Parameterized Algorithms for Boxicity*

Abhijin Adiga[†] Rajesh Chitnis[‡] Saket Saurabh[§]

Abstract

In this paper we initiate an algorithmic study of Boxicity, a combinatorially well studied graph invariant, from the viewpoint of parameterized algorithms. The boxicity of an arbitrary graph G with the vertex set $V(G)$ and the edge set $E(G)$, denoted by $\text{box}(G)$, is the minimum number of interval graphs on the same set of vertices such that the intersection of the edge sets of the interval graphs is $E(G)$. In the Boxicity problem we are given a graph G together with a positive integer k, and asked whether the $\text{box}(G)$ is at most k. The problem is notoriously hard and it is known to be NP-complete even to determine whether the boxicity of a graph is at most two. This rules out any possibility of having an algorithm with running time $|V(G)|^{O(f(k))}$, where f is an arbitrary function depending on k alone. Hence we look for other structural parameters like "vertex cover number" and "max leaf number" and see their effect on the problem complexity.

In particular, we give an algorithm that given a vertex cover of size k finds $\text{box}(G)$ in time $2^{O(2^k k^2)}|V(G)|$. We also give a faster additive 1-approximation algorithm for finding $\text{box}(G)$ that given a graph with vertex cover of size k runs in time $2^{O(k^2 \log k)}|V(G)|$. Our other related results include a factor $\left(2 + \frac{2}{\text{box}(G)}\right)$-approximation when parameterized by the feedback vertex set number of the input graph and a FPT algorithm for computing boxicity on co-bipartite graphs when parameterized by the vertex cover number of the "associated bipartite graph". Finally we give an additive 2-approximation algorithm for Boxicity when parameterized by the max leaf number with running time $2^{O(k^3 \log k)}|V(G)|^{O(1)}$. Our results are based on structural relationships between boxicity and the corresponding parameters and could be of independent interest.

*A preliminary version of this paper was presented in ISAAC 2010

[†]Department of Computer Science and Automation, Indian Institute of Science, Bangalore–560012, India. Supported by Infosys Technologies Ltd., Bangalore, under the "Infosys Fellowship Award". Email: *abhijin@csa.iisc.ernet.in*

[‡]Department of Computer Science, University of Maryland at College Park, USA. This work was done when the author was a summer intern at Institute of Mathematical Sciences, Chennai–600113, India. Email: *rchitnis@cs.umd.edu*

[§]The Institute of Mathematical Sciences, Chennai–600113, India. Email:*saket@imsc.res.in*

1 Introduction

Let $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ be a family of sets. An intersection graph associated with $\mathcal{F}$ has $\mathcal{F}$ as the vertex set and we add an edge between S_i and S_j if and only if $i \neq j$ and $S_i \cap S_j \neq \varnothing$. Any graph can be represented as an intersection graph, but many important graph families can be described as intersection graphs of more restricted types of set families, for instance sets derived from some kind of geometric configuration, like interval graphs, circular arc graphs, grid-intersection graphs, unit-disk graphs and boxicity k-graphs, to name a few. These graph classes are not only interesting from a graph theoretic viewpoint and combinatorial perspective but are also useful in modeling many real life applications. In this paper our object of interest is boxicity k-graphs, an intersection graph obtained by a family of boxes in the k-dimensional Euclidean space.

A k-box is a Cartesian product of closed intervals $[a_1, b_1] \times [a_2, b_2] \times \cdots \times [a_k, b_k]$. A k-box representation of a graph G is a mapping of the vertices of G to k-boxes in the k-dimensional Euclidean space such that two vertices in G are adjacent if and only if their corresponding k-boxes have a non-empty intersection. The *boxicity* of a graph G, denoted $\text{box}(G)$, is the minimum integer k such that G has a k-box representation. Boxicity was introduced by Roberts [31] in 1969 and it finds applications in modeling problems in social sciences and biology.

There has been significant amount of work done recently on finding lower and upper bounds on boxicity of different graph classes. Chandran and Sivadasan [10] showed that $\text{box}(G) \leq \text{tree-width}\,(G) + 2$. Chandran et al. [8] proved that $\text{box}(G) \leq \chi(G^2)$ where, $\chi(G^2)$ is the chromatic number of G^2. In [19] Esperet proved that $\text{box}(G) \leq \Delta^2(G) + 2$, where $\Delta(G)$ is the maximum degree of G. Scheinerman [32] showed that the boxicity of outerplanar graphs is at most 2 while Thomassen [33] proved that the boxicity of planar graphs is at most 3. In [13], Cozzens and Roberts studied the boxicity of split graphs.

While there has been a lot of work on boxicity from graph theoretic view point, the problem remains hitherto unexplored in the light of algorithms and complexity; with exceptions that are few and far between. Cozzens [12] showed that computing the boxicity of a graph is NP-hard. This was later strengthened by Yannakakis [34] and finally by Kratochvíl [26] who showed that determining whether boxicity of a graph is at most two itself is NP-complete. Recently in [1, 2], Adiga et al. showed that there exists no polynomial-time algorithm to approximate the boxicity of a bipartite graph on n vertices with a factor of $O(n^{0.5-\epsilon})$ for any $\epsilon > 0$, unless $NP = ZPP$. In [9], Chandran et al. gave a randomized algorithm to construct a box representation of a graph in $\lceil(\Delta + 2)\ln n\rceil$ dimensions, that runs in $O(\Delta n^2 \ln^2 n)$ time with high probability. They also derandomized it to obtain an $O(n^4\Delta)$ time deterministic algorithm. Recently Chandran, Francis and Mathew [7] used a different derandomization technique to obtain a $O(n^2\Delta^2 \ln n)$ time deterministic algorithm. In this paper we study the Boxicity problem – here we are given a graph G together with a positive integer k and asked whether the $\text{box}(G)$ is at most k – from the parameterized complexity perspective.

Parameterized complexity is basically a two-dimensional generalization of "P vs. NP" where in addition to the overall input size n, one studies the effects on computational complexity of a secondary measurement that captures additional relevant information. This additional information can be, for example, a structural restriction on the input distribution considered, such as a bound on the treewidth of an input graph or the size of solution set. Parameterization can be deployed in many different ways; for general background on the theory see [15, 23, 30].

For decision problems with input size n, and a parameter k, the two dimensional analogue (or generalization) of P, is solvability within a time bound of $O(f(k)n^{O(1)})$, where f is a function of k alone, as contrasted with a trivial $n^{k+O(1)}$ algorithm. Problems having such an algorithm are said to be *fixed parameter tractable* (FPT), and such algorithms are practical when small parameters cover practical ranges. The book by Downey and Fellows [15] provides a good introduction to the topic of parameterized complexity. For recent developments see the books by Flum and Grohe [23] and Niedermeier [30].

In the framework of parameterized complexity, an important aspect is the *choice of parameter* for a problem. Exploring how one parameter affects the complexity of different parameterized or unparameterized versions of the problem, often leads to non-trivial combinatorics and better understanding of the problem. In general there are two kinds of parameterizations. In the first kind the parameter reflects the value of the objective function in question. The second kind, *structural parameterizations*, measure the structural properties of the input. A well developed structural parameter is the treewidth of the input graph. Other well established structural parameters include the *vertex cover number*, the size of the minimum vertex cover of graph [21, 22] and the *max leaf number*, the maximum number of the leaves possible in a spanning tree of the input graph [20]. Observe that since determining whether the boxicity of the input graph is at most 2 is NP-complete we cannot hope to have an algorithm to test whether $\mathrm{box}(G)$ is at most k running in time $|V(G)|^{O(f(k))}$, where f is an arbitrary function depending on k alone. This initiates a study of Boxicity from the structural parameterizations like treewidth, vertex cover number of the graph and the max leaf number. We parameterize the problem with vertex cover number and max leaf number of the input graph and obtain the following results:

1. an FPT algorithm for Boxicity running in time $2^{O(2^k k^2)}|V(G)|$ when parameterized by the vertex cover number;

2. an additive 1-approximation for Boxicity when parameterized by the vertex cover number running in time $2^{O(k^2 \log k)}|V(G)|$; and

3. an additive 2-approximation for Boxicity when parameterized by the max leaf number running in time $2^{O(k^3 \log k)}|V(G)|^{O(1)}$.

Our other results include a $\left(2 + \frac{2}{\mathrm{box}(G)}\right)$-approximation when parameterized by the feedback vertex set number of the input graph and an FPT algorithm for computing boxicity on co-bipartite graphs when parameterized by the vertex cover number of the "associated bipartite graph".

Fixed-Parameter and Approximation Algorithms: Most of the approximation algorithms in the literature run in polynomial time. However by combining parameterized complexity and approximation algorithms we can possibly tackle some problems which are intractable to both the areas. We refer to the survey of Marx [27] on connections between fixed-parameter algorithms and approximation algorithms. There are some known negative results [17, 28, 18] in the area of fixed-parameter approximation which say that some problems are not approximable in FPT time. Our positive results on fixed-parameter approximation for BOXICITY are among the very few known results [29, 14, 24] of such kind. These results contribute positively to the developing area of parameterized approximation and we refer to [5, 11, 16] for further details on parameterized approximation.

All our results are based on structural relationships between boxicity and the corresponding parameter and they could be of independent interest. It is natural to ask why we do not consider parameterizing with the treewidth of the input graph. The reason for this is that, though we are not able to show it, we believe that BOXICITY is NP-hard even on graphs of constant treewidth. We leave this as an open problem.

2 Preliminaries

In this section we first give the known equivalent representation of boxicity k-graphs in terms of interval graphs. Then we show how to enumerate these graphs as they are useful for our algorithm. Finally we set up notations used throughout the paper.

2.1 Interval graphs and box representations

2.1.1 Equivalent characterization:

It is easy to see that a graph has boxicity at most 1 if and only if it is an *interval graph*, that is, each vertex of the graph can be associated with a closed interval on the real line such that two intervals intersect if and only if the corresponding vertices are adjacent. By definition, boxicity of a complete graph is 0. Let G be any graph and G_i, $1 \leq i \leq k$ be graphs on the same vertex set as G such that $E(G) = E(G_1) \cap E(G_2) \cap \cdots \cap E(G_k)$. Then we say that G is the *intersection* of G_i s for $1 \leq i \leq k$ and denote it as $G = \bigcap_{i=1}^{k} G_i$. Boxicity can be stated in terms of intersection of interval graphs as follows:

Lemma 1. Roberts [31]: *The boxicity of a non-complete graph G is the minimum positive integer b such that G can be represented as the intersection of b*

interval graphs.

Definition 1. Box representation: *We say that* $\mathcal{B} = \{I_1, I_2, \ldots, I_b\}$ *is a* b*-box representation of graph* G *if* $G = \bigcap_{i=1}^{b} I_i$*, where* $I_1, I_2, \ldots I_b$ *are interval graphs with fixed interval representations, i.e.,* $\mathcal{B}$ *can be considered as a collection of* b *interval representations.*

2.1.2 Distinguishing interval representation of an interval graph:

Let I be an interval graph. Let f_I be an *interval representation* for I, that is, it is a mapping from the vertex set to closed intervals on the real line such that for any two vertices u and v, $\{u, v\} \in E(I)$ if and only if $f_I(u) \cap f_I(v) \neq \varnothing$. Let $l(u, f_I)$ and $r(u, f_I)$ denote the left and right end points of the interval corresponding to the vertex u respectively. In some sections we will never consider more than one interval representation for an interval graph, in which case we will simplify the notations to $l(u, I)$ and $r(u, I)$. Further, when there is no ambiguity about the graph under consideration and its interval representation, we simply denote the left and right end points as $l(u)$ and $r(u)$ respectively. For any interval graph there exists an interval representation with all end points distinct. Such a representation is called a *distinguishing* interval representation. It is an easy exercise to derive such a distinguishing interval representation starting from an arbitrary interval representation of the graph.

2.2 Enumeration of b-box representations

Consider an interval graph I on n vertices. We can assume that the distinguishing interval representation f_I is a bijective map from the end points of the vertices to $[2n]$, that is, every end point is mapped to a distinct integer between 1 and $2n$. Therefore, every distinguishing interval representation can be considered as a permutation of the $2n$ end points. This immediately leads to a bound of $2n! = 2^{O(n \log n)}$ on the number of interval graphs possible on n vertices.

Let G be a graph on n vertices with boxicity at most b. A brute force enumeration of all distinct b-box representations of G will require time $\binom{2^{O(n \log n)}}{b} bn^2 = 2^{O(bn \log n)} bn^2$. The term bn^2 is required to check the validity of the box representation. This results in the following proposition.

Proposition 1 ([4])**.** *There are at most* $2^{O(nb \log n)}$ *distinct* b*-box representations of a graph* G *on* n *vertices and all these can be enumerated in time* $2^{O(bn \log n)} bn^2$.

2.3 Vertex cover and boxicity

Let G be a graph. A set $U \subset V(G)$ is a vertex cover of G if every edge of G is incident with a vertex in U. We denote by $\mathrm{MVC}(G)$ the minimum cardinality of a vertex cover in G. It is a well-known fact that $V \setminus U$ is an independent set and if $|U| = \mathrm{MVC}(G)$, then $V \setminus U$ is a maximum independent set. In [6]

Chandran, Das and Shah proved that

$$\mathrm{box}(G) \leq \left\lfloor \frac{\mathrm{MVC}\,(G)}{2} \right\rfloor + 1. \qquad (1)$$

2.4 Some definitions and notations

Let $[p]$ denote $\{1, 2, \ldots, p\}$ where p is a positive integer. For any graph G, let $V(G)$ and $E(G)$ denote its vertex set and edge set respectively. For $U \subseteq V$, let $G[U]$ be the subgraph of G induced by U. We use n to denote the number of vertices in the input graph. By $N(u)$ we denote (open) neighborhood of u that is the set of all vertices adjacent to u and by $N[u]$, the set $N(u) \cup \{u\}$. Similarly, for a subset $D \subseteq V$, we define $N[D] = \bigcup_{v \in D} N[v]$ and $N(D) = N[D] \setminus D$. The vertex cover number of a graph is the cardinality of the minimum sized subset of vertices of the graph that contains at least one end-point of every edge of the input graph. The max leaf number is the maximum number of the leaves possible in a spanning tree of the input graph. If $B_G = \{I_1, I_2, \ldots, I_k\}$ is a k-box representation, then we say that B_G has *width* k. A box representation of G of width $\mathrm{box}(G)$ is called as an *optimal box-representation.*

Property 1. Helly property of intervals: *Suppose $A_1, A_2, \ldots, A_k$ is a finite set of intervals on the real line with pairwise non-empty intersection. Then, there exists a common point of intersection for all the intervals, that is, $\bigcap_{i=1}^{k} A_i \neq \varnothing$.*

3 Boxicity parameterized by vertex cover

In this section we show that Boxicity parameterized by the vertex cover number is fixed parameter tractable. This implies that given a graph G on n vertices we can find $\mathrm{box}(G)$ by finding the minimum $1 \leq k \leq n$ for which there exists a k-box representation for G.

We start with a few definitions. Let G be a graph with the vertex set $V(G)$. Let $U \subseteq V(G)$ be a vertex cover of G size k. A k-vertex cover can be computed in time $O(2^k + kn)$ (See [30] for references). Let $S = V(G) \setminus U$ be the independent set. We partition the vertices in the independent set S based on their neighborhoods in U. For every $A \subseteq U$, let $S_A = \{u \in S \mid N(u) = A\}$. Observe that in this way we can partition the vertices of S into at most 2^k parts – one for every subset of U.

Pruning: For $A \subseteq U$, if $|S_A| \geq 1$ then, we retain an arbitrary vertex in S_A, say $v(A)$, which we call the *representative vertex* and remove other vertices of S_A from G. We call this step as *pruning* of the parts and denote the resulting graph obtained from this process by G' and let $S' = V(G') \setminus U$. The pruning step requires time $O(2^k n)$. Notice that every vertex in S' has a distinct neighborhood of U in G'. Throughout this section, G and G' represent the graphs defined above. Now we have the following definition:

Definition 2. *Let $B_{G'} = \{I'_1, I'_2, \ldots, I'_\ell\}$ be a box-representation of G' of width ℓ. We say that we can extend $B_{G'}$ to a representation, say B_G, of G of the same width if we can add intervals for vertices in $V(G) \setminus U$ while keeping the intervals of vertices from U as they are in each of the interval graphs $I'_1, I'_2, \ldots, I'_\ell$, and get an ℓ-box representation for G.*

Remark 1. *Observe that according to Definition 2, while extending the given box-representation for G', the only intervals that we are not allowed to change are that of the vertex set U. The intervals corresponding to $S' = V(G') \setminus U$ are allowed to be replaced/modified.*

Now we characterize a relationship between box-representations of G' and G which is used crucially in the correctness of our algorithms later.

Lemma 2. *Let $B_{G'} = \{I'_1, I'_2, \ldots, I'_\ell\}$ be a box-representation of G'. Then $B_{G'}$ can be extended to a representation of G, say $B_G = \{I''_1, I''_2, \ldots, I''_\ell\}$ if and only if $\forall A \subseteq U$ such that $|S_A| > 1$ there exists a $j \in [\ell]$ such that A forms a clique in I'_j.*

Proof. First we prove the forward direction. Suppose that $B_{G'} = \{I'_1, I'_2, \ldots, I'_\ell\}$ can be extended to $B_G = \{I''_1, I''_2, \ldots, I''_\ell\}$. Recall that the intervals corresponding to the vertices from U remain unchanged by the definition. Let $A \subseteq U$ such that $|S_A| > 1$ and let a_1, a_2 be two arbitrary vertices in S_A. Since $\{a_1, a_2\} \notin E(G)$ there exists $j \in [\ell]$ such that intervals of a_1 and a_2 do not intersect in I''_j. Without loss of generality, let $r(a_1, I''_j) < l(a_2, I''_j)$. Then, for every vertex in A, its interval in I''_j contains the interval $[r(a_1, I''_j), l(a_2, I''_j)]$ as they need to intersect intervals corresponding to both a_1 and a_2. Therefore, A forms a clique in I''_j. Hence A forms clique in I'_j also as the intervals of vertices from U remain unchanged while extending $B_{G'}$ to B_G.

Next we show the reverse direction of the lemma. Without loss of generality, we can assume that each interval graph representation in $B_{G'}$ is a distinguishing interval representation (see Section 2.1.2). Let $A \subseteq U$ such that $|S_A| > 1$. We know that A forms a clique in some I'_j such that $j \in [\ell]$. By Property 1, the intervals in I'_j of vertices of A have a common intersection. Since I'_j is a distinguishing interval representation, this common intersection is not a point but rather a non-trivial interval, say J. Let the interval corresponding to $v(A)$ in I'_j be J_v. Now we assign all vertices from S_A including $v(A)$ to distinct point intervals in the common interval $J \cap J_v$. Notice that we can do this because $J \cap J_v$ is not a point interval. This follows from the fact that all the intervals have pairwise distinct end-points. In all other interval graphs from $B_{G'}$, we assign to all members of S_A the same interval as that of the representative element $v(A)$ of S_A. We do this for every $A \subseteq U$ for which $|S_A| > 1$. From the description above it is evident that the new interval graphs we get by above procedure is an ℓ-box representation for G. This concludes the proof of the lemma. $\square$

Lemma 2 has the following important algorithmic consequence. This can be proved using the fact there are at most 2^k subsets of U such that $|S_A| > 1$.

Lemma 3. *Given a box representation for G' of width ℓ, in time $O(2^k k^2 \ell)$ we can determine whether it can be extended to a box representation for G and if so we can find an ℓ-box representation in time $O(2^k k^2 \ell n)$.*

Proof. Let $B_{G'} = \{I'_1, I'_2, \ldots, I'_\ell\}$ be a box-representation for G'. Lemma 2 provides a simple criteria to check whether $B_{G'}$ can be extended to an ℓ-box representation for G. The only thing we need to check is whether for every $A \subseteq U$ with $|S_A| > 1$ there is at least one interval graph, say I'_t, in $B_{G'}$ such that the vertices corresponding to A forms a clique in I'_t. This can be done in $O(|A|^2 \ell) = O(k^2 \ell)$ time. Note that the list of As such that $|S_A| > 1$ can be assumed to be available from the pruning step. Since there are at most 2^k such subsets of U the total time required is $O(2^k k^2 \ell)$. We can find the explicit representation in the stated time using the construction given in the second half of the proof of Lemma 2. □

Now we present a lemma which ensures an ℓ-box representation for G' which can be extended to an ℓ-box representation for G.

Lemma 4. *Let ℓ be the minimum integer such that there exists an ℓ-box representation of G' that can be extended to an ℓ-box representation of G. Then $\mathrm{box}(G) = \ell$.*

Proof. Let $\beta = \mathrm{box}(G)$. From the statement of the lemma we assume without loss of generality that $\beta \le \ell$. Now consider a β-box-representation of G and look at its induced representation on vertices of G'. Clearly this induced representation on G' can be extended to a representation of G. In fact, it can be extended to the β-box-representation of G we started with. Thus $\beta \ge \ell$ and hence $\mathrm{box}(G) = \beta = \ell$. □

Remark 2. *Observe that the proof of Lemma 4 also implies that there exists a box representation for G' which can be extended to a box representation for G of the same width.*

Observe that Lemmata 3 and 4 together with the above remark gives us an algorithm to find $\mathrm{box}(G)$. From [6] it is known that if G has vertex cover at most k, then, $\mathrm{box}(G) \le \lfloor k/2 \rfloor + 1$. Hence we can enumerate all possible box-representation of G', where $|V(G')| \le 2^k + k$, of width at most $\lfloor k/2 \rfloor + 1$, which by Proposition 1 takes time at most $2^{O(2^k k^2)}$, and check by Lemma 3 whether it can be extended to a box representation for G in time $O(2^k k^2)$. All this can be done in time $2^{O(2^k k^2)} n$. This gives the following theorem.

Theorem 1. *For graphs on n vertices with vertex cover bounded by k, the boxicity and an optimal box-representation can be computed in time $2^{O(2^k k^2)} n$.*

The running time obtained in Theorem 1 to compute the boxicity of a graph exactly is high. However if we are willing to accept an additive error of 1, that is, if we want an additive one approximation algorithm to compute boxicity of a graph parameterized by the vertex cover number then we can do much faster. We have the following result.

Theorem 2. *Let G be a graph with vertex cover number at most k, then in time $2^{O(k^2 \log k)}n$ we can find a w-box-representation of G such that* $\text{box}(G) \leq w \leq \text{box}(G)+1$.

The next subsection is devoted to the proof of this theorem.

3.1 Proof of Theorem 2

Let G' be the graph introduced in the pruning step and U be a vertex cover of G. Now we prove a lemma which given a box-representation of $G'[U]$ of width w gives us an efficient algorithm to check whether it can be extended to a $w+1$-box representation of G'. Towards this we need a notion of locally valid w-box-representation of G' and locally valid extensions of w-box representation.

Definition 3. *Let $B_{G'} = \{I_1, I_2, \ldots, I_w\}$ be a set of interval graphs on vertex set $V(G')$. We call it a* locally valid box *representation if the following conditions are satisfied: (a) $E(G') \subseteq \bigcap_{j\in[w]} E(I_j)$; (b) for every pair of non-adjacent vertices u and v in $V(G')$ such that at least one of them belongs to U then there exists an interval graph I_j in $B_{G'}$ such that the intervals associated to u and v do not intersect.*

Definition 4. *Let $B_U = \{I_1, I_2, \ldots, I_t\}$ be a box-representation of $G'[U]$ of width t. We say that we can extend B_U to a locally valid box-representation, say $B_{G'}$, of G' if we can add intervals for vertices in $V(G') \setminus U$ while keeping the intervals of vertices from U as they are in each of the interval graphs $I_1, I_2, \ldots, I_t$, and get a locally valid box representation for G' of width at most t.*

Remark 3. *Observe that in locally valid box representation of G' all non adjacencies have been taken care of except for those occurring in the independent set S'.*

Lemma 5. *Given a representation of $G'[U]$ of width w, we can check in time $O((4k)^w 4^k w)$ whether it can be extended to some locally valid box representation of G' of width w.*

Proof. Let $B_U = \{I_1, I_2, \ldots, I_w\}$ be a representation of $G'[U]$. Consider a distinguishing interval representation for each interval graph in B_U. Furthermore we will also assume that they have integer end-points (see Section 2.1.2). Let $A \subseteq U$ such that $|S_A| > 0$. Consider $v(A)$, the representative element of S_A and $I_j \in B_U$. If A does not induce a clique in I_j, then we assign the following interval for $v(A)$: $l(v(A), I_j) = \min_{x\in A} r(x, I_j)$ and $r(v(A), I_j) = \max_{x\in A} l(x, I_j)$, that is, we assign the interval whose left end-point corresponds to the leftmost right end-point of a neighbor and the right end-point corresponds to the rightmost left end-point of a neighbor. This is an optimal assignment as it happens to be the shortest possible interval that can be assigned to $v(A)$ in I_j, as the interval corresponding to $v(A)$ must intersect intervals corresponding to the vertices in A. We consider every A such that A is not a clique in I_j and perform the above operation. Let the resulting interval graph be denoted by I_j^*. We apply this procedure to every interval graph in B_U. Let the resulting set of interval graphs be denoted by $B_U^* = \{I_1^*, I_2^*, \ldots, I_w^*\}$. The following observation is evident from the procedure described above.

Observation 1. *We do not create new end-points in any $I^* \in B_U^*$.*

Now we give a procedure to assign interval for $v(A)$ when A induces a clique in $I \in B_U$.

Clique interval: We denote by $\mathcal{E}(A)$ the set of points on the real line which are contained in the intervals corresponding to all the vertices of A. By Property 1, $\mathcal{E}(A)$ is an interval on the real line. Therefore, we call $\mathcal{E}(A)$ the A-clique interval with respect to the interval representation of I or I^*. Since we consider only one interval representation for any interval graph, $\mathcal{E}(A)$ will be simply referred to as "A-clique interval of I^*". It is to be observed that a point in $\mathcal{E}(A)$ can be contained in intervals corresponding to vertices not in A.

Exclusive super-clique and $C(A, I^*)$: For every point $a \in \mathcal{E}(A)$, let A' be the set of all vertices whose intervals contain a. Clearly, A' is a clique. We call such a clique an exclusive super-clique of A in I^*. We use the term "exclusive" to indicate the fact that there exists a point in the interval representation which is contained in only the intervals corresponding to the vertices of A'. Note that A itself can be an exclusive super-clique. Let $C(A, I^*)$ be the set of all exclusive super-cliques of A in I^*.

Observation 2. *Every point in the A-clique interval belongs to some $A' \in C(A, I^*)$ and every $A' \in C(A, I^*)$ contains A.*

Recall that $v(A) \in S$ is the representative vertex of A in G': the vertex belonging to S_A which is retained during the pruning operation. Since $v(A)$ is adjacent to only the vertices of A in G', an optimal interval assignment for $v(A)$ is a point interval somewhere in the A-clique interval of I^*. Let $Q(I^*) \subseteq S'$ be the set of representative vertices whose neighborhood form a clique in I^*, that is, $Q(I^*) = \{v(A) | A \subseteq U \text{ and } A \text{ is a clique in } I^*\}$. Next we characterize the possible placement of vertices in the interval graphs when their neighborhood form a clique.

Claim 1. *Assigning a point interval for $v(A)$ among vertices in $Q(I^*)$ is equivalent to just choosing an A' from $C(A, I^*)$.*

Proof. Suppose we assign a point interval to $v(A)$. Clearly it has to be in the A-clique interval and from Observation 2, it follows that there exists an exclusive super-clique $A' \in C(A, I^*)$ such that the point belongs to an interval in $\mathcal{E}(A')$. Therefore fixing a point interval fixes an A' from $C(A, I^*)$.

Now suppose we pick an $A' \in C(A, I^*)$. Note that there can exist several points in $\mathcal{E}(A)$ corresponding to the exclusive super-clique A'. Let $X_{A'}$ be the set of all these points. Assigning any point from $X_{A'}$ as the point interval for

$v(A)$ will imply that $v(A)$ is adjacent to a vertex $u \in V(I^*)$ if and only if $u \in A'$. This means choosing any point from $X_{A'}$ will have the same effect on the adjacencies between $v(A)$ and U. Therefore, choosing A' from $C(A, I^*)$ is equivalent to fixing a point interval for $v(A)$. ■

We can bound the time complexity of the algorithm by showing the following.

Claim 2. *For every $A \subseteq U$ and $I^* \in B_U^*$ such that A forms a clique in I^* we have that $|C(A, I^*)| \leq 4k$.*

Proof. We call a clique X an *exclusive clique* in I^* if there exists a point in the interval representation of I^* which belongs exclusively to the intervals corresponding to all the vertices of X. Clearly, an exclusive super-clique of A is also an exclusive clique. We prove the claim by showing that the total number of exclusive cliques in I^* is less than $4k$. First we will show that the number of exclusive cliques is bounded by a function of the number of distinct end points in the interval representation of I^*. Let $p_1 < p_2 < \cdots < p_t$ be the distinct end points in the interval representation. Consider the open interval (p_i, p_{i+1}) for some $i \in [t-1]$. Note that all the points in this interval correspond to exactly one clique as there are no intervals starting or ending in this region. Now we consider the distinct end points p_i, $i \in [t]$. Note that a p_i may correspond to an exclusive clique different from the exclusive cliques corresponding to (p_{i-1}, p_i) and (p_i, p_{i+1}). This will happen if there exist two vertices u, v, such that $r(u, I^*) = l(v, I^*)$. Hence, the total number of exclusive cliques in I^* is at most $2t - 1$.

Now we show that $t \leq 2k$. Note that we started with a distinguishing interval representation for I which implies it had $2k$ distinct end points. The construction of I^* did not lead to additional end points (Observation 1). Therefore $t = 2k$ and hence, $|C(A, I^*)| < 4k$. ■

Now we consider all $I_j^* \in B_U^*$ such that A is a clique. Let $C_A^w = C(A, I_1^*) \times C(A, I_2^*) \times \cdots \times C(A, I_w^*)$. If A does not form a clique in some I_j then the corresponding $C(A, I_j^*)$ is set to $\varnothing$. We choose a candidate from C_A^w. We are required to verify if the resulting interval assignments for $v(A)$ will lead to a "valid" box representation. By validity we mean the following: look at the intervals corresponding to $v(A)$ in I_j^*, $j \in [w]$, and let $N_U(v(A), I_j^*)$ be its set of neighbors in I_j^* which belong to U. Then $A = \bigcap_{j \in [w]} N_U(v(A), I_j^*)$. Observe that we can check whether a given assignment is valid in $O(2^k w)$ time since $|V(G')| \leq 2^k + k$. If not, we consider another candidate from C_A^w and continue. This procedure is repeated until we obtain a valid box representation with respect to $v(A)$. If all the candidates of C_A^w are exhausted, then we discard the box representation B_U. Observe that the given representation of $G'[U]$ can be extended to some locally valid box representation of G' of width w if and only if B_U^* is valid with respect to every vertex in S'.

From Claim 2, it follows that $|C(A, I_j^*)| \leq 4k$. Therefore, for each set $A \subseteq U$, $|C_A^w| \leq (4k)^w$ and there are at most 2^k such sets. Thus the total time to check whether B_U^* is valid with respect to every vertex in S' takes $O(2^k(4k)^w 2^k w)$ time. This concludes the proof. □

Now we are ready to prove Theorem 2.

Let U be the vertex cover and G' be the graph described above. We know by [6] that $\text{box}(G) \leq \lfloor k/2 \rfloor + 1$. Now enumerate all the box representations B_U of width at most $1 \leq w \leq \lfloor k/2 \rfloor + 1$ of $G'[U]$. First check if for all $A \subseteq U$ such that $|S_A| > 1$ there is at least one interval graph I in B_U such that A forms a clique in I. We call such B_U good. Now using Lemma 5 check whether B_U can be extended to a locally valid representation of G'. Let w be the least integer such that B_U is good and can be extended to a locally valid representation of G'. Now we add the following interval graph I: for every vertex $w \in S'$ we associate a disjoint interval, say $[l(w), r(w)]$ and for every vertex $v \in U$ we assign the following interval $l(v) = \min_{w \in S'} l(w)$ and $r(v) = \max_{w \in S'} r(w)$. Clearly the introduction of I ensures that the non-adjacencies among the vertices of S' has been taken care of. This implies that $B_U \cup I$ is a $(w+1)$-box representation of G'. Since B_U is good, $B_U \cup I$ is also good and hence by Lemma 2 we have that $B_U \cup I$ can be extended to a $(w+1)$-box representation of G. The one question that remains unanswered is how do we know that we will always enumerate a box-representation of $G'[U]$ that can be extended to a locally valid box representation of G'. To see this it is enough to observe that given a box-w-representation of G, its induced representation on U is also a box-w-representation of $G'[U]$ and clearly it can be extended to a locally valid box representation of G' (in fact to a box representation of G'). The time taken for our algorithm using Proposition 1 is bounded by

$$O((k/2+1) \times 2^{O(k^2 \log k)} \times 2^k k \times (4k)^k 4^k k \times 2^k k^3 n) = 2^{O(k^2 \log k)} n.$$

This concludes the proof of the theorem.

3.2 On the Boxicity of Co-bipartite Graphs

In this section we give an algorithm to find the boxicity of co-bipartite graphs. A graph is called co-bipartite if it is the complement of a bipartite graph. In [34], Yannakakis showed that it is NP-complete to determine if the boxicity of a co-bipartite graph is ≥ 3. Recently, Adiga, Bhowmick and Chandran [1] showed that it is hard to approximate the boxicity of a bipartite graph within $\sqrt{n}$ factor, where n is the order of the graph. A similar result can be derived for co-bipartite graphs too.

Observe that a co-bipartite graph on n vertices has a minimum vertex cover of size $n-2$. Therefore, Theorem 1 or parameterization by the vertex cover number of the input graph is not interesting for the class of co-bipartite graphs. However, we show that given a co-bipartite graph G, finding $\text{box}(G)$ is fixed parameter tractable when parameterized by the vertex cover of the following bipartite graph associated with it.

Definition 5. *Let H be an XY co-bipartite graph, that is, $V(H)$ is partitioned into cliques X and Y. The associated bipartite graph of H, denoted by H^* is the graph obtained by making the sets X and Y independent sets, but keeping the set of edges between vertices of X and Y identical to that of H, that is, $\forall u \in X, v \in Y$, $\{u, v\} \in E(H^*)$ if and only if $\{u, v\} \in E(H)$.*

We also need the following notion:

Definition 6 ([1]). **Canonical interval representation of a co-bipartite interval graph:** *Let I be an XY co-bipartite interval graph. A canonical interval representation of I satisfies: $\forall u \in X$, $l(u) = l$ and $\forall u \in Y$, $r(u) = r$, where the points l and r are the leftmost and rightmost points respectively of the interval representation. That is, every interval associated with a vertex in X has the same left end-point and every interval associated with a vertex in Y has the same right end-point.*

It is easy to verify that such a representation exists for every co-bipartite interval graph and can be derived from an arbitrary interval representation of this graph (see [1] for more details). We need the following relation between $\mathrm{box}(H^*)$ and $\mathrm{box}(H)$.

Lemma 6. (Adiga et al. [1]) *Let H be an XY co-bipartite graph and H^* its associated bipartite graph. If H is a non-interval graph, then* $\mathrm{box}(H^*) \leq \mathrm{box}(H) \leq 2\mathrm{box}(H^*)$. *If H is an interval graph, then* $\mathrm{box}(H^*) \leq 2$.

Our main theorem of this section is as follows.

Theorem 3. *Let G be a co-bipartite graph on n vertices and G^* be its associated bipartite graph. If the vertex cover of G^* is bounded by k, then the* $\mathrm{box}(G)$ *and an optimal box representation can be computed in time* $2^{O(k^2 \log k)} n^2$.

Proof. Let G be an XY co-bipartite graph and let U be a vertex cover of G^* such that $|U| \leq k$. We first check if G is an interval graph. This can be accomplished in linear time [3] and if G is an interval graph then $\mathrm{box}(G) = 1$ and we can find an interval representation of G in polynomial time. Otherwise, by (1) we have $\mathrm{box}(G^*) \leq \left\lfloor \frac{k}{2} \right\rfloor + 1$ and combining this with Lemma 6 we have that $\mathrm{box}(G) \leq 2\mathrm{box}(G^*) \leq (k + 2)$. Observe that since G is a co-bipartite graph, the set of interval graphs such that their intersection is G consists of only co-bipartite interval graphs and therefore each of them can be assumed to have a canonical interval representation. We refer to [1] for more details. A box representation of a co-bipartite graph such that all the interval graphs are given with canonical representation is called *canonical box representation.* Now we have the following claim.

Claim 3. *Let $B_U = \{I_1, I_2, \ldots, I_w\}$ be a canonical w-box representation of $G[U]$. Then in time $O(n^2 w^2 k)$ we can decide whether we can extend this to a canonical representation of G. Here extension means obtaining a canonical box representation $B_G = \{I'_1, I'_2, \ldots, I'_w\}$ of G by assigning intervals to the vertices of $V(G) \setminus U$ in each of the interval graphs in B_U and not changing the intervals of the vertices in U.*

Proof. We assume that all the end points are mapped to integers. Since $B_U = \{I_1, I_2, \ldots, I_w\}$ is a canonical w-box representation of $G[U]$ we have that for all $I_j \in B_U$, the left end point of vertices in $U \cap X$ is same and the right end point of vertices in $U \cap Y$ is same. Let these be l_j and r_j respectively. We try to extend B_U to a canonical box representation of G in the following manner. Let $S = V(G) \setminus U$ and observe that since U is a vertex cover of G^* we have that $G^*[S]$ is an independent set. From the definition of G^*, it is clear that no vertex in $S \cap X$ is adjacent to any vertex in $S \cap Y$ in G. Now we define the intervals associated with vertices in S. Fix some $I_j \in B_U$. If $u \in S \cap X$, then we set $l(u, I_j) = l_j$ and $r(u, I_j) = \max_{z \in N(u) \cap Y} l(z, I_j)$. Similarly if $u \in S \cap Y$, then $r(u, I_j) = r_j$ and $l(u, I_j) = \min_{z \in N(u) \cap X} r(z, I_j)$. Observe that if $u \in S \cap X$ ($u \in S \cap Y$ respectively), then the optimal placement of its right (left) end point is essentially dependent on the left (right) end points of just the vertices in $U \cap Y$ ($U \cap X$) and not all of Y (X resp.). Given this observation, it is clear that we have assigned the shortest possible interval while adhering to the canonical structure of the interval representation.

For each $u \in S$, we need $O(k)$ time to find its interval on I_j. So we need total $O((n-k)kw) = O(nkw)$ time to build the new proposed representation for G. After assigning intervals to all the vertices in S we need to validate the box representation. This requires time $O(n^2 w)$ (see Section 2.2). Thus, the total time required to check if B_U can be extended to a box representation of G is $O(nkw + n^2 w) = O(n^2 kw)$. ■

Claim 3 gives us an algorithm to find $\mathrm{box}(G)$ and an optimal box representation of G. We enumerate all canonical box representations of width at most $k+2$ of $G[U]$. After that using Claim 3 we can test in time $O(n^3 k^3)$ whether we can extend the enumerated canonical box representation to a canonical box representation of G. We return the minimum w between 1 and $k+2$ for which the Claim 3 returns a canonical w-box representation for G. The one question that remains unanswered is how do we know that we will always enumerate a canonical box-representation of $G[U]$ of width $\mathrm{box}(G)$ that can be extended to a canonical box representation of G. To see this it is enough to observe that given a canonical w-box representation of G, its induced representation on U is a canonical w-box representation of $G[U]$. The time taken for our algorithm is bounded by

$$O((k+2)2^{O(k^2 \log k)} n^2 w^2 k) = 2^{O(k^2 \log k)} n^2.$$

This concludes the proof of the theorem. □

3.3 Boxicity Parameterized by Feedback Vertex Set Number

In this section we obtain a factor $\left(2 + \frac{2}{\mathrm{box}(G)}\right)$-approximation algorithm for finding the boxicity of a graph G running in time $f(k)|V(G)|^{O(1)}$ where k is the size of the minimum feedback vertex set of the input graph G. A feedback vertex set of a graph G is a subset $U \subseteq V(G)$ such that $G[V(G) \setminus U]$ is a forest. More precisely we have the following.

Theorem 4. *Let G be a graph with the minimum feedback vertex set size bounded by k. Then, there is a factor $\left(2 + \frac{2}{\mathrm{box}(G)}\right)$-approximation algorithm to compute boxicity of G running in time $f(k)|V(G)|^{O(1)}$, where $f(k)$ is the exponential part of the running time of the algorithm to compute boxicity of a given graph having vertex cover of size at most k.*

Proof. Let $U \subseteq V(G)$ be a feedback vertex set of G. By definition, $S = V(G) \setminus U$ induces a forest in G and therefore, $G[S]$ is a bipartite graph with partite sets say X and Y. Let $G_1 = G[U \cup X]$ and $G_2 = G[U \cup Y]$. Clearly U is a vertex cover of G_1 and G_2. Using Theorem 1, we obtain a box representation $B_1 = \{I_{11}, I_{12}, \ldots, I_{1r}\}$ for G_1 and $B_2 = \{I_{21}, I_{22}, \ldots, I_{2s}\}$ for G_2 where $\mathrm{box}(G_1) = r$ and $\mathrm{box}(G_2) = s$. It is a well-known fact that the boxicity of a forest is at most 2 and can be constructed in polynomial time. Let $B_3 = \{I_{31}, I_{32}\}$ be a box representation for $G[S]$.

For each $I_{1j} \in B_1$, we construct I'_{1j} by introducing vertices of Y as universal vertices, that is, to every vertex $v \in Y$ we assign the following interval $l(v, I_{1,j}) = \min_{w \in V(G_1)} l(w, I_{1,j})$ and $r(v, I_{1,j}) = \max_{w \in V(G_1)} r(w, I_{1,j})$. Similarly, for each $I_{2j} \in B_2$, we construct I'_{2j} by introducing vertices of X as universal vertices. Finally, for each $I_{3j} \in B_3$, we construct I'_{3j} by introducing vertices in U as universal vertices. We call the new box representations B'_1, B'_2 and B'_3 respectively. It is easy to verify that the intersection of edge sets of interval graphs in B'_1, B'_2 and B'_3 is $E(G)$. This implies that the $\mathrm{box}(G) \leq \mathrm{box}(G_1) + \mathrm{box}(G_2) + 2$. Since, $\mathrm{box}(G_1), \mathrm{box}(G_2) \leq \mathrm{box}(G)$, we have a box representation of G comprising of at most $2\mathrm{box}(G) + 2$ interval graphs. This gives us the desired approximation factor of the algorithm. □

4 Boxicity Parameterized by Max Leaf Number

In this section we obtain an additive 2-approximation algorithm for finding the boxicity of a connected graph G running in time $f(k)|V(G)|^{O(1)}$ where k is the number of the maximum possible leaves in any spanning tree of the input graph G. The number of the maximum possible leaves in any spanning tree of the input graph G is called the *max-leaf number* of G.

Suppose graph G has n vertices and max-leaf number k. Consider the following partitioning of the vertex set: (1) $V_{=1}$: vertices of degree 1; (2) $V_{=2}$: vertices of degree 2 and (3) $V_{>2}$: degree at least 3. Let $H = G[N[V_{>2} \cup V_{=1}]]$, i.e., the graph induced on the vertex set $V_{>2} \cup V_{=1}$ and its neighbor set. Here, we assume that G has at least one vertex of degree ≥ 3. Otherwise, G is a cycle or a path, whose boxicity is trivial to compute. Our algorithm is based on the following lemma.

Lemma 7. *If G and H are as defined above, then* $\mathrm{box}(H) \leq \mathrm{box}(G) \leq \mathrm{box}(H)+2$*. Furthermore given a b-box-representation of H it can be made into $(b+2)$-box-representation of G in polynomial time.*

Proof. Since H is an induced subgraph of G, $\mathrm{box}(H) \leq \mathrm{box}(G)$. Now we will give a construction that shows that $\mathrm{box}(G) \leq \mathrm{box}(H)+2$. Let $\mathrm{box}(H) = b$ and $B = \{I_1, I_2, \ldots, I_b\}$ be a box representation of H.

Definition 7. Excluded path: *An excluded path in a graph is a path in which the interior vertices have degree 2. In other words, each interior vertex is adjacent to only its two neighbors of the path in the graph.*

Claim 4. *$G[V(G) \setminus V(H)]$ is a collection of paths which happen to be excluded paths in G.*

Proof. Since $V(G) \setminus V(H) \subseteq V_{=2}$, the maximum degree of $G[V(G) \setminus V(H)]$ is 2, which in turn implies that it is a collection of cycles and paths. However, if there is a cycle, then every vertex participating in this cycle would be of degree 2 in G and hence form a separate component in G, contradicting the fact that G is connected and is not a cycle.

Since each vertex belongs to $V_{=2}$, every interior vertex of the paths is adjacent to only its two neighbors of the path in the G and therefore, each path is an excluded path in G. ■

Claim 5. *The end points of every excluded path $P \in G[V(G) \setminus V(H)]$ are adjacent to two distinct vertices of $V_{=2}$ which are neighbors of some vertices in $V_{>2} \cup V_{=1}$.*

Proof. The proof is by contradiction. If an end point of path P was not adjacent to a vertex in $V_{=2}$, then, it implies that either it is adjacent to a vertex in $V_{>2} \cup V_{=1}$ or it is not adjacent to any other vertex. In the latter case, it would imply that the end point belongs to $V_{=1}$. In either case, this would imply that the end point belongs to $N[V_{>2} \cup V_{=1}] = V(H)$, a contradiction.

If the end points of the path are adjacent to the same vertex in $V_{=2}$, then it would imply that the path and the vertex together would induce an isolated cycle in G, a contradiction to the fact that G is connected and is not a cycle. ■

Construction of the box representation $B' = \{I'_1, I'_2, \cdots, I'_{b+1}, I'_{b+2}\}$ for G: We order the paths of $G[V(G) \setminus (V_{>2} \cup V_{=1})]$. Let them be denoted as $P_1, P_2, \ldots, P_m$ where m is the total number of paths. Let $P'_i = x_i P_i y_i$, where $x_i, y_i \in V_{=2} \cap N(V_{>2} \cup V_{=1})$ (see Claim 5). We note that x_i and y_i are not adjacent $\forall i \in [m]$. This is because both x_i and y_i have degree 2 in G and have one neighbor each in the path P_i and one neighbor in $V_{>2} \cup V_{=1}$.

For each $I \in B$, we construct I' by introducing the vertices in $G \setminus H$ as follows: If x_i and y_i are not adjacent in I, then, without loss of generality we assume that $r(x_i, I_a) < l(y_i, I_a)$. Suppose $P'_i = x_i v_{i1} v_{i2} \cdots v_{ip(i)} y_i$, then we consider $p(i)+1$ distinct points $r(x, I_a) = c_0 < c_1 < c_2 < \cdots < c_{p(i)-1} < c_{p(i)} = l(y_i, I_a)$. We assign to v_{ij} the interval $[c_{j-1}, c_j]$.

If x_i and y_i are adjacent in I, then we assign the same point interval within the region of intersection of the intervals of x_i and y_i to all vertices v_{ij}. We also construct two extra graphs I'_{b+1} and I'_{b+2} as follows:

$\mathbf{I'_{b+1}}$: All vertices in $V_{>2} \cup V_{=1}$ are assigned to the interval $[m+1, m+2]$. Consider the path P'_i. The vertices x_i and y_i are assigned $[i, m+2]$ and other vertices in P'_i are assigned the point interval $[i+\frac{1}{2}, i+\frac{1}{2}]$.

$\mathbf{I'_{b+2}}$: All vertices in $V_{>2} \cup V_{=1}$ are assigned to the interval $[-1, 0]$. For a path P_i, the vertices x_i and y_i are assigned $[-1, i]$ and all other vertices in P'_i are assigned the point interval $[i-\frac{1}{2}, i-\frac{1}{2}]$.

Proof of $G = \bigcap_{a=1}^{b+2} I'_a$: Note that $\forall j \in [b]$, the interval of every vertex from $V(H)$ in I'_j is same as the interval in I_j and $V(H)$ forms a clique in both I'_{b+1} and I'_{b+2}. Thus $\bigcap_{j=1}^{b+2} I'_j$ restricted to $V(H)$ is exactly H.

Observe that the way we have constructed I'_j for $j \in [b]$, we have introduced intervals for every vertex in P'_i in I'_j for $i \in [m]$. Since x_i and y_i are not adjacent in H there exists $j \in [b]$ such that intervals of x_i and y_i do not intersect in I_j. Again, without loss of generality let $r(x_i, I_j) < l(y_i, I_j)$. Then by our procedure we would have packed intervals of the interior vertices of P'_i in between $r(x_i, I_j)$ and $l(y_i, I_j)$ in the interval graph I'_j. Thus, in I'_j, $G[V(P'_i)] = P'_i$. On the other hand if x_i and y_i intersect in some I_l then $V(P'_i)$ forms a clique in I'_l. Also $V(P'_i)$ forms a clique in I'_{b+1} and I'_{b+2}. Therefore, $\forall i \in [m]$, $\bigcap_{a=1}^{b+2} I'_a$ restricted to $V(P'_i)$ is exactly P'_i.

We now need to only show that vertices of an excluded path P_i are separated from other excluded paths and vertices in $V(H) \setminus \{x_i, y_i\}$. Clearly vertices of P_i are separated from the interior vertices of other excluded paths P_j and $V_{>2} \cup V_{=1}$ in both I'_{b+1} and I'_{b+2}. Suppose $v \in P_i$ and x_j (or y_j) is an end vertex of P'_j, $j \neq i$. If $i < j$, then v and x_j (or y_j) are separated in I'_{b+1}, else, they are separated in I'_{b+2}. By the same argument it follows that x_i (and y_i) are separated from the vertices of P_j in either I'_{b+1} or I'_{b+2}.

Hence, we have proved that $G = \bigcap_{a=1}^{b+2} I'_a$. Thus $\{I'_1, \ldots, I'_{b+1}, I'_{b+2}\}$ is a $b+2$-box-representation of G. $\square$

Our result in this section depends on Lemma 7 and the following known structural result.

Lemma 8. Kleitman and West [25] *If the max leaf number of a graph is equal to k, then, it is a subdivision of a graph on at most $4k-2$ vertices.*

From this we obtain the following theorem.

Theorem 5. *Let G be a connected graph on n vertices with max leaf number bounded by k. Then we can obtain an additive 2-approximation algorithm to compute the boxicity of the graph G running in time $2^{O(k^3 \log k)} n^{O(1)}$.*

Proof. Lemma 8 says that we can obtain G as a subdivision of a graph on at most $4k-2$ vertices. Since a subdivision only introduces degree 2 vertices, it follows that there are at most $4k-2$ vertices of degree ≥ 3 and equal to 1 in G. Therefore $|V_{>2} \cup V_{=1}| \leq 4k-2$. Furthermore since the max-leaf number of the graph is at most k, the maximum degree of the graph G is at most k, for otherwise, we can start with a vertex of degree at least $k+1$ and grow it to a spanning tree with more than k leaves. This implies that $|V(H)| \leq 4(k-2) + k(4k-2)$ where H is as defined at the beginning of this section.

Since there are $O(k^2)$ vertices in H, by Proposition 1 enumeration of all box-b-representation of H takes $2^{O(bk^2 \log k)}$ time. Let $tw(G)$ denote the treewidth of the graph G. The $box(H) \leq tw(H) + 2 \leq tw(G) + 2 \leq 2ml(G) + 2 = 2k + 2$ [10, 20]. Hence by Proposition 1 the time taken for enumerating all box-representations of H of width at most $2k+2$ is upper bounded by $2^{O(k^3 \log k)}$. By Lemma 7, in polynomial time the given representation can be extended to G. This gives the desired running time of the algorithm and concludes the proof. □

5 Conclusion

In this paper we initiated a systematic study of computing the boxicity of a graph in the realm of parameterized complexity. The problem is notoriously hard and it is known to be NP-complete even to determine whether the boxicity of a graph is at most two. Hence we studied this problem by parameterizing with parameters that are FPT like the vertex cover number and the max-leaf number of the input graph. We showed that finding boxicity of a graph when parameterized by the vertex cover number is FPT, obtained a faster additive 1-approximation algorithm when parameterized by the vertex cover number and finally obtained an additive 2-approximation algorithm to boxicity of the graph when parameterized by the max leaf number of the graph. Our other results included factor $\left(2 + \frac{2}{\mathrm{box}(G)}\right)$-approximation when parameterized by the feedback vertex set number of the input graph and a FPT algorithm for computing boxicity on co-bipartite graphs when parameterized by the vertex cover number of the associated bipartite graph. Our results were based on structural relationships between boxicity and the corresponding parameters and could be of independent interest. We have not only obtained several algorithms for computing boxicity but also have opened up a plethora of interesting open problems. The main ones include. (1) Is Boxicity FPT when parameterized by the feedback vertex set or the max-leaf number? (2) Is it NP-hard to compute the boxicity of graphs of constant treewidth?

References

[1] A. Adiga, D. Bhowmick, L. S. Chandran, Boxicity and poset dimension., in: M. T. Thai, S. Sahni (eds.), COCOON, vol. 6196 of Lecture Notes in

Computer Science, Springer, 2010.

[2] A. Adiga, D. Bhowmick, L. S. Chandran, The hardness of approximating the threshold dimension, boxicity and cubicity of a graph, DAM 158 (16) (2010) 1719–1726.

[3] K. S. Booth, G. S. Lueker, Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms, J. Comput. Syst. Sci. 13 (3) (1976) 335–379.

[4] A. Brandstädt, V. B. Le, J. P. Spinrad, Graph classes: a survey, SIAM Monographs on Discrete Mathematics and Applications, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1999.

[5] L. Cai, X. Huang, Fixed-parameter approximation: Conceptual framework and approximability results, in: IWPEC, vol. 4169 of Lecture Notes in Computer Science, 2006.

[6] L. S. Chandran, A. Das, C. D. Shah, Cubicity, boxicity, and vertex cover, Disc. Math. 309 (2009) 2488–2496.

[7] L. S. Chandran, M. C. Francis, R. Mathew, Finding a box representation for a graph in $o(n^2\Delta^2 \ln n)$ time, in: ICIT, Bhubaneswar, 2008.

[8] L. S. Chandran, M. C. Francis, N. Sivadasan, Boxicity and maximum degree, J. Comb. Theory Ser. B 98 (2) (2008) 443–445.

[9] L. S. Chandran, M. C. Francis, N. Sivadasan, Geometric representation of graphs in low dimension using axis parallel boxes, Algorithmica 56 (2) (2010) 129–140.

[10] L. S. Chandran, N. Sivadasan, Boxicity and treewidth, J. Comb. Theory Ser. B 97 (5) (2007) 733–744.

[11] Y. Chen, M. Grohe, M. Grüber, On parameterized approximability, in: IWPEC, vol. 4169 of Lecture Notes in Computer Science, 2006.

[12] M. B. Cozzens, Higher and multi-dimensional analogues of interval graphs, Ph.D. thesis, Department of Mathematics, Rutgers University, New Brunswick, NJ (1981).

[13] M. B. Cozzens, F. S. Roberts, Computing the boxicity of a graph by covering its complement by cointerval graphs, Disc. Appl. Math. 6 (1983) 217–228.

[14] E. D. Demaine, M. T. Hajiaghayi, K. ichi Kawarabayashi, Algorithmic graph minor theory: Decomposition, approximation, and coloring, in: FOCS, 2005.

[15] R. G. Downey, M. R. Fellows, Parameterized complexity, Springer-Verlag, New York, 1999.

[16] R. G. Downey, M. R. Fellows, C. McCartin, Parameterized approximation problems, in: IWPEC, vol. 4169 of Lecture Notes in Computer Science, 2006.

[17] R. G. Downey, M. R. Fellows, C. McCartin, F. A. Rosamond, Parameterized approximation of dominating set problems, Inf. Process. Lett. 109 (1) (2008) 68–70.

[18] K. Eickmeyer, M. Grohe, M. Grüber, Approximation of Natural W[P]-Complete Minimisation Problems Is Hard, in: IEEE Conference on Computational Complexity, 2008.

[19] L. Esperet, Boxicity of graphs with bounded degree, European J. Combin. 30 (5) (2009) 1277–1280.

[20] M. R. Fellows, D. Lokshtanov, N. Misra, M. Mnich, F. A. Rosamond, S. Saurabh, The complexity ecology of parameters: An illustration using bounded max leaf number, Theory Comput. Syst. 45 (4) (2009) 822–848.

[21] M. R. Fellows, D. Lokshtanov, N. Misra, F. A. Rosamond, S. Saurabh, Graph layout problems parameterized by vertex cover, in: ISAAC, vol. 5369 of Lecture Notes in Computer Science, 2008.

[22] J. Fiala, P. A. Golovach, J. Kratochvíl, Parameterized complexity of coloring problems: Treewidth versus vertex cover, in: TAMC, vol. 5532 of Lecture Notes in Computer Science, 2009.

[23] J. Flum, M. Grohe, Parameterized Complexity Theory, Texts in Theoretical Computer Science. An EATCS Series, Springer-Verlag, Berlin, 2006.

[24] M. Grohe, M. Grüber, Parameterized Approximability of the Disjoint Cycle Problem, in: ICALP, 2007.

[25] D. J. Kleitman, D. B. West, Spanning trees with many leaves, SJDM 4 (1991) 99–106.

[26] J. Kratochvíl, A special planar satisfiability problem and a consequence of its NP-completeness, Disc. Appl. Math. 52 (1994) 233–252.

[27] D. Marx, Parameterized complexity and approximation algorithms, Comput. J. 51 (1) (2008) 60–78.

[28] D. Marx, Completely inapproximable monotone and antimonotone parameterized problems, in: IEEE Conference on Computational Complexity, 2010.

[29] D. Marx, I. Razgon, Constant ratio fixed-parameter approximation of the edge multicut problem, in: ESA, vol. 5757 of Lecture Notes in Computer Science, 2009.

[30] R. Niedermeier, Invitation to fixed-parameter algorithms, vol. 31 of Oxford Lecture Series in Mathematics and its Applications, Oxford University Press, Oxford, 2006.

[31] F. S. Roberts, Recent Progresses in Combinatorics, chap. On the boxicity and cubicity of a graph, Academic Press, New York, 1969, pp. 301–310.

[32] E. R. Scheinerman, Intersection classes and multiple intersection parameters, Ph.D. thesis, Princeton University (1984).

[33] C. Thomassen, Interval representations of planar graphs, J. Comb. Theory Ser. B 40 (1986) 9–20.

[34] M. Yannakakis, The complexity of the partial order dimension problem, SIAM J. Alg. Disc. Math. 3 (3) (1982) 351–358.