Exploring Boxicity: Graph Representations in Multidimensional Spaces

Amna Adnan[1], Matthew Barclay[2], Marco Caoduro[3], Joshua Childs[2], Will Evans[3], and Tao Gaede[2]

1: University of Calgary, 2: University of Victoria, 3: The University of British Columbia

Pacific Institute *for the* Mathematical Sciences

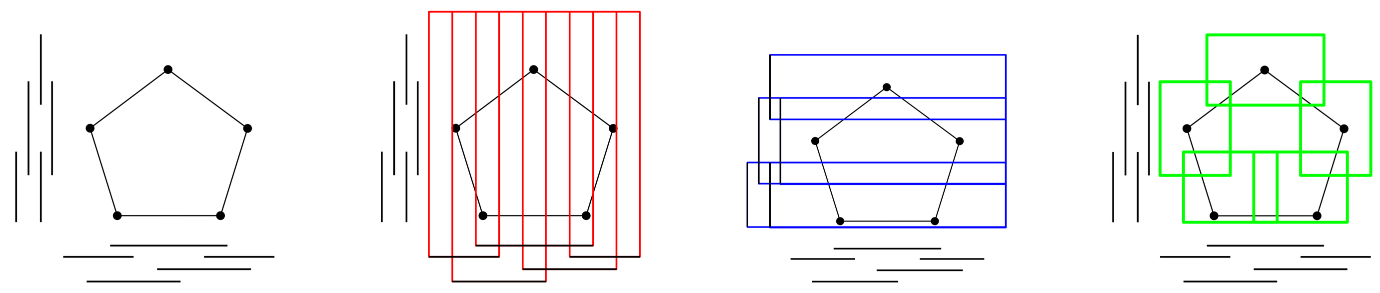
1. Introduction

For a graph $G = (V, E)$, the *boxicity* of G, denoted by box(G), is the minimum dimension d such that G is the intersection graph of a family $(B_v)_{v\in V}$ of d-dimensional boxes. An axis-parallel box in $\mathbb{R}^d$ is a Cartesian product $I_1 \times I_2 \times \cdots \times I_d$, where I_i is an interval of the real line. Boxicity was first came about through F.S. Roberts and J.E. Cohen work on combinatorial applications to biological sciences [5] [4]. It is a useful descriptor of interdependence for graphs representing intersecting elements in a multidimensional system.

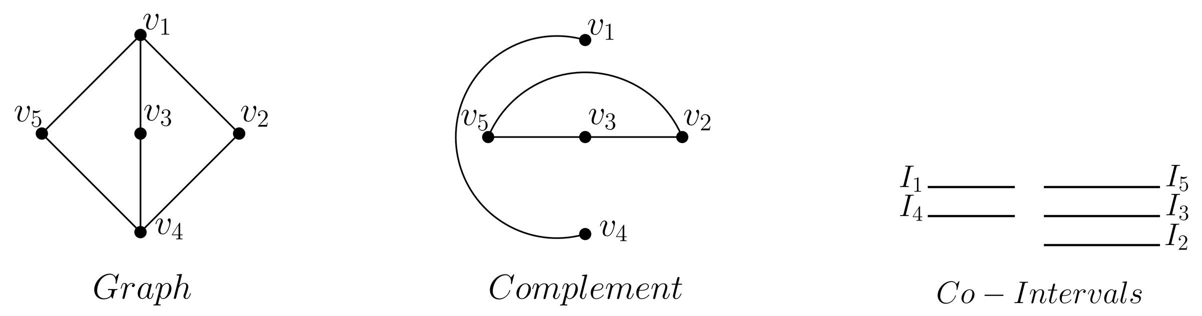
2. Background and Methods

Boxicity is the k-dimensional extension of interval graphs. A box-intersection representation of a graph G may be constructed by the intersection of interval graphs. This is described in the following theorem.

Theorem 1 *A graph G has boxicity at most k if and only if it can be represented as the intersection graph of k interval graphs.*



Depending on the graph, taking these intersections can be difficult - they are reductive, not constructive, requiring a reasonable estimate of a graph's boxicity. To avoid these challenges, we may use DeMorgan's laws to obtain a more constructive approach wherein we cover the graph complement using complements of interval graphs, which we call *co-interval graphs.* For any ordering σ of $V(G)$, G^σ is a co-interval subgraph of G [2], constructed by iteratively intersecting the neighbourhoods of the vertices in order. Co-interval subgraphs will be hereafter referred to as interval-order subgraphs.



Definition 1 (Interval-Order Cover) *Let G be a graph, and let $\mathcal{F}$ be a family of k interval-order subgraphs of G. Then $\mathcal{F}$ is a* k-interval-order cover *of G if $\cup_{i=1}^{k} V(H_i) = V(G)$ and $\cup_{i=1}^{k} E(H_i) = E(G)$, for $H_i \in \mathcal{F}$.*

The following lemma is used extensively to determine boxicity upper bounds.

Lemma 2 (Cozzens and Roberts [3], 1983) *Let $G = (V, E)$ be a graph, and let $\overline{G}$ be its complement. Then* box$(G) \leq k$ *if and only if $\overline{G}$ has a k-interval-order cover.*

Observe from Lemma 2 that to find the boxicity of a graph G, it is sufficient to find the least number of interval-order subgraphs required to cover $E(G)$. Following the techniques of Caoduro and Sebő [2], every graph G has a class of *maximal* interval-order subgraphs that may be used to minimally cover $\overline{G}$. We define such maximal graphs as follows:

Definition 2 (Maximal Interval-Order Subgraphs (MIOS)) *Let $G = (V, E)$ be a graph, and let σ be an ordering of $V(G)$. We say that G^σ is* inclusion-wise maximal *if and only if there does not exist an ordering τ of $V(G)$ such that G^σ is a proper subgraph of G^τ.*

Following the example of previous literature (see for e.g. [1], [2]), we seek ways of characterizing the maximal interval-order subgraphs of select graph classes.
If we can characterize (or better, enumerate) the maximal interval-order subgraphs of a graph, then it becomes much easier to find a minimal covering of the graph's complement. Without such characterizations, we would have to try every possible interval-order subgraph to find a potentially minimal covering.

4. Other Results

1. A similar argument to the algorithm for the complements of trees was proved for the boxicity of the complement of line graphs of trees.
2. We determined that, for the Kneser graph $K(n, k)$, finding the maximal subgraphs in the $k = 3$ case was much more complex than the $k = 2$ case, requiring new methods.
3. Investigation of the boxicity of the Mycielskian of cycles and complete graphs.

5. Future Directions

Given that trees are outer planar graphs, we can probe for a more general result: perhaps there is a polynomial-time algorithm for finding the boxicity of complements of outerplanar graphs, which is similar to our algorithm for the complements of trees?

6. References

[1] M. Caoduro and L. Lichev. On the boxicity of kneser graphs and complements of line graphs. *Discrete Mathematics*, 346(5):113333, 2023.

[2] M. Caoduro and A. Sebő. Boxicity and interval-orders: Petersen and the complements of line graphs, 2023.

[3] M. B. Cozzens and F. S. Roberts. Computing the boxicity of a graph by covering its complement by cointerval graphs. *Discrete Applied Mathematics*, 6(3):217–228, 1983.

[4] F. S. Roberts. Discrete mathematical models, with applications to social, biological, and environmental problems. 1976.

[5] F. S. Roberts. Food webs, competition graphs, and the boxicity of ecological phase space. In Y. Alavi and D. R. Lick, editors, *Theory and Applications of Graphs*, pages 477–490, Berlin, Heidelberg, 1978. Springer Berlin Heidelberg.

3. Main Result

One of our main results was a polynomial-time algorithm that produces a family of ants that can be used to compute the boxicity of the complement of a tree T. By this, we mean that by finding a minimum ant-covering of trees in polynomial time, we were able to obtain an algorithm to compute the boxicity of the complement of a tree.

To prove this, we characterized the maximal interval-order subgraphs of a trees, and then discovered a polynomial-time algorithm for finding a minimal interval-order cover. The maximal interval-order subgraphs for trees are a construction we define as *ants.*

Definition 3 (uv-ant) *Let $G = (V, E)$ be a graph and $uv \in E$. A uv-ant with respect to G is the graph A having vertex-set $V(A) = N(u) \cup N(v)$ and the edge set $E(A) = \delta(u) \cup \delta(v)$. We call the edge uv the* body *of the ant and the edges $E(A) \setminus \{uv\}$ the* legs *of the ant.*

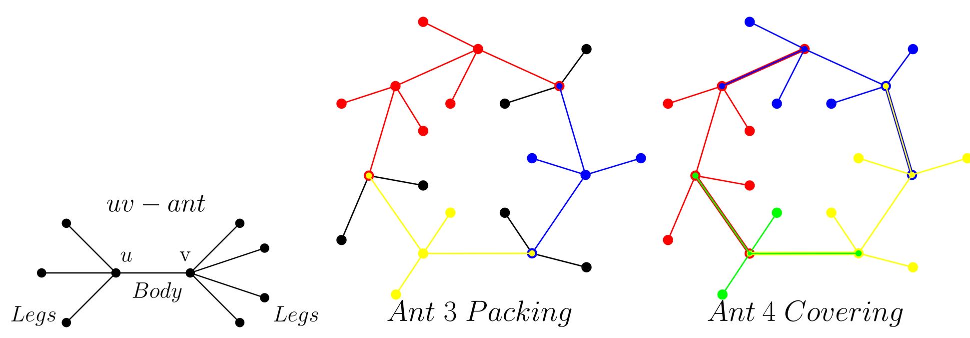
Lemma 3 (Cozzens and Roberts [3], 1983) *Let G be a graph, and let $uv \in E(G)$. Then the uv-ant is an interval-order subgraph..*

Lemma 4 *Let G be a subgraph of T. If G is a maximal interval-order subgraph of T, then G is a uv-ant for some $uv \in E(T)$.*

Now that we can characterize the maximal interval-order subgraphs for trees as ants, then the problem of bounding the boxicity of a graph is reduced to finding a minimal covering with maximal interval-order subgraphs, ants, of the complement of T, $\overline{T}$.

Finding a Minimal Covering with Maximal Interval-Order Subgraphs, Ants, of the Complement of T, $\overline{T}$: Generally for a graph, G, the covering, $\nu(G)$, is at least the packing, $\tau(G)$, i.e. $\tau(G) \leq \nu(G)$. The covering, $\nu(G)$, and packing, $\tau(G)$, are not necessarily equal, but in trees they are. We define the ant-packing number and the ant-covering number of a graph G by the following:

Definition (Ant-Packing Number): Let G be a graph. The ant-packing number of G, denoted by $\alpha_A(G)$, is the maximum number of pairwise edge-disjoint ants in G.
Definition (Ant-Covering Number): Let G be a graph. The ant-covering number of G, denoted by $\tau_A(G)$, is the minimum number of ants needed to cover the edges of G.



Relationship between the boxicity of the complement of T, box$(\overline{T})$, and the ant-packing number, $\alpha_A(G)$, and the ant-covering number, $\tau_A(G)$: For a tree T, the boxicity of the complement of T, *box*$(\overline{T})$, is upper bounded by the ant-covering number, $\tau_A(T)$, and lower bounded by the ant-packing number, $\alpha_A(T)$, i.e. $\alpha_A(T) \leq box(\overline{T}) \leq \tau_A(T)$.

We will see that the aforementioned algorithm proves that for a tree, the ant-packing number and the ant-covering number are equal: **Theorem:** Let T be a forest, then $\alpha_A(T) = \tau_A(T)$. Now, *because for a tree T*, $\alpha_A(T) \leq box(\overline{T}) \leq \tau_A(T)$ and the algorithm proves that $\alpha_A(T) = \tau_A(T)$, we have $box(\overline{T}) = \alpha_A(T) = \tau_A(T)$. Therefore as a Corollary of the aforementioned Theorem, the algorithm allows us to compute the boxicity of the complement of a tree, $box(\overline{T})$, in polynomial-time.

Algorithm: If the graph G is a forest, then this algorithm is performed for each connected component in parallel. Note: The algorithm makes mention of an "almost-leaf", which is defined as a vertex that becomes a leaf after all leaves of the original tree T are removed.

1. Choose an arbitrary vertex, r, and root the tree, T, at r. For a connected component, if the height, h, is at most 2, then for each depth 1 vertex, there will be an ant for the covering and packing, therefore we can see that $\alpha_A(T) = \tau_A(T)$.
2. For the tree T, rooted at r, select all almost leaves furthest away from the root vertex.
3. Label each almost leaf, $u_{h,i}$, where h is the height of the tree, and i is the almost-leaf count for the tree T.
4. Choose a non-leaf neighbour of $u_{h,i}$ and label it $v_{h,i}$. Subsequently, choose a neighbour of $u_{h,i}$ which is a leaf and call it $l_{h,i}$.
5. Use u and v to construct an u, v-ant.
6. We define a class C, $C := C \cup \{u_i, v_i\}$, where the bodies of the ants used for *covering* are stored; and a class P, $P := P \cup \{u_i, l_i\}$, where the bodies of the ants used for *packing* are stored. For each almost-leaf, we get one packing ant and one covering ant.
7. Now define T_{h-1} as T_h minus the edges of the $u_{h,i} - v_{h,i}$ ant.
8. Repeat this process.