adjacent to every other vertex in C. As we change the box of v only within a grid interval, the old and the new box of v are in the same position. Thus, we do not lose any edge from v to X. Performing this transformation iteratively for every box of C in every dimension, and for every thin cluster $C \in \mathcal{C}$, we obtain a box representation of H satisfying (4).

Next, we claim that

there is a pair of distinct thin clusters $C, C' \in \mathcal{C}$ such that for every $v \in C$ and $v' \in C'$ with $N_H(v) \cap X = N_H(v') \cap X$, the boxes of v and v' are in the same position. (5)

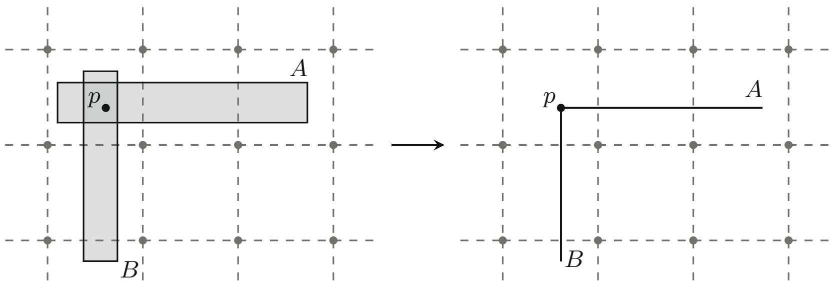

Fig. 3. Shrinking the boxes.

Note that, as C and C' are equivalent, there is indeed a bijection between the vertices of C and C' that maps a vertex v to $v' \in C'$ with $N_H(v) \cap X = N_H(v') \cap X$.

Observe that for the endpoints $\ell_i(v), r_i(v)$ of the interval representing a vertex $v \in V(H)$ in the i-th dimension, there are at most $(2k+1)^2$ many choices to select the grid intervals they lie in. Thus, any set of thin boxes, pairwise not in the same position, has size at most $(2k+1)^{2d}$. Because G is devoid of true twins, no cluster has two vertices whose boxes are in the same position.

Recall that every cluster has at most 2^k vertices. Thus, among any choice of more than $(2k+1)^{2d \cdot 2^k}$ thin clusters there are two thin clusters satisfying (5). As $(2k+1)^{2d \cdot 2^k} \leq (2k+1)^{2(2^k+k+1)) \cdot 2^k}$, by (1), and since $\mathcal{C}$ contains at least $(2k+2)^{2^{k+1}(2^k+k+1)}$ thin clusters, by (3), the claim follows.

Consider clusters C, C' as in (5). We now embed the deleted cluster C^* in the box representation of $H = G - C^*$. For this, choose $\epsilon > 0$ small enough so that

for all $v \in C$ and $w \in V(H - C)$ and all dimensions i it holds that $|s_i - t_i| > \epsilon$, when s is a corner of the box of v and t is a corner of the box of w. (6)

(If such an ϵ does not exist, we may again perturb the box representation slightly so as to guarantee (2) while keeping (4)).

Define $q \in \mathbb{R}^d$ by setting

$$q_i = \begin{cases} 1 & \text{if } p_i(C) < p_i(C') \\ -1 & \text{if } p_i(C) > p_i(C') \\ 0 & \text{if } p_i(C) = p_i(C'). \end{cases}$$

Let $v \mapsto v^*$ be the bijection between C and C^* with $N_G(v) \cap X = N_G(v^*) \cap X$. We define a box for every $v^* \in C^*$ by taking a copy of the box of v and shifting its coordinates by the vector $\epsilon \cdot q$, that is, for every dimension i we set

$$\ell_i(v^*) = \ell_i(v) + \epsilon q_i \text{ and } r_i(v^*) = r_i(v) + \epsilon q_i.$$

Note that, by choice of ϵ, the box of v^* and the box of v are in the same position.

Let $\tilde{G}$ be the graph defined by this new box representation. We claim that $\tilde{G} = G$, which then finishes the proof of the lemma.

To prove this, we first note that we only added edges between vertices in C^* and H, while all other adjacencies remain unchanged. Next, as $p(C) + \epsilon q$ is a point that lies in every box of C^*, it follows that $\tilde{G}[C^*]$ is a complete graph. Moreover, by choice of ϵ, we have

$$N_{\tilde{G}}(v^*) \setminus (C \cup C^*) = N_G(v) \setminus (C \cup C^*)$$

for any $v \in C$. In particular, $N_{\tilde{G}}(v^*) \cap C' = \emptyset$. It remains to show that also $N_{\tilde{G}}(v^*) \cap C = \emptyset$.

For this, let $w^* \in C^*$ and $v \in C$ be arbitrary, where we allow that $v = w$. Let us show that the boxes of v and w^* do not intersect.

Since v and w' are nonadjacent in H, there is a dimension i such that either $r_i(v) < \ell_i(w')$ or $r_i(w') < \ell_i(v)$. By symmetry, we may assume $r_i(v) < \ell_i(w')$. Let I be the grid interval such that $r_i(v) \in I$. If $\ell_i(w') \notin I$, then $r_i(v) < \ell_i(w^*)$, since by our construction $\ell_i(w^*)$ is in the same grid interval as $\ell_i(w')$. This means that the boxes of v and w^* do not intersect. Thus, we may assume that $\ell_i(w^*) \in I$. As v and w are in the same cluster and thus adjacent, it follows that $\ell_i(w) \leq r_i(v)$, which implies that $p_i(C) \in [\ell_i(w), r_i(v)] \subseteq I$. Now, (4) implies that $r_i(v) = p_i(C) = \ell_i(w)$.

Since $p_i(C) = r_i(v) < \ell_i(w')$, it follows that $p_i(C) < p_i(C')$. Thus, $r_i(v) = \ell_i(w) < \ell_i(w) + \epsilon = \ell_i(w^*)$. Consequently, the boxes of v and w^* do not intersect. This completes the proof. $\square$

4 An Additive 1-Approximation Algorithm

Bounded pathwidth suggests a dynamic programming approach, and this is precisely what we do. There is a hitch, though. The standard approach would be to solve the Boxicity problem on one bag after another of the path decomposition, so that the local solutions can be combined to a global one. Boxicity, however, does not permit this: as we are constructing the box representation of the graph, we may have to completely rearrange the previous boxes to add a new one.

Thus, the key issue is to force the problem to become "localized". To this end, we introduce a special interval graph I^* that reflects the path structure of the graph: two vertices are adjacent if and only if they appear in the same bag of the path decomposition. Doing so, we can safely compute local box representations of the subgraphs induced by the bags without paying attention to how these representations overlap. Indeed, the interval graph I^* gets rid of any unwanted adjacency.

Theorem 7. *There is an algorithm that, for any graph G with a given path decomposition of width w, determines in $2^{O(w^2 \log w)} \cdot |V(G)|$ time a $d \in \mathbb{N}$ so that $d \leq \mathrm{box}(G) \leq d+1$ together with a box representation of dimension $d+1$.*

Together with the algorithm of Bodlaender [7] that computes a path-decomposition of a graph G of width $\mathrm{pw}(G)$ in $f(\mathrm{pw}(G))\cdot|V(G)|$ time, we obtain Theorem 2. We note that the running time could conceivably be improved by using a faster approximation algorithm with, say, a constant approximation factor.

5 Bounded Bandwidth does not Help

It is an open problem whether boxicity is polynomial-time solvable on graphs of bounded treewidth. While we cannot solve the problem, we can offer an indication why we suspect boxicity to be hard.

The first approach to prove tractability is usually dynamic programming. Evidently, this is because Courcelle [12] proved that a vast number of problems, namely those expressible in monadic second order logic, can be solved in polynomial time by a generic dynamic programming algorithm, if the treewidth is bounded. However, nobody appears to know how to formulate "$\mathrm{box}(G) \leq d$?" in monadic second order logic, and it is doubtful that this is possible at all. More generally, dynamic programming seems to fail. Why is that so? We think this is because the tree-like structure of the input graph does not translate to a tree-like structure in the interval representation: given an input graph G of bounded treewidth, it may very well be the case that at least one interval graph in any optimal interval representation of G has unbounded treewidth.

To illustrate this, consider a $K_{2,n}$, where the smaller bipartition class is comprised of two vertices x and y, and the larger consists of $v_1, \ldots, v_n$. Clearly, $K_{2,n}$ has pathwidth 2 and boxicity 2 as well: in fact, $K_{2,n}+xy$ and $K_{2,n}+\{v_iv_j : i,j\}$ are two interval graphs whose intersection is $K_{2,n}$. Now, let I_1, I_2 be any two interval graphs with $K_{2,n} = I_1 \cap I_2$. The vertices x and y are not adjacent in at least one of I_1 and I_2, say in I_1. Suppose that I_1 contains a pair of non-adjacent v_i, v_j: then xv_iyv_jx is an induced 4-cycle, which is impossible in an interval graph. Thus, $\{v_i\}_{i=1}^n$ form a clique of size n in I_1, and I_1 has therefore pathwidth at least $n-1$.

What about stronger width-parameters? We have found a similar, albeit more complicated, example for bounded bandwidth, a parameter even more restrictive than pathwidth. Theorem 3 is a direct consequence of the following lemma.

Lemma 5. *For every n there is a graph G^n of bandwidth at most 16 and boxicity 2, so that in any interval representation $G = I_1 \cap I_2$ one of I_1 and I_2 has treewidth $\geq |V(G^n)|/32$.*

In light of the lemma, we would like to strengthen the conjecture of Adiga et al. [4]: We believe that Boxicity remains NP-complete even for graphs of bounded bandwidth.

6 Discussion

In some respect, the method of our first algorithm is a generalization of the true twin reduction. The key insight is that if there are many vertex sets (the clusters) that are identical in the graph then many of these sets will have essentially the same geometric realization. Deleting one of these many "geometric twins" is unlikely to change boxicity.

We believe this approach can exploited further. Indeed, we are convinced that with similar methods as developed in this article, we can also formulate a parameterized algorithm for Boxicity when the parameter is *distance to stars* – the smallest number of vertices whose removal results in a disjoint union of stars. Like cluster vertex deletion, distance to stars provides a non-trivial parameterization for Boxicity between vertex cover (solved) and feedback vertex set (open). Moreover, given a graph G, computing a minimum set $X \subseteq V(G)$ such that $G[V - X]$ is a disjoint union of stars can be done in $f(|X|) \cdot |V(G)|^{O(1)}$ time [9].

Our second algorithm yields an additive 1-approximation for Boxicity on graphs of bounded pathwidth. Two questions that immediately arise are: can we get rid of the additive 1, such that the algorithm computes $\mathrm{box}(G)$ exactly? Can the algorithm be lifted to run on graphs of bounded treewidth?

We turn to the second question: why is it difficult to extend the algorithm to graphs of bounded treewidth? We rely heavily on the fact that the one extra dimension is sufficient to reflect the path decomposition of the whole graph. If we mimick this approach for bounded treewidth we have to describe the tree decomposition of the graph with as few extra dimensions as possible. How many extra dimensions would we need? As many as the boxicity of the chordal supergraph obtained by turning each bag of the decomposition into a clique. If we started with a path decomposition, the boxicity will be one. For a general tree decomposition, however, it could well be that the boxicity of this chordal graph is about the treewidth of the input graph [11]. This suggests that there might be input graphs G for which $\mathrm{box}(G)$ is much lower than the number of dimensions required to describe their tree decomposition, which makes it impossible to approximate using only the techniques of Sect. 4.

References

1. Adiga, A., Babu, J., Chandran, L.S.: Polynomial time and parameterized approximation algorithms for boxicity. In: Thilikos, D.M., Woeginger, G.J. (eds.) IPEC 2012. LNCS, vol. 7535, pp. 135–146. Springer, Heidelberg (2012)
2. Adiga, A., Bhowmick, D., Chandran, L.S.: The hardness of approximating the boxicity, cubicity and threshold dimension of a graph. Discrete Appl. Math. **158**(16), 1719–1726 (2010)
3. Adiga, A., Bhowmick, D., Chandran, L.S.: Boxicity and poset dimension. SIAM J. Discrete Math. **25**(4), 1687–1698 (2011)
4. Adiga, A., Chitnis, R., Saurabh, S.: Parameterized algorithms for boxicity. In: Cheong, O., Chwa, K.-Y., Park, K. (eds.) ISAAC 2010, Part I. LNCS, vol. 6506, pp. 366–377. Springer, Heidelberg (2010)
5. Asplund, E., Grünbaum, B.: On a coloring problem. Math. Scand. **8**, 181–188 (1960)
6. Bielecki, A.: Problem 56. Colloq. Math. **1**, 333 (1948)

7. Bodlaender, H.L.: A linear-time algorithm for finding tree-decompositions of small treewidth. SIAM J. Comput. **25**(6), 1305–1317 (1996)
8. Bruhn, H., Chopin, M., Joos, F., Schaudt, O.: Structural parameterizations for boxicity. CoRR (2014). abs/1402.4992
9. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. Inf. Process. Lett. **58**(4), 171–176 (1996)
10. Chalermsook, P., Laekhanukit, B., Nanongkai, D.: Graph products revisited: tight approximation hardness of induced matching, poset dimension and more. In: Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013), pp. 1557–1576 (2013)
11. Chandran, L.S., Sivadasan, N.: Boxicity and treewidth. J. Comb. Theor. Ser. B **97**(5), 733–744 (2007)
12. Courcelle, B.: The monadic second-order logic of graphs I. Recognizable sets of finite graphs. Inf. Comput. **85**(1), 12–75 (1990)
13. Cozzens, M.: Higher and multi-dimensional analogues of interval graphs. Ph.D. thesis, Department of Mathematics, Rutgers University, New Brunswick, NJ (1981)
14. Diestel, R.: Graph Theory, 4th edn. Springer, Heidelberg (2010)
15. Doucha, M., Kratochvíl, J.: Cluster vertex deletion: a parameterization between vertex cover and clique-width. In: Rovan, B., Sassone, V., Widmayer, P. (eds.) MFCS 2012. LNCS, vol. 7464, pp. 348–359. Springer, Heidelberg (2012)
16. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Springer, London (2013)
17. Fellows, M.R., Hermelin, D., Rosamond, F.A.: Well quasi orders in subclasses of bounded treewidth graphs and their algorithmic applications. Algorithmica **64**(1), 3–18 (2012)
18. Ganian, R.: Twin-cover: beyond vertex cover in parameterized algorithmics. In: Marx, D., Rossmanith, P. (eds.) IPEC 2011. LNCS, vol. 7112, pp. 259–271. Springer, Heidelberg (2012)
19. Kostochka, A.: Coloring intersection graphs of geometric figures with a given clique number. In: Pach, J. (ed.) Towards a Theory of Geometric Graphs of Contemp. Math., vol. 342, pp. 127–138. Amer. Math. Soc. (2004)
20. Kratochvíl, J.: A special planar satisfiability problem and a consequence of its NP-completeness. Discrete Appl. Math. **52**(3), 233–252 (1994)
21. Roberts, F.S.: On the boxicity and cubicity of a graph. In: Tutte, W.T. (ed.) Recent Progress in Combinatorics, pp. 301–310. Academic Press, New York (1969)
22. Scheinerman, E.: Intersection classes and multiple intersection parameters. Ph.D. thesis, Princeton University (1984)
23. Spinrad, J.: Efficient Graph Representations: Fields Institute monographs. American Mathematical Society, USA (2003)
24. Thomassen, C.: Interval representations of planar graphs. J. Comb. Theor. Ser. B **40**(1), 9–20 (1986)
25. Yannakakis, M.: The complexity of the partial order dimension problem. SIAM J. Algebraic Discrete Methods **3**(3), 351–358 (1982)