Exponential Bounds on Graph Enumerations from Vertex Incremental Characterizations

Jérémie Lumbroso* Jessica Shi†

Abstract

In this paper, building on previous work by Nakano *et al.* [23], we develop an alternate technique which almost automatically translates (existing) vertex incremental characterizations of graph classes into asymptotics of that class. Specifically, we construct trees corresponding to the sequences of vertex incremental operations which characterize a graph class, and then use analytic combinatorics to enumerate the trees, giving an upper bound on the graph class. This technique is applicable to a wider set of graph classes compared to the tree decompositions, and we show that this technique produces accurate upper bounds.

We first validate our method by applying it to the case of distance-hereditary graphs, and comparing the bound obtained by our method with that obtained by Nakano et al. [23], and the exact enumeration obtained by Chauve *et al.* [7, 8]. We then illustrate its use by applying it to switch cographs, for which there are few known results: our method provide a bound of $\sim 6.301^n$, to be compared with the precise exponential growth, $\sim 6.159^n$, which we obtained independently through the relationship between switch cographs and bicolored cographs, first introduced by Hertz [19].

We believe the popularity of vertex incremental characterizations might mean this may prove a fairly convenient to tool for future exploration of graph classes.

1 Introduction

Much about trees—their enumeration and asymptotics—is generally well understood; thus, a particularly powerful approach to graph enumeration has been tree decomposition: a bijection establishes a correspondence between a family of graphs with a family of trees, and we study the family of trees. Two well-known examples of such decompositions are the modular decomposition, and the split decomposition [12, 18]. The latter was recently used by Chauve *et al.* [7, 8], to obtain an exact enumeration of an important class of (perfect) graphs, the distance hereditary graphs.

Interestingly Chauve *et al.* built on work by Nakano *et al.* [23], which approximated the distance-hereditary graphs by encoding their construction sequence of operations as a tree; they then used a compact encoding to find a bound for the number of such trees. While this approach proved less accurate—only able to approximate rather than enumerate the distance-hereditary graphs—it's generalization seems to be both directly amenable to tree enumeration and more easily extensible.

The operations in this constructive sequence, are called vertex-incremental operations and they build the graph, by repeated application of any of a (fixed) subset of operation taken from Table 1 to a growing graph starting with a single node. *Vertex incremental* (or *one-vertex extension*) characterizations are the necessary and sufficient conditions under which adding a vertex to a graph in a certain class would produce another graph in that class, and that this operation is generative of the set [4, 24]. [1] A characterization can be viewed algorithmically, as a set of operations under which the class is closed. As such, it is possible to exhaustively enumerate graphs in a certain class using its vertex incremental characterization — this provides us a reference enumeration for small sizes of a graph class. It is also possible to describe the sequence of operations as a tree, to the extent that we need not count the graphs but the combination of operations which builds these graphs (these combinations may provide a superset).

We call these trees *vertex incremental trees* [5], and they are structures that encode the vertex incremental operations used to construct the corresponding graphs. Historically, this idea first emerged in the enumeration of cographs [11]. More recently, Nakano *et al.* [23] used a similar idea for distance-hereditary graphs to obtain an upper bound enumeration. Specifically, Nakano *et al.* used *compact encoding* to enumerate a superset of the vertex incremental trees.

*Dept. of Computer Science, Princeton University, 35 Olden Street, Princeton, NJ 08540, USA, lumbroso@cs.princeton.edu

†Dept. of Mathematics, Princeton University, Fine Hall, Washington Road, Princeton, NJ 08540, USA, jessicashi@princeton.edu

[1] Like tree decompositions, vertex incremental characterizations have led to algorithmic improvements on certain graph classes. For example, the vertex incremental characterization of distance-hereditary graphs [1] has had applications in obtaining a linear-time algorithm for the domination problem [6] and in deriving linear-time algorithms for weighted vertex cover problems and computing a minimum fill-in and treewidth [5].

1.1 Our results. We use analytic combinatorics here instead, since it is much more powerful in that it can fully characterize certain classes of trees, and as such tends to provide either exact enumerations or much tighter asymptotic bounds. We demonstrate on *switch cographs* that it is relatively straightforward to obtain a good characterization of the underlying vertex incremental trees. We derive an upper bound of $O(6.301^n)$ on these trees. While we do not have an exact enumeration of switch cographs to serve as a point of reference, note that switch cographs are closely related to bicolored cographs [19]. As such, we use the number of bicolored cographs to precisely bound the number of switch cographs, giving us $\Omega(6.159^n)$.

We also derive an upper bound of $O(7.250^n)$ on the vertex incremental trees for connected distance-hereditary graphs. We compare this to the upper bound obtained by Nakano *et al.* [23] using compact encoding and to the exact enumeration obtained by Chauve *et al.* [8] using graph-labeled trees, of $O(12.042^n)$ and $O(7.213^n)$ respectively.

On a higher level, previous results, including those by Nakano *et al.* [23], approach the problem of enumeration by first finding a bijection between graph classes and tree classes. Our work on vertex incremental trees is constructive, and we focus instead on obtaining a surjection. These trees are naturally less accurate in terms of giving a precise upper bound, but are much more straightforward to enumerate. This constructive focus is particularly relevant in light of recent results on the preferential attachment model [2, 3].

In Section 2, we introduce *vertex incremental* characterizations and we detail the characterizations for some common graph classes, and specifically, *switch cographs*. In Section 3, we discuss prior work by Nakano *et al.* [23] and derive an upper bound on the number of distance-hereditary graphs using analytic combinatorics. In Section 4, we derive a precise bound on the number of switch cographs through their relationship to bicolored cographs, which was first introduced by Hertz [19] and later refined by de Montgolfier and Rao [13]. We then use the vertex incremental characterization of switch cographs in conjunction with analytic combinatorics to derive an upper bound enumeration.

2 Vertex incremental characterizations and trees

In this section, we introduce preliminary definitions regarding *vertex incremental* characterizations, the main graph classes we use throughout this paper, and the construction of vertex incremental trees.

2.1 Principles and operations. For the purposes of this paper, every graph $G = (V, E)$ is *simple*, that is to say, undirected, unlabeled, without self-loops, and without multiple edges. For a vertex $v \in V$, the *neighborhood* of v, denoted by $N(v)$, is defined to be the set of vertices in $V \setminus \{v\}$ that are adjacent to v. Also, for a set of vertices $U \subseteq V$, the *induced subgraph* is the graph H with vertices U where $\{u, v\}$ is an edge in H if and only if it is an edge in G.

A *vertex incremental characterization* of a class of graphs $\mathcal{A}$ is the necessary and sufficient conditions under which adding a vertex v to a graph from $\mathcal{A}$ would produce another graph from $\mathcal{A}$ [18].

The characterizations are often written as a set of operations, which when repeatedly applied to a starting graph of one vertex (either in a specified order or in any order), would produce all graphs in $\mathcal{A}$ (considering all possible combinations of applying these operations).

First, we present various common vertex incremental operations here:

- Pick a target vertex x in G, add a new vertex x' to G:
  - *Pendant.* Add edge $\{x, x'\}$.
  - *Strong twin.* Add an edge $\{x, x'\}$ between the target and new vertex, and for all neighbors of x, $\forall y \in N(x)$, add an edge $\{x', y\}$.
  - *Weak twin.* For all neighbors of x, $\forall y \in N(x)$, add an edge $\{x', y\}$.
  - *Strong anti-twin.* Add an edge $\{x, x'\}$ between the target and new vertex, and for all non-neighbors of x, $\forall y \notin N(x)$, add an edge $\{x', y\}$.
  - *Weak anti-twin.* For all non-neighbors of x not including x, $\forall y \notin N(x) \cup \{x\}$, add an edge $\{x', y\}$.
- Pick a set of weak twins $X = \{x_1, x_2, \ldots, x_n\}$ in G—where $x_i \in X$ and $x_j \in X$ iff $N(x_i) = N(x_j)$ and $\{x_i, x_j\} \notin E(G)$:
  - *Bipartite.* Add a bipartite graph $B = (V_1 \cup V_2, E)$ to G (where V_1 and V_2 is a bipartition of B such that every edge in E connects a vertex from V_1 to a vertex from V_2), identifying certain vertices in V_1 with the weak twins of X.

These operations are illustrated in Table 1, and Table 2 describes certain well-known vertex incremental characterizations of graph classes which are the subject of the next subsection.

2.2 Characterization of certain classes of graphs. We now provide definitions for the main graph classes we consider throughout this paper, along with their vertex incremental characterizations.

A *distance-hereditary graph* is a graph in which every induced path is the shortest path [20]. That is, for any two vertices u, v and any induced subgraph containing those vertices, the path between u and v in the induced subgraph is in fact the shortest path between u and v in the original graph.

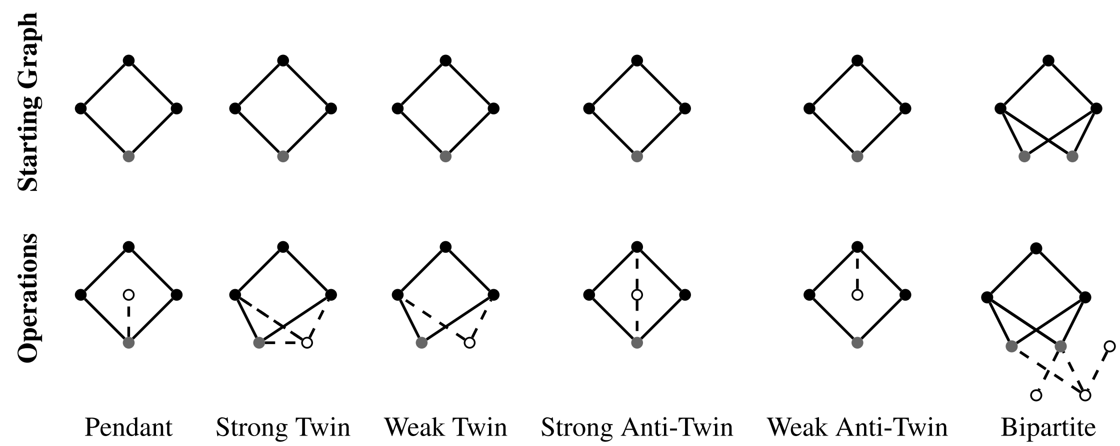


Table 1. Vertex incremental operations: in gray, the target vertex that is picked, and on which the extension will be done; in white, the vertex x' that is added with its corresponding dashed edges. In the special case of the bipartite extension, we may have several target vertices (which must be weak twins of each other) and we may add several vertices $\{x_1, \ldots, x_n\}$.

Graph Class	**Pendant**	**Twin**		**Anti-Twin**		**Bipartite**
		strong	weak	strong	weak	
3-Leaf Power [17]	(1)	(2)				
Cograph [23]		✓	✓			
Distance-Hereditary [1]	✓	✓	✓			
Switch Cograph [14]		✓	✓	✓	✓	
(6, 2)-Chordal Bipartite [9]	✓		✓			
Parity [10]		✓	✓			✓

Table 2. Vertex-incremental characterizations of certain classes of graphs. The numbers denote operations that must be performed in a certain order (*e.g.*, 3-leaf power graphs are obtained from a single vertex by some number of pendant operations followed by some number of true twin operations), while the checkmarks denote operations that can be performed in any order.

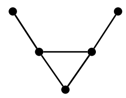


Figure 1. Bull graph.

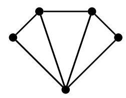


Figure 2. Gem graph.

Alternatively, a distance-hereditary graph can be viewed as a graph that is constructed using some sequence[2] of pendant, strong twin, and weak twin operations, which was first introduced by Bandelt and Mulder [1].

DEFINITION 2.1. A *switch cograph*, also known as a *(C_5, bull, gem, co-gem)-free graph*, is a graph which does not contain as induced subgraphs a cycle graph of length 5, a bull graph, a gem graph, or the complement of a gem graph [19]. A *cycle* graph of length n is a graph that consists of a single cycle, with n vertices. A *bull* graph is the graph shown in Figure 1. A *gem* graph is the graph shown in Figure 2.

Alternatively, a switch cograph can be viewed as a graph that is constructed using some sequence of strong twin, weak twin, strong anti-twin, and weak anti-twin vertex incremental operations, which was first introduced by de Montgolfier and Rao [14].

We also provide a definition for a graph class that is closely related to switch cographs, known as *cographs*.

DEFINITION 2.2. A *cograph*, also known as a *P_4-free* graph, is a graph which does not contain a path on 4 vertices (alternatively, a path of length 3) as an induced subgraph.

Alternatively, a cograph can be viewed as a graph that is constructed using some sequence of strong twin and weak twin vertex incremental operations, which was first introduced by Corneil *et al.* [11].

2.3 Vertex incremental trees. Nakano *et al.* [23] used the vertex incremental characterization of distance-hereditary graphs to construct corresponding *DH-trees*. In this paper, we generalize their characterization and introduce the notion of *vertex incremental trees*, which encode the vertex incremental operations used to construct a given graph and can be applied to other graph classes [3].

DEFINITION 2.3. *A* vertex incremental tree *is a rooted, ordered tree in which each internal node n_i has at least two children and is labeled with a vertex incremental operation $\mathtt{op}(n_i)$, and each leaf node is unlabeled. Each vertex incremental tree has a corresponding graph, in which the leaf nodes n_j are in bijection with the vertices $v(n_j)$ of the graph and the internal nodes are in bijection with the vertex incremental operations used to construct the graph.*

For a graph class $\mathcal{A}$, the corresponding class of vertex incremental trees is precisely the trees in which the internal nodes are labeled with the vertex incremental operations that characterize $\mathcal{A}$. Since multiple sequences of operations may correspond to the same graph, note that multiple vertex incremental trees may correspond to the same graph.

A well known example of vertex incremental trees are *cotrees*, or the modular decomposition trees of cographs. They represent a subclass of the vertex incremental trees for cographs, where the labels on the internal nodes correspond with the strong and weak twin operations. We discuss this in more detail in Section 4.3.

We can construct a vertex incremental tree as follows. In general, we can define a graph $G = (V, E)$ by the sequence of vertex incremental operations $\mathtt{op}_1, \ldots, \mathtt{op}_m$ used to construct G. Each operation $\mathtt{op}_i$ is applied in order to some vertex v_i, where we may have $v_i = v_j$ for $i \neq j$; *e.g.*, if the graph is constructed by repeatedly applying an operation to the same vertex, then $v_1 = \ldots = v_m$. $\mathtt{op}_i(v_i)$ produces a pair (V_i, E_i), which represent the vertices and edges added in the operation respectively; *e.g.*, a pendant operation on a vertex v_i adds a vertex v_i' and an edge $\{v_i, v_i'\}$, so we would have $V_i = \{v_i\}$ and $E_i = \{\{v_i, v_i'\}\}$. Note that V_i may be ordered, which allows E_i to be unordered.[4] Then,

1. Start with a tree T consisting of a single root node, n_1.
2. For every operation $\mathtt{op}_i$:
   (a) Let n_i be the leaf node corresponding to the vertex v_i ($v(n_i) = v_i$). Label n_i with $\mathtt{op}_i$ ($\mathtt{op}(n_i) := \mathtt{op}_i$).
   (b) Define the children of n_i to be the sequence of nodes n_i, n_j for each $v_j \in V_i$. The corresponding vertex to each child n_k is the vertex v_k ($v(n_k) := v_k$).

When we read a vertex incremental tree, we note that each leaf node n_i is associated with a vertex in the graph $v(n_i) = v_i$. Extend this function v to every internal node in the tree, such that the vertex associated with each internal node is

[2]Note that several sequences may generate the same graph, and the number of sequences that generate any given graph is not fixed.

[3]Nakano *et al.* [23] define DH-trees constructively, and obtain a bijection between distance-hereditary graphs and DH-trees. They then use several properties of these DH-trees to obtain an upper bound. Unlike Nakano *et al.*, we define vertex incremental trees as a superclass of DH-trees, and observe their properties as *normalizations*. These vertex incremental trees are more naturally suited to obtaining upper bounds, but do not give a bijection with connected distance-hereditary graphs.

[4]This characterization is not entirely correct, since it does not allow for the bipartite operation. The bipartite operation may apply to a subset of vertices, and must also be associated with a fixed bipartite graph. We can extend the concept of vertex incremental trees to this operation by allowing a subset of vertices to act as a single parent node and including bipartite graphs in its labels; while this does not create a formal tree, it can still be described completely using analytic combinatorics. For the purposes of this paper, we omit such extensions.

precisely the vertex associated with its leftmost child. Then, we can read a vertex incremental tree as follows:

1. Start with the root node and a graph G consisting of a single vertex, v.
2. Traverse tree in level order; for each internal node n:
   (a) Apply the operation in the node's label $\mathtt{op}(n)$ to its corresponding vertex $v(n)$.[5]

There is at least one straightforward way to improve these trees. If in our vertex incremental characterization we have a pair of consecutive operations such that $\mathtt{op}_i = \mathtt{op}_{i+1}$ and $v_i = v_{i+1}$, then in constructing the tree, we can define the children of n_i to be the sequence of nodes n_i, n_j for each $v_j \in V_i$, and n_k for each $v_k \in V_{i+1}$, essentially combining the two operations. In reading the tree, the number of children of any given node implies the number of times in which the operation is applied.

2.4 Exponential growth. Given a class of graphs $\mathcal{A}$, we can consider its enumeration as a number sequence $\{a_n\}$ where a_n is the number of objects in $\mathcal{A}$ of size n. More formally, the *ordinary generating function* (OGF) of $\mathcal{A}$ is given by the power series $\mathcal{A}(z) = \sum_{n=0}^{\infty} a_n z^n$. We then define the *exponential growth* of A to be k^n if $a_n = k^n \cdot \theta(n)$, where $\theta(n)$ is a *subexponential factor*, satisfying $\limsup |\theta(n)|^{1/n} = 1$ [16]. Equivalently, $\limsup |a_n|^{1/n} = k$.

It is useful to note that because these graphs are in bijection with trees (modulo symmetries), Meir and Moon's theorem [22] on simple varieties of trees has well established that all trees share similar asymptotic growth schemes.

3 Distance-hereditary graphs

In this section, we illustrate our method by deriving an exponential upper bound for distance-hereditary graphs using their vertex incremental characterization, extending upon results by Nakano *et al.* [23]. We then compare this result with its exact enumeration, derived by Chauve *et al.* [8].

3.1 Prior work. We begin by recalling the vertex incremental characterization for distance-hereditary graphs, which we discussed in Section 2.2 and which was first derived by Bandelt and Mulder [1]. The characterization involves the following operations:

- Pick a target vertex x in G, add a new vertex x' to G:

[5]We may be applying operations in a slightly different order than given in constructing the tree. As long as we read the tree in level order, this does not change the underlying graph. The proof of this follows from the fact that operations associated with disjoint , since all added vertices must treat all neighbors of $v(n)$ the same (and similarly for non-neighbors). We omit this proof.

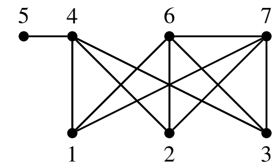


Figure 3. Example of a connected distance-hereditary graph.

Pendant. Add edge $\{x, x'\}$.

Strong twin. Add an edge $\{x, x'\}$ between the target and new vertex, and for all neighbors of x, $\forall y \in N(x)$, add an edge $\{x', y\}$.

Weak twin. For all neighbors of x, $\forall y \in N(x)$, add an edge $\{x', y\}$.

Now, the vertex incremental trees for distance-hereditary graphs have internal nodes labeled with P, ${}^sT$, and ${}^wT$, corresponding to the pendant, strong twin, and weak twin operations respectively.

Notice that multiple vertex incremental trees may correspond to the same distance-hereditary graph. For example, in Figures 4 and 5, we have two vertex incremental trees for connected distance-hereditary graphs which correpond to the same connected distance-hereditary graph, in Figure 3[6].

As such, we apply the following normalizations, given by Nakano *et al.*:

- DH-1. *Commutativity of twins.* The children of a node labeled ${}^wT$ or ${}^sT$ are unordered.
- DH-2. *Commutativity of pendants.* The non-leftmost children of a node labeled P are unordered.[7]
- DH-3. *Connectivity.* The root is not labeled ${}^wT$.
- DH-4. *Associativity of twins.* No child of a node labeled ${}^wT$ can be labeled ${}^wT$, and no child of a node labeled ${}^sT$ can be labeled ${}^sT$.
- DH-5. Any non-leftmost child of a node labeled P cannot labeled ${}^wT$.
- DH-6. If the root has 2 children, it is labeled ${}^sT$.
- DH-7. If the root has 2 children, the labels of the children are either both ${}^wT$ or both P.

3.2 Enumerating distance-hereditary graphs using their vertex incremental characterization. We begin by introducing one additional normalization rule for vertex incremental trees of connected distance-hereditary graphs, namely:

[6] The graphs we consider are unlabeled; we have labeled the vertices of Figure 3 so that it is clear which vertex corresponds to which leaf of the vertex incremental trees in Figures 4 and 5.

[7]The leftmost child of a node labeled P is still ordered; this is the reason why we bold and number the leftmost edge of a node labeled P in our trees.

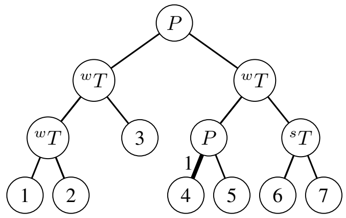


Figure 4. Example of a vertex incremental tree for connected distance-hereditary graphs, corresponding to the graph in Figure 3. Importantly, this tree is not normalized. Its root node breaks normalization DH-6 and the left child of the root node breaks normalization DH-4.

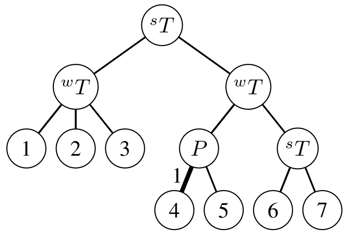


Figure 5. Example of a normalized vertex incremental tree for connected distance-hereditary graphs, corresponding to the graph in Figure 3.

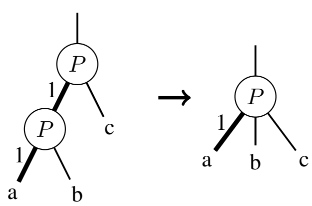


Figure 6. Normalizing a node labeled P with a leftmost child labeled P.

DH-8. *Associativity of pendants*. The leftmost child of a node labeled P cannot be labeled P.

Proof. We can replace a subtree of a node labeled P with a leftmost child labeled P with an equivalent normalized subtree, as shown in Figure 6. □

We obtain the following grammar for normalized vertex incremental trees for connected distance-hereditary graphs $\mathcal{DH}_T$, which is also an upper bound for the number of connected distance-hereditary graphs:

$$\mathcal{DH}_T = \mathcal{PR} + \mathcal{SR} + \mathcal{Z}$$

$$\begin{aligned}\mathcal{PR} &= (\mathcal{S} + \mathcal{W} + \mathcal{Z}) \times \text{SET}_{\geq 2}(\mathcal{P} + \mathcal{S} + \mathcal{Z}) \\ \mathcal{SR} &= \text{SET}_{\geq 3}(\mathcal{P} + \mathcal{W} + \mathcal{Z}) + \text{SET}_{=2}(\mathcal{W}) + \text{SET}_{=2}(\mathcal{P}) \\ &\quad + \text{SET}_{=2}(\mathcal{Z})\end{aligned}$$

$$\begin{aligned}\mathcal{P} &= (\mathcal{S} + \mathcal{W} + \mathcal{Z}) \times \text{SET}_{\geq 1}(\mathcal{P} + \mathcal{S} + \mathcal{Z}) \\ \mathcal{S} &= \text{SET}_{\geq 2}(\mathcal{P} + \mathcal{W} + \mathcal{Z}) \\ \mathcal{W} &= \text{SET}_{\geq 2}(\mathcal{P} + \mathcal{S} + \mathcal{Z})\end{aligned}$$

The following list explains the symbols for each class:

- $\mathcal{PR}$: A root node with label P
- $\mathcal{SR}$: A root node with label ${}^sT$
- $\mathcal{P}$: An internal node with label P
- $\mathcal{S}$: An internal node with label ${}^sT$
- $\mathcal{W}$: An internal node with label ${}^wT$
- $\mathcal{Z}$: An unlabeled leaf node

Proof. This follows directly from the normalizations for vertex incremental trees for connected distance-hereditary graphs. □

Using Maple with the `combstruct` package, we obtain the following enumeration:

COROLLARY 3.1. *The first few terms of the upper bound OGF of the class of normalized vertex incremental trees for connected distance-hereditary graphs, which is also an upper bound OGF of the class of connected distance-hereditary graphs, is*

$$\mathcal{DH}_T = z + z^2 + 2z^3 + 10z^4 + 48z^5 + 270z^6 + \ldots$$

Using Maple, we plotted the ratio between the first 500 consecutive terms of the generating function, obtained in Corollary 3.1; this ratio is known to tend towards the exponential growth of the class. We compared this plot to that of the ratio of the first 500 consecutive terms of the exact generating function for connected distance-hereditary graphs, which was discovered by Chauve *et al.* [8]. We also compared this to the plot of the upper bound on the number of connected distance-hereditary graphs obtained by Nakano *et al.* [23], using normalized vertex incremental trees and compact encoding. The overlaid plots are shown in Figure 7.

Fusy and Lumbroso [21] wrote Maple code to apply the asymptotic treatment in Section 4 of Chauve *et al.* [8] (which follows Drmota [15]) to a broad set of grammars (including any grammar of a vertex incremental tree). We utilize this to compute the following bound:

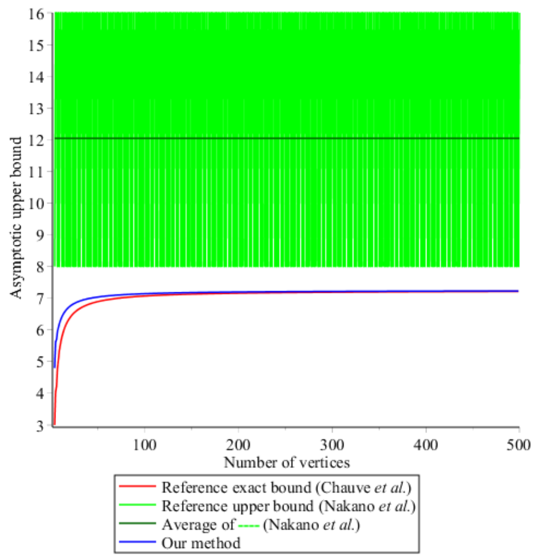


Figure 7. Comparison of the upper bound asymptotic for connected distance-hereditary graphs, obtained using normalized vertex incremental trees and analytic combinatorics, with exact bounds obtained by Chauve *et al.* [8] and previous upper bounds obtained by Nakano *et al.* [23] through normalized vertex incremental trees and compact encoding.

COROLLARY 3.2. *The upper bound on the number of connected distance-hereditary graphs, obtained using their vertex incremental characterization and analytic combinatorics, is given by* $O(7.250^n)$.

Note that this upper bound is relatively close to the actual bound of $O(7.213^n)$ and is a significant improvement over Nakano *et al.*'s upper bound of approximately $O(12.042^n)$, using compact encodings.

4 Switch cographs

In this section, we illustrate our method by deriving an exponential upper bound for this class based on its vertex incremental characterization. Because we know of no exact enumeration for this class, we first use a surjection by De Montgolfier and Rao [13], to provide a precise reference exponential order of growth with which to compare our upper bound result.

4.1 The Seidel reduction. We begin by introducing an operation which links switch cographs to bicolored cographs, known as the *Seidel reduction* [13].

DEFINITION 4.1. Given a graph G and a vertex $r \in V(G)$, the *Seidel reduction* of G on r, noted $\overline{G}^r$, is the graph obtained from G by:

(a) complementing[8] all edges between $N_G(r)$ and $N_G(r)^c$;
(b) and removing vertex r.

Remark 1. The notion of Seidel reduction is often confused with the related notion of *Seidel switch* [26]: given a graph $G = (V, E)$ and a vertex subset $W \subset V$, the Seidel switch produces a graph on the same vertex set, $G' = (V, E')$, such that edge $\{x, y\} \in E'$ if and only if:

- either $\{x, y\} \in E$ and $\{x, y\} \cap W = \emptyset$;
- or $\{x, y\} \notin E$ and $\{x, y\} \cap W \neq \emptyset$.

Seidel used this operation to define an equivalence relation between graphs, wherein two graphs are equivalent if there exists a vertex subset such that one graph can be obtained from the other by a Seidel switch. The class of switch cographs is noteworthy in this regard because they are an equivalence class for the Seidel switch—they are in fact the only such equivalence class known to be *perfect* [19, §2].

In the Seidel reduction, we pick a vertex r, apply the Seidel switch with $W = N_G(r)$, and remove r (which at that point has been disconnected from the graph).

The class of switch cographs was long known only by its forbidden induced subgraph characterization: (C_5, bull, gem, co-gem)-free graphs. De Montgolfier and Rao [13] showed that for any such graph G, the Seidel reduction of G on any of its vertices is a cograph.

This is how the class got its new name: they are the *switch* cographs, because they are the class of graphs which, after applying the switching transformation, yields cographs.

THEOREM 4.1. *For a graph G, the following statements are equivalent:*

1. *G can be obtained from a single vertex by a sequence of strong twin, weak twin, strong anti-twin, and weak anti-twin operations.*
2. *For all* $r \in V(G)$*, the Seidel reduction* $\overline{G}^r$ *is a cograph.*

While this theorem was proven by De Montgolfier and Rao [13, Thm. 20], we provide a full standalone proof of it in Appendix B.

We now claim a stronger statement, namely that as long as there exists a vertex $r \in V(G)$ such that $\overline{G}^r$ is a cograph, then G is a switch cograph. It then follows from Theorem 4.1 that for all vertices $r \in V(G)$, $\overline{G}^r$ is a cograph.

LEMMA 4.1. *Let G be a graph. If there exists a vertex* $r \in V(G)$ *such that* $\overline{G}^r$ *is a cograph, then G is a switch cograph.*

[8] In more rigorous terms, let $x \in N_G(r)$ be a vertex connected to r, and $y \in N_G(r)^c$ a vertex not connected to r, then $\{x, y\} \in E(\overline{G}^r)$ if and only if $\{x, y\} \notin E(G)$. Thus we remove the edge between x and y if it exists, and add if it doesn't.

Proof. This proof follows directly from our proof in Appendix B of $(2) \Rightarrow (1)$ in Theorem 4.1, since in that proof it suffices to make the hypothesis that there exists a vertex $r \in V(G)$ such that $\overline{G}^r$. $\square$

4.2 From switch cographs to cographs.

With the results of the previous subsection, we can now, for any switch cograph G, represent G as a bicolored cograph.

Consider any $r \in V(G)$ and the cograph $\overline{G}^r$. We can color the vertices $\overline{G}^r$ in the following manner. For any vertex $v \in V(\overline{G}^r)$:

- if $\{r, v\} \in E(G)$, then color the vertex v in black;
- otherwise, color the vertex v in white.

In this way, the color of vertex v in the cograph $\overline{G}^r$ indicates whether it was connected to the removed vertex r in the original switch cograph (black) or not (white).

We now prove that this bicolored cograph exactly represents the switch cograph G, and indeed all bicolored cographs represent switch cographs in this manner.

LEMMA 4.2. *Let H be a bicolored cograph and let V_B be the set of black vertices in H, and V_W be the set of white vertices in H. Let $\phi(H)$ be an operation which constructs an (uncolored) graph G from H by complementing all of the edges between V_B and V_W, adding a vertex v, and adding all edges $\{v, x\}$ for all $x \in V_B$.*

Then, the function ϕ on the set of bicolored cographs is well-defined, and for every bicolored cograph H, $\phi(H)$ is a switch cograph.

Proof. By construction, ϕ is a well-defined function (since it is deterministic). Moreover, by definition of ϕ, for every bicolored cograph H, $\phi(H)$ is a graph in which there exists a vertex $v \in V(\phi(H))$ such that $\overline{\phi(H)}^v$ is a cograph. By Lemma 4.1, $\phi(H)$ is a switch cograph. $\square$

Remark 2. Theorem 4.1 guarantees that any bicolored cograph (*i.e.*, any Cartesian product of a cograph of size n, and one of the 2^n colorings of the graph's vertices in black or white) is a valid pre-image to a switch cograph.

We now show that ϕ provides an upper bound on the number of switch cographs, in terms of bicolored cographs.

LEMMA 4.3. *Let s_n be the number of switch cographs with n vertices and let b_{n-1} be the number of bicolored cographs with $n-1$ vertices. Then,*

$$s_n \leqslant b_{n-1} \leqslant n \cdot s_n. \tag{4.1}$$

[9]The left graph was obtained by applying a Seidel reduction on the bottom-left vertex of the switch cograph; the right graph was obtained by applying a Seidel reduction on the top-left vertex (note that the vertex in white was not adjacent to it in the switch cograph)

Proof. Without loss of generality, let us fix n, and define ϕ_n as the restriction of ϕ to the set of bicolored cographs with $n-1$ vertices. Thus for any graph H in its domain, $\phi_n(H)$ is a switch cograph with n vertices.

Let G be a switch cograph (with n vertices), so by Theorem 4.1, for all vertices $r \in V(G)$, $\overline{G}^r$ is a cograph with $n-1$ vertices—which is to say that there is at least one such vertex r—and the product of the cograph $\overline{G}^r$ and the choice of a vertex r expresses a bicolored cograph with $n-1$ vertices. By consequence, every switch cograph G with n vertices has at least one corresponding bicolored cograph with $n-1$ vertices, and ϕ_n is a surjective function.

Thus, we have $s_n \leqslant b_{n-1}$, where b_{n-1} is the order of the domain of ϕ_n and s_n is the order of the range of ϕ_n.

We now claim that for any switch cograph G with n vertices, there are at most n bicolored cographs H (irrespective of their number of vertices) such that $\phi(H) = G$.

Assume by contradiction that there are at least $n+1$ bicolored cographs H such that $\phi(H) = G$. First, by definition of ϕ, which is an operation that adds exactly one vertex to the graph H it is given, the considered bicolored graphs must have exactly $n-1$ vertices.

There are at most n induced subgraphs of G on $n-1$ vertices— these are the induced subgraphs in which we have removed exactly one vertex. For our assumption to be true, there would have to be one such induced subgraph that has two colorings that encode the adjacencies of the removed vertex: in other words, two ways of adding a vertex to an identical cograph H, would yield the same switch cograph, which is impossible. Thus, for each switch cograph G, there are at most n bicolored cographs H such that $\phi(H) = G$. It follows that $b_{n-1} \leqslant n \cdot s_n$. $\square$

4.3 Exponential bounds through bicolored cographs.

The transformation ϕ introduced in Lemma 4.2 is not a bijection, because ϕ is not injective; this point is illustrated in Figure 8, in which two bicolored cographs correspond to the same switch cograph.

Importantly, this overcounting is not systematically the same. Indeed, the figure provides an example of a switch cograph of size $n = 4$ which has two bicolored graph pre-images; but there are other switch cographs of the same size which only have one pre-image by ϕ (for example, K_4 the complete graph on four vertices). As such, it is non-trivial to deduce an exact enumeration of switch cographs from the exact enumeration of bicolored cographs. We can however use the following facts:

1. both the enumeration [11] and asymptotics [25] for cographs are well-known;
2. if there are c_n cographs of size n, there are $b_n = 2^n \cdot c_n$ bicolored cographs of size n;
3. by Lemma 4.3, the enumeration of bicolored cographs

(a) Bicolored cographs, both of which are obtained[9] from the switch cograph in Figure 8b. **(b)** A switch cograph.

Figure 8. The transformation ϕ described in Lemma 4.3, which yields a switch cograph, from a specially bicolored cographs in which some vertices are colored black and others are colored white. This transformation uses the equivalence of Theorem 4.1 that states a switch cograph G yields a cograph by Seidel reduction $\overline{G}^r$ over any vertex r: in the obtained cograph, we color all the vertices that used to be adjacent to r in black, and all others in white.

has the same exponential growth as the enumeration of switch cographs (this also means their associated generating functions have the same dominant singularity ρ).

Thus, we now are going to enumerate bicolored cographs exactly, using analytic combinatorics. To do so, we use a tree representation of cographs, known as *cotrees*, which was first introduced by Corneil *et al.* [11], and which uniquely represents cographs[10].

DEFINITION 4.2. A *cotree* is a rooted, unordered tree in which each internal node is labeled ${}^sT$ or ${}^wT$ and each leaf node is unlabeled. Each internal node must have at least 2 children, and for each parent, its child may not have the same label. There is a bijection between cotrees and cographs [11].

Figure 9 shows an example of a cograph, and Figure 10 shows an example of the corresponding cotree.

Remark 3. A cotree is exactly the modular decomposition tree of a cograph; the internal nodes labeled with ${}^sT$ correspond to the *series node* of the modular decomposition tree, and the internal nodes labeled with ${}^wT$ correspond to the *parallel nodes* of the modular decomposition tree [25].

We obtain the following grammar for bicolored cotrees, $\mathcal{BC}$:[11]

$$\begin{aligned}\mathcal{BC} &= \mathcal{S} + \mathcal{W} + \mathcal{Z}\\ \mathcal{S} &= \textsc{Set}_{\geq 2}\left(\mathcal{W} + \mathcal{Z}\right)\\ \mathcal{W} &= \textsc{Set}_{\geq 2}\left(\mathcal{S} + \mathcal{Z}\right)\\ \mathcal{Z} &= \mathcal{Z}_{\text{white}} + \mathcal{Z}_{\text{black}}\end{aligned}$$

The following list explains the symbols for each class:

- $\mathcal{S}$ An internal node with label ${}^sT$.
- $\mathcal{W}$ An internal node with label ${}^wT$.
- $\mathcal{Z}$ An unlabeled leaf node.
- $\mathcal{Z}_{\text{white}}$ An unlabeled leaf node, colored white (which corresponds with white vertices in the cograph).
- $\mathcal{Z}_{\text{black}}$ An unlabeled leaf node, colored black (which corresponds with black vertices in the cograph).

Proof. This follows directly from the definition of cotrees. Since the vertices of the associated cograph can be colored black or white, we allow the leaf nodes of cotrees to also bear a color: and this is expressed by the fact that a leaf can be either one of two unit size elements, $\mathcal{Z}_{\text{white}}$ or $\mathcal{Z}_{\text{black}}$. □

Using Maple with the combstruct package, we received the following enumeration:

COROLLARY 4.1. *The first few terms of the OGF of the class of bicolored cographs is*

$$\mathcal{BC} = 2z + 6z^2 + 20z^3 + 80z^4 + 340z^5 + 1570z^6 + \ldots$$

Note that by Lemma 4.3, the number of bicolored cographs gives a precise bound on the number of switch cographs. As in Corollary 3.2, we use Fusy and Lumbroso [21]'s Maple code to compute an upperbound on the number of switch cographs:

COROLLARY 4.2. *The precise bound on the number of switch cographs, as obtained using bicolored cographs, is given by* $\Omega(6.159^n)$.

4.4 Enumerating switch cographs using their vertex incremental characterization. We now reiterate the vertex incremental characterization for switch cographs, which we discussed in Section 2.2 and which was first defined by de Montgolfier and Rao [14]. The characterization involves the following operations:

- Pick a target vertex x in G, add a new vertex x' to G:
  - *Strong twin.* Add an edge $\{x, x'\}$ between the target and new vertex, and for all neighbors of x, $\forall y \in N(x)$, add an edge $\{x', y\}$.
  - *Weak twin.* For all neighbors of x, $\forall y \in N(x)$, add an edge $\{x', y\}$.
  - *Strong anti-twin.* Add an edge $\{x, x'\}$ between the target and new vertex, and for all non-neighbors of x, $\forall y \notin N(x)$, add an edge $\{x', y\}$.

[10] These trees are exactly the same as the vertex incremental trees for cographs later developed by Nakano *et al.* [23].

[11] Note that when we use SET in our grammar, we are simply indicating that there is no order to the children. As such, when unlabeled leaf nodes are involved, we are referring to MSET rather than PSET.

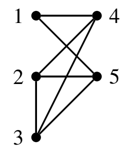


Figure 9**.** Example of a cograph, which corresponds to the cotree in Figure 10.

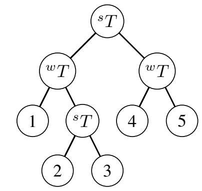


Figure 10**.** Example of a cotree, which represents the cograph in Figure 9.

Weak anti-twin. For all non-neighbors of x not including $x, \forall y \notin N(x) \cup \{x\}$, add an edge $\{x', y\}$.

We derive an exponential upper bound on switch cographs by constructing trees from their vertex incremental characterization. This argument is parallel to that in Section 3.2.

The vertex incremental trees for switch cographs have internal nodes labeled with ${}^{s}T$, ${}^{w}T$, ${}^{s}\overline{T}$, or ${}^{w}\overline{T}$, which correspond to the operations strong twin, weak twin, strong anti-twin, or weak anti-twin respectively. For any given normalization, we define the *conjugate normalization* to be that in which the labels ${}^{s}T$ and ${}^{w}T$ are switched, and similarly for ${}^{s}\overline{T}$ and ${}^{w}\overline{T}$; this new notion allows us to express the normalizations more succinctly.

The following normalizations apply:

SC-1. *Commutativity of twins*. The children of a node labeled ${}^{s}T$ or ${}^{w}T$ are unordered.

SC-2. *Commutativity of anti-twins*. The non-leftmost children of a node labeled ${}^{s}\overline{T}$ or ${}^{w}\overline{T}$ are unordered.

SC-3. The non-leftmost children of a node labeled ${}^{s}\overline{T}$ cannot be labeled ${}^{w}T$. The conjugate is also a normalization.

SC-4. The root is not labeled ${}^{s}\overline{T}$ or ${}^{w}\overline{T}$.

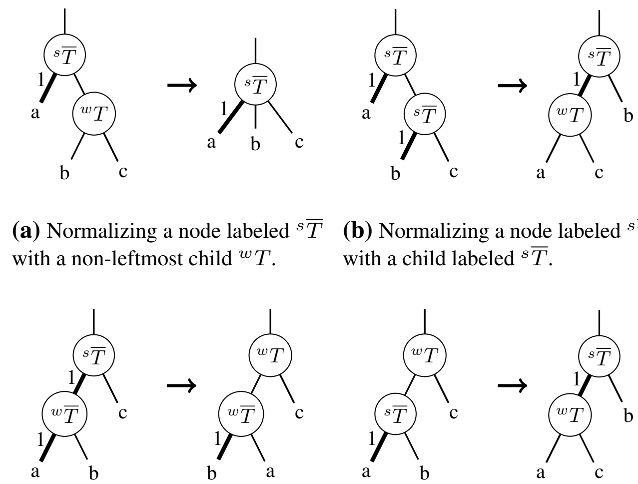
SC-5. *Associativity of anti-twins*. The children of a node labeled ${}^{s}\overline{T}$ cannot be labeled ${}^{s}\overline{T}$. The conjugate is also a normalization.

SC-6. The children of a node labeled ${}^{s}\overline{T}$ cannot be labeled ${}^{w}\overline{T}$. The conjugate is also a normalization.

SC-7. *Associativity of twins*. The children of a node labeled ${}^{s}T$ cannot be labeled ${}^{s}T$. The conjugate is also a normalization.

SC-8. *Operator associativity of twins and anti-twins*. The children of a node labeled ${}^{w}T$ cannot be labeled ${}^{s}\overline{T}$. The conjugate is also a normalization.

Proof. Normalizations SC-1 and SC-7 are equivalent to those found in Nakano *et al.*'s [23] work on vertex incremental trees for distance-hereditary graphs. For the remaining normalizations, we replace any subtree with the specified property with an equivalent normalized subtree. For example, in



(a) Normalizing a node labeled ${}^{s}\overline{T}$ with a non-leftmost child ${}^{w}T$.

(b) Normalizing a node labeled ${}^{s}\overline{T}$ with a child labeled ${}^{s}\overline{T}$.

(c) Normalizing a node labeled ${}^{s}\overline{T}$ with a child labeled ${}^{w}\overline{T}$.

(d) Normalizing a node labeled ${}^{w}T$ with a child labeled ${}^{s}\overline{T}$.

Figure 11**.** These figures show the replacements for normalizations SC-3, SC-5, SC-6, and SC-8 respectively. Note that the conjugate figures prove the conjugate normalizations.

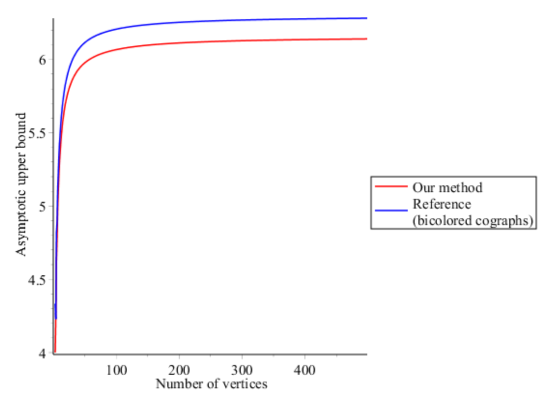


Figure 12**.** Comparison of the upper bound asymptotic for switch cographs, obtained using normalized vertex incremental trees and analytic combinatorics, with the precise bound asymptotic for switch cographs, obtained using bicolored cographs.

normalization SC-3, any subtree with a root node labeled ${}^s\overline{T}$ such that one of its non-leftmost children is labeled ${}^wT$ can be replaced with the subtree in Figure 11a.

Figures 11a, 11b, 11c, and 11d show the replacements for normalizations SC-3, SC-5, SC-6, and SC-8 respectively. Note that the conjugate figures prove the conjugate normalizations.

For normalization SC-6, note that we must replace a subtree with a normalized subtree with a different root node, shown in Figure 11c. As a result, we must modify the labels of all internal nodes on the path from the root node of the subtree to the root node of the tree, from ${}^sT$ to ${}^w\overline{T}$ and vice versa, and from ${}^wT$ to ${}^s\overline{T}$ and vice versa.

Finally, for normalization SC-4, note that if the root node is labeled ${}^s\overline{T}$, it is equivalent to the same vertex incremental tree with the root node labeled ${}^sT$. The conjugate statement holds as well. Normalization SC-2 follows directly from the definition of anti-twins. □

We obtain the following grammar for normalized vertex incremental trees for switch cographs $\mathcal{SC}_T$, which by construction is also an upper bound for switch cographs.

$$\begin{aligned}
\mathcal{SC}_T &= \mathcal{ST} + \mathcal{WT} + \mathcal{Z}\\
\mathcal{ST} &= \textsc{Set}_{\geq 2}\left(\mathcal{WT} + \mathcal{SA} + \mathcal{Z}\right)\\
\mathcal{WT} &= \textsc{Set}_{\geq 2}\left(\mathcal{ST} + \mathcal{WA} + \mathcal{Z}\right)\\
\mathcal{SA} &= \left(\mathcal{ST} + \mathcal{WT} + \mathcal{Z}\right) \times \textsc{Set}_{\geq 1}\left(\mathcal{ST} + \mathcal{Z}\right)\\
\mathcal{WA} &= \left(\mathcal{ST} + \mathcal{WT} + \mathcal{Z}\right) \times \textsc{Set}_{\geq 1}\left(\mathcal{WT} + \mathcal{Z}\right)
\end{aligned}$$

The following list explains the symbols for each class:

- $\mathcal{ST}$: An internal node with label ${}^sT$
- $\mathcal{WT}$: An internal node with label ${}^wT$
- $\mathcal{SA}$: An internal node with label ${}^s\overline{T}$
- $\mathcal{WA}$: An internal node with label ${}^w\overline{T}$
- $\mathcal{Z}$: An unlabeled leaf node

Proof. This follows directly from the normalizations for vertex incremental trees for switch cographs. □

Using Maple with the `combstruct` package, we obtained the following enumeration:

COROLLARY 4.3. *The first few terms of the OGF of the class of normalized vertex incremental trees for switch cographs, which is also an upper bound OGF of the class of switch cographs, is*

$$\mathcal{SC}_T = z + 2z^2 + 6z^3 + 26z^4 + 110z^5 + 530z^6 + \ldots$$

As previously, using Maple, we plotted the ratio between the first 500 consecutive terms of the generating function obtained in Corollary 4.3, and we compared the plot to that of the ratio between the first 500 consecutive terms of the generating function for bicolored cographs, obtained in Corollary 4.1. Recall that the number of bicolored cographs provides a precise bound on the number of switch cographs, by Lemma 4.3. We overlaid both plots, which is shown in Figure 12.

We again use the Maple code written by Fusy and Lumbroso [21] to compute an upper bound:

COROLLARY 4.4. *The upper bound on the number of switch cographs, using their vertex incremental characterization and analytic combinatorics, is given by* $O(6.301^n)$.

Note that this upper bound is relatively close to the precise bound of $\Omega(6.159^n)$, obtained in Corollary 4.2[12].

5 Conclusion

In this work, we have studied the methodology of using vertex incremental characterizations to construct trees, which we call vertex incremental trees, corresponding to certain graph classes and subsequently enumerating those trees using analytic combinatorics to obtain upper bounds.

We first investigated connected distance-hereditary graphs, and obtained an upper bound of $O(7.250^n)$, in comparison with the actual bound of $O(7.213^n)$ as obtained by Chauve *et al.* [8]. This improves upon Nakano *et al.*'s [23] previous upper bounds on their version of these vertex incremental trees, using compact encoding, of $O(12.042^n)$.

We also investigated switch cographs, and received an upper bound of $O(6.301^n)$, in comparison with a precise bound of $O(6.159^n)$, the latter of which we obtained using their relationship to bicolored cographs.

In the future, we would like to extend our work to consider other classes of graphs which may be more difficult to construct vertex incremental trees from. Specifically, the vertex incremental characterization of parity graphs includes the bipartite operation, which is difficult to integrate given the current definition of vertex incremental trees and which may be difficult to normalize as well. This is because the bipartite operation acts on a set of vertices rather than a single vertex, and given the isomorphisms that may or may not arise by attaching bipartite graphs, there are no clear normalizations.

A further extension is to obtain an experimental understanding of how each operation contributes to multiplying the underlying enumeration. Any subset of the vertex incremental operations across distance-hereditary graphs and switch cographs could be considered with the applicable normalizations, and the resulting grammars would encapsulate the effect of any given operation. Furthermore, it may be

[12]We have also obtained the grammars and the asymptotics which arise after each normalization step, thus obtaining a measure of the optimization gained from each normalization step. This is in Appendix A.

possible to obtain a direct transfer theorem that does not require trees as intermediate representations.

Moreover, we would like to compare the upper bounds obtained from these vertex incremental trees with upper bounds obtained from graph-labeled trees and classical tree decompositions. In particular, switch cographs are the totally decomposable graphs of the bijoin decomposition; it may be possible to characterize the bijoin decomposition in terms of a graph-labeled tree, and subsequently obtain an upper bound on the number of switch cographs.

Acknowledgment

We thank Fabien de Montgolfier for pointing us towards the idea of using bicolored cographs as a reference enumeration. Also, we thank the anonymous reviewers of this paper for their helpful and insightful comments.

References

[1] Hans-Jürgen Bandelt and Henry Martyn Mulder. Distance-Hereditary Graphs. *Journal of Combinatorial Theory, Series B*, 41(2):182–208, 1986.

[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[3] Noam Berger, Christian Borgs, Jennifer T. Chayes, and Amin Saberi. Asymptotic behavior and distributional limits of preferential attachment graphs. *Ann. Probab.*, 42(1):1–40, 01 2014.

[4] Anne Berry, Pinar Heggernes, and Yngve Villanger. A vertex incremental approach for maintaining chordality. *Discrete Mathematics*, 306(3):318–336, 2006.

[5] Maw-Shang Chang, Sun-yuan Hsieh, and Gen-Huey Chen. Dynamic programming on distance-hereditary graphs. In Hon Wai Leong, Hiroshi Imai, and Sanjay Jain, editors, *Algorithms and Computation: 8th International Symposium, ISAAC '97 Singapore, December 17–19, 1997 Proceedings*, pages 344–353. Springer Berlin Heidelberg, Berlin, Heidelberg, 1997.

[6] Maw-Shang Chang, Shaur-Ching Wu, Gerard J. Chang, and Hong-Gwa Yeh. Domination in distance-hereditary graphs. *Discrete Applied Mathematics*, 116(1–2):103–113, 2002.

[7] Cédric Chauve, Éric Fusy, and Jérémie Lumbroso. An enumeration of distance-hereditary and 3-leaf power graphs. Preprint presented at ICGT 2014, 2014.

[8] Cédric Chauve, Éric Fusy, and Jérémie Lumbroso. An exact enumeration of distance-hereditary graphs. In *Proceedings of the Fourteenth Workshop on Analytic Algorithmics and Combinatorics, ANALCO 2017, Barcelona, Spain, Hotel Porta Fira, January 16-17, 2017*, pages 31–45, 2017.

[9] Serafino Cicerone and Gabriele Di Stefano. Graph classes between parity and distance-hereditary graphs. *Discrete Applied Mathematics*, 95:197–216, 1999.

[10] Serafino Cicerone and Gabriele Di Stefano. On the extension of bipartite to parity graphs. *Discrete Applied Mathematics*, 95:181–195, 1999.

[11] Derek G. Corneil, Helmut Lerchs, and Lorna Stewart Burlingham. Complement reducible graphs. *Discrete Applied Mathematics*, 3(3):163–174, 1981.

[12] William H. Cunningham. Decomposition of Directed Graphs. *SIAM Journal on Algebraic Discrete Methods*, 3(2):214–228, 1982.

[13] Fabien de Montgolfier and Michaël Rao. Bipartitive families and the bi-join decomposition. submitted to special issue of Discrete Mathematics devoted to ICGT05, 2005.

[14] Fabien de Montgolfier and Michaël Rao. The bi-join decomposition. *Electronic Notes in Discrete Mathematics*, 22:173–177, 2005.

[15] Michael Drmota. Systems of Functional Equations. *Random Structures & Algorithms*, 10(1-2):103–124, 1997.

[16] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009.

[17] Emeric Gioan and Christophe Paul. Dynamic Distance Hereditary Graphs Using Split Decomposition. In Takeshi Tokuyama, editor, *Algorithms and Computation: 18th International Symposium, ISAAC 2007, Sendai, Japan, December 17-19, 2007. Proceedings*, pages 41–51. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[18] Emeric Gioan and Christophe Paul. Split decomposition and graph-labelled trees: Characterizations and fully dynamic algorithms for totally decomposable graphs. *Discrete Applied Mathematics*, 160(6):708–733, 2012.

[19] Alain Hertz. On perfect switching classes. *Discrete Applied Mathematics*, 94(1):3–7, 1999.

[20] Edward Howorka. A Characterization of Distance-Hereditary Graphs. *The Quarterly Journal of Mathematics*, 28(4):417–420, December 1977.

[21] Jérémie O. Lumbroso. Automatic Extraction of the Growth Constant from a `Combstruct` Grammar. `https://github.com/jlumbroso/maple-treelike-exponential-growth`, 2014.

[22] A Meir and J.W Moon. On an asymptotic method in enumeration. *Journal of Combinatorial Theory, Series A*, 51(1):77–89, 1989.

[23] Shin-ichi Nakano, Ryuhei Uehara, and Takeaki Uno. A New Approach to Graph Recognition and Applications to Distance-Hereditary Graphs. *Journal of Computer Science and Technology*, 24(3):517–533, 2009.

[24] Michaël Rao. Clique-width of graphs defined by one-vertex extensions. *Discrete Mathematics*, 308(24):6157–6165, 2008.

[25] Vlady Ravelomanana and Loÿs Thimonier. Asymptotic enumeration of cographs. *Electronic Notes in Discrete Mathematics*, 7(0):58–61, 2001.

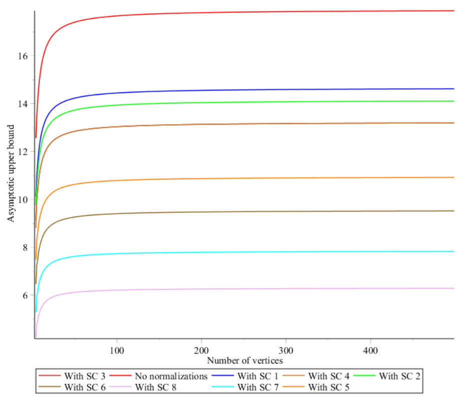
[26] Johan J. Seidel. A survey of two-graphs. *Colloquio Internazionale sulle Teorie Combinatorie*, 1(17):481–511, 1973.

A Stepwise normalizations for switch cographs

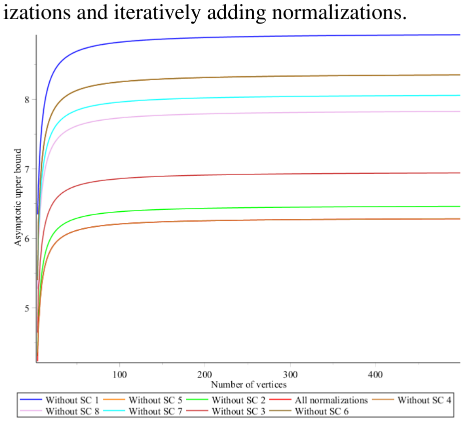
One of the main obstacles to obtaining good bounds using this methodology is finding the appropriate normalizations.

However, normalizations are often intuitive and represent basic properties, such as associativity and commutativity. We note that each normalization only improves the bound by a small amount, and we show these stepwise improvements.

In Figure 13a, we start with the bounds obtained from the vertex incremental trees with no normalizations, and add a normalization at every step; the final bound represents the fully normalized trees. In Figure 13b, we start with the bounds obtained from the fully normalized vertex incremental trees, and display the bounds obtained from the fully normalized trees excepting the stated normalization.



(a) Upper bound asymptotics for switch cographs, starting from vertex incremental trees with no normalizations and iteratively adding normalizations.



(b) Upper bound asymptotics for switch cographs, displaying from fully normalized vertex incremental trees and fully normalized vertex incremental trees excepting each normalization.

Figure 13. These figures attempt to capture the relative "importance" of each normalization insofar as reducing the number of times a graph is a over-counted, because of various structral symmetries.

B Proof of Theorem 4.1

Proof. We first show that the vertex incremental characterization implies the Seidel reduction characterization. We proceed by induction on the number of vertices, $n = |V(G)|$. The base case is trivial.

Assume that for any graph G, $n < |V(G)|$, obtained from a sequence of strong twin, weak twin, strong anti-twin, and weak anti-twin operations, for any $r \in V(G)$, $\overline{G}^r$ is a cograph. We now show that this property holds when we extend G to graph G' by adding a new vertex w.

A few preliminary remarks:

- The vertex r was so-named because it is *removed*.
- The graph $\overline{G}^r$ is an induced subgraph of $\overline{G'}^r$ with
$$V(\overline{G'}^r) = V(\overline{G}^r) \cup \{w\}.$$
Thus, to describe $\overline{G'}^r$, it suffices to begin with $\overline{G}^r$ and describe how vertex w was added to $\overline{G}^r$.
- There are exactly four ways to add w, namely by the operations strong twin, weak twin, strong anti-twin, and weak anti-twin to some existing vertex $u \in V(G)$.
- For each of those vertex incremental operations, we must consider the two cases $u = r$ and $u \neq r$.
- We can ignore the case $w = r$, since adding w as a strong/weak twin/anti-twin of u is equivalent to adding u as a strong/weak twin/anti-twin of w—as such, the case is symmetric to the case $u = r$.
- We use two characterizations of cographs in this proof (see for example, Corneil *et al.* [11]):
  (i) they are the P_4-free graphs (a path of size 4);
  (ii) they are the graphs which can be obtained from a single vertex by a sequence of strong and weak twin operations (and so cographs are closed by those operations).
- Finally, recall $N_G(r)$ is the neighborhood of vertex r in graph G, and it *excludes* r; $N_G[r]$ is the closed neighborhood of r, and it *includes* r.

We may then proceed by looking at each of the eight cases:

(a) w is a STRONG TWIN of u.

$u = r$: We have $w \in N_{G'}(r)$, and for all $x \in N_G(r)^c$ such that $x \neq r$, we have $\{w, x\} \notin E(G')$.

Thus, in the Seidel reduction, we add edges $\{w, x\}$ for all $x \in N_G(r)^c$ such that $x \neq r$. As such, $\overline{G'}^r$ can be constructed from $\overline{G}^r$ by adding the vertex w and adding all edges $\{w, x\}$ for all vertices $x \in V(\overline{G}^r)$.

The graph $\overline{G'}^r$ cannot contain an induced P_4. Indeed, no induced P_4 of $\overline{G'}^r$ can include w, as w is adjacent to every other vertex of $\overline{G'}^r$ (and so w would create a cycle in any induced subgraph containing more than two vertices). Thus, an induced P_4 of $\overline{G'}^r$ would also have to be an induced P_4 of $\overline{G}^r$, which is a contradiction of our inductive hypothesis, since $\overline{G}^r$ is a cograph. Thus, $\overline{G'}^r$ must be a cograph (a P_4-free graph).

$u \neq r$: By definition, as w is a strong twin of u, we have $N_{G'}[u] = N_{G'}[w]$ (*i.e.*, the closed neighborhood including u and w).

After the Seidel reduction, we still have $N_{\overline{G'}^r}[u] = N_{\overline{G'}^r}[w]$. This is clear because u and w have the same neighborhood, and are thus either both connected to the removed vertex r, or both not connected—that is they are either both in $N_{G'}(r)$ or both in $N_{G'}(r)^c$. Thus:

- if $x \notin N_{G'}[u]$ and an edge $\{u, x\}$ is added in the reduction, we have $x \notin N_{G'}[w]$ and $\{w, x\}$ is added;
- and if $x \in N_{G'}[u]$ and an edge $\{u, x\}$ is removed in the reduction, we have $x \notin N_{G'}[w]$ and $\{w, x\}$ is removed.

In the end, $\overline{G'}^r$ can be constructed from $\overline{G}^r$ by adding the vertex w as a strong twin of u. Cographs are closed by strong twin operation: so applying the strong twin operation to a cograph produces another cograph, and thus $\overline{G'}^r$ is a cograph.

(b) w is a WEAK TWIN of u.

$u = r$: We have $w \in N_{G'}(r)^c$, and for all $x \in N_G(r)$, we have $\{w, x\} \in E(G')$.

Similar to the case of the strong twin operation, with $u = r$. Here, the added vertex w ends up being connected to no other vertex in the graph after Seidel reduction, $\overline{G'}^r$: so by inductive hypothesis $\overline{G}^r$ is a cograph, and since no new P_4 induced subgraph can have been created $\overline{G'}^r$ remains a cograph.

$u \neq r$: By definition, as w is a weak twin of u, we have $N_{G'}(u) = N_{G'}(w)$.

Again, as in the analogous case for the strong twin operation, with $u \neq r$, this allows us to show that $\overline{G'}^r$ can be constructed from $\overline{G}^r$ by adding the vertex w as a weak twin. As cographs are closed by weak twin operation, we conclude $\overline{G'}^r$ is a cograph.

(c) w is a STRONG ANTI-TWIN of u.

$u = r$: We have $w \in N_{G'}(r)$, and for all $x \in N_G(r)^c$, we have $\{w, x\} \in E(G')$.

As such, in the Seidel reduction, we remove edges $\{w, x\}$ for all $x \in N_G(r)^c$. As such, $\overline{G'}^r$ can be constructed from $\overline{G}^r$ simply by adding the vertex w (and adding no edges). This is exactly what occurs when w is a weak twin of u and $u = r$, so $\overline{G'}(r)$ is a cograph.

$u \neq r$: We claim that, here too, $\overline{G'}^r$ can be constructed from $\overline{G}^r$ by adding w as a weak twin of u. This allows us to use the same argument, that cographs are closed by the weak twin operation, and so, by induction, $\overline{G'}^r$ is a cograph.

Consider that since u and w are strong anti-twins, their neighboorhoods are a partition of $V(G') \setminus \{u, w\}$, and exactly one of them is connected to r—in other words, either $u \in N_{G'}(r)$ and $w \in N_{G'}(r)^c$, or vice versa. Suppose, without loss of generality, that u is connected to r.

After the Seidel reduction, the neighborhood of u, which is connected to r, will be complemented, and both u and w will have the same neighborhood. Indeed, for any vertex x:

- if edge $\{u, x\}$ is changed after the reduction (that is, added or removed), then $x \in N_{G'}(r)^c$ and edge $\{w, x\}$ is unaffected.
- conversely, if edge $\{u, x\}$ is unchanged after the reduction, then $x \in N_{G'}(r)$ and edge $\{w, x\}$ is changed (that is, added or removed).

Since the neighborhoods of u and w are a partition of $V(G') \setminus \{u, w\}$, in both cases, after the Seidel reduction, u and w have the same neighborhoods. Hence, $\overline{G'}^r$ can be constructed from $\overline{G}^r$ by adding the vertex w as a weak twin of u. Applying the weak twin operation to a cograph produces another cograph, so $\overline{G'}^r$ is a cograph.

(d) w is a WEAK ANTI-TWIN of u.

$u = r$: We have $w \in N_{G'}(r)^c$, and for all $x \in N_G(r)$, we have $\{w, x\} \notin E(G')$.

Similar to the case of the strong anti-twin operation, with $u = r$. Here, the added vertex w ends up being connected to every vertex in the graph after the Seidel reduction, $\overline{G'}^r$. This is exactly what occurs when w is a strong twin of u and $u = r$, so $\overline{G'}(r)$ is a cograph.

$u \neq r$: Again, as in the analogous case for the strong anti-twin operation, with $u \neq r$, this allows us to show that $\overline{G'}^r$ can be constructed from $\overline{G}^r$ by adding w as a strong twin of u. As cographs are closed by the strong twin operation, we conclude $\overline{G'}^r$ is a cograph.

This concludes all possible cases, so $\overline{G'}^r$ is a cograph. Thus, the vertex incremental characterization implies the Seidel reduction characterization.

We now show the Seidel reduction characterization implies the vertex incremental characterization. We again proceed by induction on the number of vertices, $n = |V(G)|$. The base case is trivial.

Assume that for any graph G, $n < |V(G)|$, such that for all $r \in V(G)$, $\overline{G}^r$ is a cograph, G can be obtained from a sequence of strong twin, weak twin, strong anti-twin, and weak anti-twin operations. We now show that this property holds for G', $|V(G')| = n + 1$.

Fix $r \in V(G')$. By the vertex incremental characterization of cographs, there exists $w, u \in V(\overline{G'}^r)$ such that w and u are either strong twins or weak twins.

Let G be the induced subgraph of G' on $V(G) \setminus \{w\}$. Note that:

- For all s, the graph $\overline{G}^s$ is an induced subgraph of $\overline{G'}^s$. Thus, $\overline{G}^s$ is a cograph, and by our inductive hypothesis, G is obtained from a sequence of strong twin, weak twin, strong anti-twin, and weak anti-twin operations.
- The graph $\overline{G}^r$ is an induced subgraph of $\overline{G'}^r$ with
$$V(\overline{G'}^r) = V(\overline{G}^r) \cup \{w\}.$$
Thus, to describe $\overline{G'}^r$, it suffices to begin with $\overline{G}^r$ and describe how vertex w was added to $\overline{G}^r$.
- There are exactly two ways to add w to $\overline{G}^r$, namely by the operations strong twin and weak twin to u.
- For each of those vertex incremental operations, we must consider the two cases $u, w \in N_{G'}(r)$ and $u \in N_{G'}(r)$, $w \in N_{G'}(r)^c$. The cases $u, w \in N_{G'}(r)^c$ and $u \in N_{G'}(r)^c$, $w \in N_{G'}(r)$ are respectively symmetric.
- Finally, recall $N_G(r)$ is the neighborhood of vertex r in graph G, and it *excludes* r; $N_G[r]$ is the closed neighborhood of r, and it *includes* r.

We may then proceed by looking at each of the four cases.

(a) w is a STRONG TWIN of u in $\overline{G'}^r$.

$u, w \in N_{G'}(r)$:

By definition, as w is a strong twin of u, we have $N_{\overline{G'}^r}[u] = N_{\overline{G'}^r}[w]$.

Prior to the Seidel reduction, we still have $N_{G'}[u] = N_{G'}[w]$. This is clear because:

- if $x \in N_{\overline{G'}^r}[u]$ and an edge $\{u, x\}$ did not exist prior to the reduction, we have $x \in N_{\overline{G'}^r}[w]$ and $\{w, x\}$ did not exist;
- and if $x \notin N_{\overline{G'}^r}[u]$ and an edge $\{u, x\}$ existed prior to the reduction, we have $x \notin N_{\overline{G'}^r}[w]$ and $\{w, x\}$ existed.

In the end, G' can be constructed from G by adding the vertex w as a strong twin of u.

$u \in N_{G'}(r)$,
$w \in N_{G'}(r)^c$:

Prior to the Seidel reduction, the neighborhood of u in $\overline{G'}^r$ is complemented, and the neighborhoods of u and w in G' are a partition of $V(G') \setminus \{u, w\}$. Indeed, for any vertex x,

- if edge $\{u, x\}$ is changed after the reduction (that is, added or removed), then $x \in N_{G'}(r)^c$ and the edge $\{w, x\}$ is unaffected.
- conversely, if edge $\{u, x\}$ is unchanged after the reduction, then $x \in N_{G'}(r)$ and edge $\{w, x\}$ is changed (that is, added or removed).

Since the neighborhoods of u and w in $\overline{G'}^r$ are the same, in both cases, prior to the Seidel reduction, the neighborhoods of u and w partition $V(G') \setminus \{u, w\}$. Hence, G' can be constructed from G by adding the vertex w as a weak anti-twin of u.

(b) w is a WEAK TWIN of u in $\overline{G}^r$.

$u, w \in N_{G'}(r)$:

By definition, as w is a weak twin of u, we have $N_{\overline{G'}^r}(u) = N_{\overline{G'}^r}(w)$.

Similar to the case of the strong twin operation, with $u, w \in N_{G'}(r)$. This allows us to show that G' can be constructed from G by adding the vertex w as a weak twin of u.

$u \in N_{G'}(r)$,
$w \in N_{G'}(r)^c$:

Again, as in the analogous case for the strong twin operation, with $u \in N_{G'}(r)$, $w \in N_{G'}(r)^c$, this allows us to show that G' can be constructed from G by adding the vertex w as a strong anti-twin of u.

In all cases, G' can be obtained from G by a strong twin, weak twin, strong anti-twin, or weak anti-twin operation. Since G is obtained from a sequence of these operations, so is G'. □