Exploring Boxicity: Graph Representations in Multidimensional Spaces

Anna Adnan, Matthew Barclay, Marco Caoduro, Joshua Childs, Will Evans, and Tao Gaede
1: University of Calgary, 2: University of Victoria, 3: The University of British Columbia

1. Introduction

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 DoMorgan’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), Gσ 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 F be a family of k interval-order subgraphs of G. Then F is a k-interval-order cover of G if \( \bigcap_{i \in F} E_i = E(G) \), for \( H_i \in F \).

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 \( \mathcal{C} \) be its complement. Then box(G) \( \leq k \) if and only if \( \mathcal{C} \) has a k-interval-order cover.

Now, 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.

Finding a Minimal Covering with Maximal Interval-Order Subgraphs:

**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 \( \alpha_c(G) \), is the minimum number of ants needed to cover the edges of G.

2. Background and Methods

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 tree, 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 \( V(A) = \{u\} \cup \{v\} \) and edge set \( E(A) = \{u \cup v\} \). We call the edge 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.

**Definition (Ant-Packaging Number)**: Let G be a graph. The ant-packaging 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 \( \alpha_c(G) \), is the minimum number of ants needed to cover the edges of G.

3. Main Result

Relationship between the boxicity of the complement of T, box(T), and the ant-covering number, \( \alpha_a(G) \), and the ant-covering number, \( \alpha_c(G) \), of a tree T. The boxicity of the complement of T, box(T), is upper bounded by the ant-covering number, \( \alpha_a(T) \), and lower bounded by the ant-packaging number, \( \alpha_a(T) \), i.e. \( \alpha_a(T) \leq \alpha_c(T) \leq \alpha_a(T) \).

We will see that the aforementioned algorithm proves that for a tree, the ant-packaging number and the ant-covering number are equal: Theorem: Let T be a forest, then \( \alpha_a(T) = \alpha_c(T) \). Now, because for a tree T, \( \alpha_a(T) \leq \alpha_c(T) \) and the algorithm proves that \( \alpha_a(T) = \alpha_c(T) \), we have \( \alpha_a(T) = \alpha_c(T) \). Therefore as a Corollary of the aforementioned Theorem, the algorithm allows us to compute the boxicity of the complement of a tree, \( \text{box}(T) \), 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 use 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 i vertex, there will be an ant for the covering and packing, therefore we can see that \( \alpha_a(T) = \alpha_c(T) \).
2. For the tree T, rooted at r, select all almost leaves furthest away from the root vertex.
3. Label each almost leaf, \( u_i \), where \( h \) is the height of the tree, and \( s \) is the almost-leaf count for the tree T.
4. Choose a non-leaf neighbour of \( u_i \), and label it \( v_{i,s} \). Subsequently, choose a neighbour of \( v_{i,s} \), which is a leaf and call it \( u_{i,s} \).
5. Use u and v to construct an uv-ant.
6. We define a class C, \( C = C(u, v) \), where the bodies of the ants used for covering are stored, and a class P, \( P = P(u, v_i) \), 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 \( T_{i,s} \) as \( T_i \) minus the edges of the \( u_i, v_{i,s} \) ant.
8. Repeat this process.

5. Future Directions

Given that trees are outer planar graphs, we can prove 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