Automorphism Groups of Geometrically Represented Graphs*

Pavel Klavík and Peter Zeman

Computer Science Institute, Charles University in Prague,
Address, Czech Republic
`{klavik,zeman}@iuuk.mff.cuni.cz.`

Abstract

Interval graphs are intersection graphs of closed intervals and *circle graphs* are intersection graphs of chords of a circle. We study automorphism groups of these graphs. We show that interval graphs have the same automorphism groups as trees, and circle graphs have the same as pseudo-forests, which are graphs with at most one cycle in every connected component.

Our technique determines automorphism groups for classes with a strong structure of all geometric representations, and it can be applied to other graph classes. Our results imply polynomial-time algorithms for computing automorphism groups in term of group products.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases automorphism group, geometric intersection graph, interval graph, circle graph

Digital Object Identifier 10.4230/LIPIcs.STACS.2015.540

1 Introduction

The study of symmetries of geometrical objects is an ancient topic in mathematics and its precise formulation led to group theory. Symmetries play an important role in many distinct areas. In 1846, Galois used symmetries of the roots of a polynomial in order to characterize polynomials which are solvable by radicals. Some big objects are highly symmetrical, for instance the well-known Rubik's Cube has $43,252,003,274,489,856,000$ symmetries. They can be understood using group theory and used for working with the Rubik's Cube (designing algorithms for solving it, etc.). Symmetries have important applications in differential equations, physics, chemistry, crystallography, etc.

Automorphism Groups of Graphs. The symmetries of a graph X are described by its automorphism group $\mathrm{Aut}(X)$. Every automorphism is a permutation of the vertices which preserves adjacencies and non-adjacencies. Frucht [9] proved that every finite group is isomorphic to the automorphism group of some graph X. General mathematical structures can be encoded by graphs [18] while preserving automorphism groups.

Most graphs are asymmetric, i.e., have only the trivial automorphism [14]. However, many combinatorial and graph theory results rely on highly symmetrical graphs. Automorphism groups are important for studying large objects, since these symmetries allow one to simplify and understand the objects. This algebraic approach is together with the recursion and counting arguments the only technique known for working with big objects.

*Supported by CE-ITI (P202/12/G061 of GAČR) and Charles University as GAUK 196213. Many omited details and proofs are in the full version: `arXiv:1407.2136`.

32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015).
Editors: Ernst W. Mayr and Nicolas Ollinger; pp. 540–553
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany
SYMPOSIUM ON THEORETICAL ASPECTS OF COMPUTER SCIENCE

Highly symmetrical large graphs with nice properties are often constructed algebraically from small graphs. For instance, Hoffman-Singleton graph is a 7-regular graph of diameter 2 with 50 vertices [19]. It has 252000 automorphisms and can be constructed from 25 "copies" of a small multigraph with 2 vertices and 7 edges [26]. Similar constructions are used in designing large computer networks [7, 34]. For instance the well-studied degree-diameter problem asks, given integers d and k, to find a maximal graph X with diameter d and degree k. Such graphs are desirable networks having small degrees and short distances. Currently, the best constructions are highly symmetrical graphs made using groups [27].

For a class $\mathcal{C}$ of graphs, let $\mathrm{Aut}(\mathcal{C})$ denote its automorphism groups, i.e., $\mathrm{Aut}(\mathcal{C}) = \{\mathrm{Aut}(X) : X \in \mathcal{C}\}$. We say that a class $\mathcal{C}$ of graphs is *universal* if every finite group is isomorphic to some group in $\mathrm{Aut}(\mathcal{C})$, and *non-universal* otherwise.

The oldest non-trivial result concerning automorphism groups of restricted graph classes is for trees (TREE) by Jordan [21] from 1869. He proved that Aut(TREE) contains precisely those groups that can be obtained from the trivial group by a sequence of two operations: the direct product and the wreath product with a symmetric group. The direct product constructs the automorphisms that act independently on non-isomorphic subtrees and the wreath product constructs the automorphisms that permute isomorphic subtrees.

Graph Isomorphism Problem. This famous problem asks whether two input graphs X and Y are the same up to a relabeling. This problem is obviously in NP, and not known to be polynomially-solvable or NP-complete. Aside integer factorization, this is a prime candidate for an intermediate problem with the complexity between P and NP-complete. It belongs to the low hierarchy of NP [30], which implies that it is unlikely NP-complete. (Unless the polynomial-time hierarchy collapses to its second level.) The graph isomorphism problem is known to be polynomially solvable for the classes of graphs with bounded degree [24] and with excluded topological subgraphs [16].

The graph isomorphism problem is closely related to computing generators of an automorphism group. Assuming X and Y are connected, we can test $X \cong Y$ by computing generators of $\mathrm{Aut}(X \dot\cup Y)$ and checking whether there exists a generator which swaps X and Y. For the converse relation, Mathon [25] proved that generators of the automorphism group can be computed using $\mathcal{O}(n^4)$ instances of graph isomorphism. Compared to graph isomorphism, automorphism groups of restricted graph classes are much less understood.

Geometric Representations. In this paper, we study automorphism groups of geometrically represented graphs. The main question is how the geometry influences their automorphism groups. For instance, the geometry of a sphere translates to 3-connected planar graphs which have unique embeddings [32]. Thus, their automorphism groups are so called spherical groups which are automorphism groups of tilings of a sphere. For general planar graphs, the automorphism groups are more complex and they were described by Babai [1] using semidirect products of spherical and symmetric groups; see also [8].

We focus on intersection representations. An *intersection representation* $\mathcal{R}$ of a graph X is a collection $\{R_v : v \in V(X)\}$ such that $uv \in E(X)$ if and only if $R_u \cap R_v \neq \emptyset$; the intersections encode the edges. To get nice graph classes, one typically restricts the sets R_v to particular classes of geometrical objects; for an overview, see the classical books [15, 31]. We show that a well-understood structure of all intersection representations allows one to determine the automorphism group. In particular, we study interval graphs and circle graphs, and our technique can be also applied to other graph classes.

To obtain an *interval representation* of a graph, we restrict the sets R_v to closed intervals of the real line. In a *circle representation*, the sets R_v are chords of a circle. A graph is an

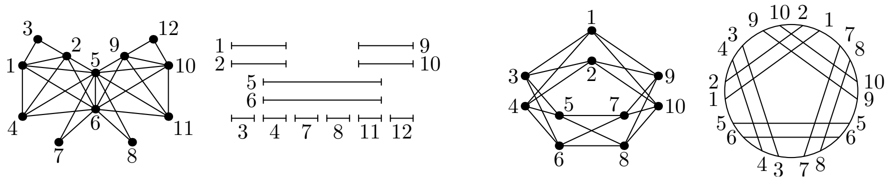


Figure 1 On the left, an interval graph and one of its interval representations. On the right, a circle graph and one of its circle representations.

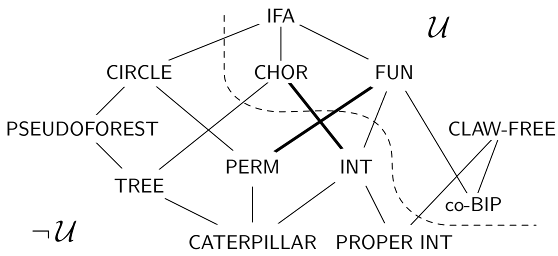


Figure 2 The inclusions between considered graph classes. We denote universal classes by $\mathcal{U}$, and non-universal by $\neg\mathcal{U}$. The bold edges are two infinite hierarchies, discussed in Section 6.

interval (resp. *circle*) *graph* if it has an interval (resp. circle) representation; see Fig. 1 for examples. We denote these classes by INT and CIRCLE, respectively.

Related Graph Classes. Figure 2 depicts graph classes important for this paper. *Caterpillar graphs* (CATERPILLAR) are trees with every leaf attached to a central path. They form the intersection of trees and interval graphs. *Chordal graphs* (CHOR) are intersection graphs of subtrees of trees. They contain no induced cycles of length four or more and naturally generalize interval graphs. Chordal graphs have universal automorphism groups [23].

Pseudoforests (PSEUDOFOREST) are graphs for which every connected component is a *pseudotree*, where pseudotree is a connected graph with at most one cycle. Each pseudoforest is a circle graph. The automorphism groups of pseudoforests can be constructed from the automorphism groups of trees by semidirect products with cyclic and dihedral groups, which constructs the automorphisms rotating/reflecting unique cycles.

Function graphs (FUN) are intersection graphs of continuous functions $f : [0,1] \to \mathbb{R}$. Equivalently, function graphs are *co-comparability graphs* which means their complements can be transitively oriented. Every interval graph is a co-comparability graph since disjoint pairs of intervals can be oriented from left to right. *Permutation graphs* (PERM) are function graphs which can be represented by linear functions.

Claw-free graphs (CLAW-FREE) are graphs with no induced $K_{1,3}$. Roberts proved [28] that CLAW-FREE $\cap$ INT is equal to the class of *proper interval graphs* (PROPER INT) which are interval graphs with representations in which no interval properly contains another. The complements of bipartite graphs (co-BIP) are universal. They are claw-free and contained in function graphs since each bipartite graph is transitively orientable.

Interval filament graphs (IFA) are intersection graphs of the following sets. For every R_u, we choose an interval $[a,b]$ and R_u is a continuous function $[a,b] \to \mathbb{R}$ such that $R_u(a) = R_u(b) = 0$ and $R_u(x) > 0$ for $x \in (a,b)$. They generalize circle, chordal, and function graphs.

▶ **Theorem 1.**
(i) $\mathrm{Aut}(\mathsf{INT}) = \mathrm{Aut}(\mathsf{TREE})$,
(ii) $\mathrm{Aut}(\text{connected } \mathsf{PROPER\ INT}) = \mathrm{Aut}(\mathsf{CATERPILLAR})$,
(iii) $\mathrm{Aut}(\mathsf{CIRCLE}) = \mathrm{Aut}(\mathsf{PSEUDOFOREST})$.

Concerning (i), this equality is not well known. It was stated by Hanlon [17] without a proof in the conclusion of his paper from 1982 on enumeration of interval graphs. Our structural analysis is based on PQ-trees [2] which combinatorially describe all interval representations of an interval graph. It explains this equality and further solves an open problem of Hanlon: for a given interval graph, to construct a tree with the same automorphism group. Without PQ-trees, this equality is surprising since these classes are very different. Caterpillar graphs which form their intersection have very limited groups and we characterize them in Lemma 5. The result (ii) easily follows from the known properties of proper interval graphs and our structural understanding of $\mathrm{Aut}(\mathsf{INT})$.

Using PQ-trees, Colbourn and Booth [4] give a linear-time algorithm to compute permutation generators of the automorphism group of an interval graph. In comparison, our description allows to construct an algorithm which outputs the automorphism group in the form of group products which reveals its structure.

Concerning (iii), we are not aware of any results on automorphism groups of circle graphs. One inclusion is trivial since $\mathsf{PSEUDOFOREST} \subsetneq \mathsf{CIRCLE}$. The other one is based on split-trees which describe all representations of circle graphs. The semidirect product with a cyclic or a dihedral group corresponds to the rotations/reflections of the central vertex of a split-tree. Geometrically, it corresponds to the rotations/reflections of the entire symmetric representation. Our approach is similar to the algorithm for circle graph isomorphism [20].

Structure. We describe the automorphism groups of interval graphs in Section 2 and of circle graphs in Section 3. In Section 4, we interpret our results in terms of actions of automorphism groups on sets of all representations. We explain our general technique for determining the automorphism group from the geometric structure of all representations. Further, we relate it to well-known results of map theory. Our results are constructive and lead to polynomial-time algorithms computing automorphism groups of interval and circle graphs; see Section 5. We conclude with several open problems.

Preliminaries. We use X and Y for graphs, M, T and S for trees and G, H and others for groups. The vertices and edges of X are $V(X)$ and $E(X)$. The set of all maximal cliques is denoted by $\mathcal{C}(X)$. A permutation π of $V(G)$ is an *automorphism* if $uv \in E(G) \iff \pi(u)\pi(v) \in E(G)$. We use $\mathbb{S}_n$, $\mathbb{D}_n$ and $\mathbb{Z}_n$ for the *symmetric*, *dihedral* and *cyclic groups*.

We quickly define semidirect and wreath products; see [3, 29] for details. Given two groups N and H, and a group homomorphism $\varphi\colon H \to \mathrm{Aut}(N)$, we can construct a new group $N \rtimes_\varphi H$ as the Cartesian product $N \times H$ with the operation defined as $(n_1, h_1) \cdot (n_2, h_2) = (n_1 \cdot \varphi(h_1)(n_2), h_1 \cdot h_2)$. The group $N \rtimes_\varphi H$ is called the *semidirect product of N and H with respect to the homomorphism φ*. The *wreath product* $G \wr \mathbb{S}_n$ is a shorthand for $G^n \rtimes_\psi \mathbb{S}_n$ where ψ is defined naturally by $\psi(\pi) = (g_1, \dots, g_n) \mapsto (g_{\pi(1)}, \dots, g_{\pi(n)})$.

2 Automorphism Groups of Interval Graphs

In this section, we prove Theorem 1(i) and (ii). We introduce PQ-trees which describe all interval representations. Using them, we derive a characterization of $\mathrm{Aut}(\mathsf{INT})$ which we prove to be equivalent to Jordan's characterization of $\mathrm{Aut}(\mathsf{TREE})$. We solve the open

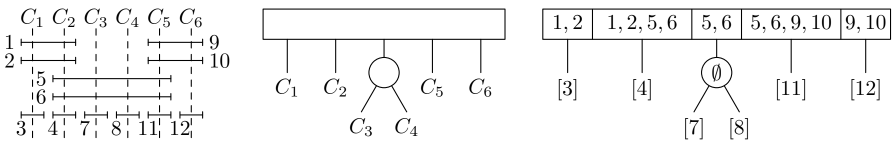


Figure 3 An ordering of the maximal cliques, and the corresponding PQ-tree and MPQ-tree. The P-nodes are denoted by circles, the Q-nodes by rectangles.

problem of Hanlon [17] by constructing for a given interval graph a tree with the same automorphism group, and we also show the converse construction.

PQ-trees. Booth and Lueker [2] invented a data structure called *PQ-tree* to solve the long-standing open problem of recognizing interval graphs in linear time. It is based on the following characterization of interval graphs.

▶ **Lemma 2** (Fulkerson and Gross [10]). *A graph X is an interval graph if and only if there exists an ordering of the maximal cliques such that for every $x \in V(X)$ the maximal cliques containing x appear consecutively.*

PQ-trees are rooted trees with two types of inner nodes: *P-nodes* and *Q-nodes*. The leaves correspond one-to-one to the maximal cliques of X. For every inner node, the order of its children is fixed. The order of the leaves from left to right is called a *frontier*. See Fig. 3.

There are two *equivalence transformations*: (i) an arbitrary permutation of the children of a P-node, and (ii) a reversal of the order of the children of a Q-node. Two PQ-trees are *equivalent* if we can get one from the other by a sequence of equivalence transformations. Booth and Lueker [2] proved that for every interval graph there exists a unique PQ-tree representing all possible orderings of the maximal cliques as frontiers of its equivalent trees. In other words, this PQ-tree encodes all interval representations.

Every automorphism $\alpha \in \mathrm{Aut}(X)$ induces some permutation of the maximal cliques $\mathcal{C}(X)$. However, multiple automorphisms can reorder $\mathcal{C}(X)$ in the same way. Two vertices are called *twin vertices* if they belong to the same maximal cliques. Two automorphisms of X can permute the maximal cliques the same but permute the twin vertices differently. PQ-trees describe the structure of the maximal cliques of an interval graph, but to determine $\mathrm{Aut}(X)$ we need some additional information about the twin vertices.

MPQ-trees. A *modified PQ-tree* is created from a PQ-tree by adding information about the vertices. They were described by Korte and Möhring [22] to simplify linear-time recognition of interval graphs. An equivalent idea was already used by Coulborn and Booth [4] for computing automorphism groups of interval graphs.

Suppose that T is a PQ-tree corresponding to an interval graph X. In the MPQ-tree M, we assign sets, called *sections*, to the nodes of T; see Fig. 3. The leaves and P-nodes have each assigned only one section, while Q-nodes have one section for every child. We assign these sections in the following way:

- For every leaf L, the section $\mathrm{sec}(L)$ contains those vertices that are only in the maximal clique represented by L, and no other maximal cliques.
- For every P-node P, the section $\mathrm{sec}(P)$ contains those vertices that are in all maximal cliques of the subtree of P, and no other maximal cliques.
- For every Q-node Q and its children $T_1, \dots, T_n$, the section $\mathrm{sec}_i(Q)$ contains those vertices that are in the maximal cliques represented by the leaves of the subtree of T_i and also

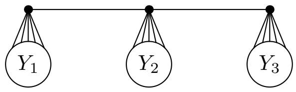
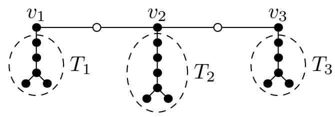
(a) 
(b) 

Figure 4 (a) Construction of the operation (d) from Lemma 4. (b) Trees attached to a path by their roots. Since the automorphism group is not isomorphic to $\big(\mathrm{Aut}(T_1)\times\mathrm{Aut}(T_2)\times\mathrm{Aut}(T_3)\big)\rtimes_\varphi\mathbb{Z}_2$, we fix it by subdividing v_1v_2 and v_2v_3.

some other T_j, but not in any other maximal clique outside the subtree of Q. We put $\mathrm{sec}(Q) = \mathrm{sec}_1(Q) \cup \cdots \cup \mathrm{sec}_n(Q)$.

Two vertices are in the same sections of an MPQ-tree if and only if they are twin vertices.

Automorphisms of PQ and MPQ-trees. Let T be a PQ-tree corresponding to an interval graph X. A sequence ε of equivalence transformations is an *automorphism of* T if there exists $\alpha \in \mathrm{Aut}(X)$ such that α reorders the maximal cliques $\mathcal{C}(X)$ in the same way as ε. We get a group homomorphism $\phi : \mathrm{Aut}(X) \to \mathrm{Aut}(T)$ where $\phi(\alpha)$ is the unique automorphism of T permuting $\mathcal{C}(X)$ the same as α. By the first isomorphism theorem, we have that $\mathrm{Aut}(T)$ is isomorphic to a subgroup of $\mathrm{Aut}(X)$.

Let M be the MPQ-tree with its nodes $N_1, \dots, N_k$. An *automorphism of a node* N is a permutation of the vertices inside the sections of N. For a P-node, $\mathrm{Aut}(N)$ is isomorphic to $\mathbb{S}_n$. For a Q-node, it is a direct product of symmetric groups. An *automorphism of* M is a $(k+1)$-tuple $(\nu_{N_1}, \dots, \nu_{N_k}, \varepsilon)$ where ν_{N_i} is an automorphism of the node N_i and ε is an automorphism of the underlying PQ-tree T. Each automorphism of N uniquely corresponds to an automorphism α of X, so $\mathrm{Aut}(M) \cong \mathrm{Aut}(X)$.

Automorphism Groups of Interval Graphs. To get $\mathrm{Aut}(X)$, we just need to determine $\mathrm{Aut}(M)$. We also make use of the following result due to Jordan:

▶ **Theorem 3** (Jordan [21]). *If $X_1, \dots, X_n$ are pairwise non-isomorphic connected graphs and X is the disjoint union of k_i copies of X_i, then* $\mathrm{Aut}(X) \cong \mathrm{Aut}(X_1) \wr \mathbb{S}_{k_1} \times \cdots \times \mathrm{Aut}(X_n) \wr \mathbb{S}_{k_n}$.

▶ **Lemma 4.** *A group $G \in \mathrm{Aut}(\mathsf{INT})$ if and only if $G \in \mathcal{I}$, where the class $\mathcal{I}$ is defined inductively as follows:*

(a) $\{1\} \in \mathcal{I}$.

(b) *If $G_1, G_2 \in \mathcal{I}$, then $G_1 \times G_2 \in \mathcal{I}$.*

(c) *If $G \in \mathcal{I}$ and $n \geq 2$, then $G \wr \mathbb{S}_n \in \mathcal{I}$.*

(d) *If $G_1, G_2, G_3 \in \mathcal{I}$ and $G_1 \cong G_3$, then $(G_1 \times G_2 \times G_3) \rtimes_\varphi \mathbb{Z}_2 \in \mathcal{I}$, where $\varphi\colon \mathbb{Z}_2 \to \mathrm{Aut}(G_1 \times G_2 \times G_3)$ is the homomorphism defined as $\varphi(0) = \mathrm{id}$ and $\varphi(1) = (g_1, g_2, g_3) \mapsto (g_3, g_2, g_1)$.*

Proof (Sketch). We first prove that $\mathcal{I} \subseteq \mathrm{Aut}(\mathsf{INT})$. Clearly $\{1\} \in \mathrm{Aut}(\mathsf{INT})$. It remains to show that the class $\mathrm{Aut}(\mathsf{INT})$ is closed under (b), (c) and (d). For (b), we can show this by attaching two interval graphs X_1 and X_2 on an asymmetric interval graph. Clearly, the resulting graph represents the direct product of $\mathrm{Aut}(X_1)$ and $\mathrm{Aut}(X_2)$. For (c), let $G \in \mathrm{Aut}(\mathsf{INT})$ and $n \geq 2$. There exists an interval graph Y such that $\mathrm{Aut}(Y) \cong G$. We construct X as the disjoint union of n copies of Y. By Theorem 3, it follows that $\mathrm{Aut}(X) \cong G \wr \mathbb{S}_n$. For (d), we construct an interval graph X by attaching X_1, X_2 and X_3 to a path as in Fig. 4a, where $\mathrm{Aut}(X_i) = G_i$ and $X_1 \cong X_3$. Then $\mathrm{Aut}(X) \cong (G_1 \times G_2 \times G_3) \rtimes_\varphi \mathbb{Z}_2$.

For the converse, we show that $\mathrm{Aut}(M) \in \mathcal{I}$. We have three cases for the root of M. For a P-node, $\mathrm{Aut}(M)$ is determined by the automorphism groups of its subtrees using

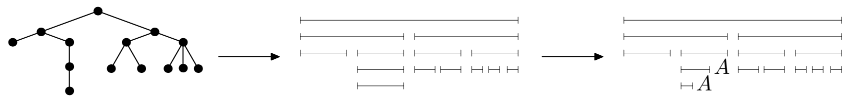


Figure 5 First, we place the intervals according to the structure of the tree. We get $\mathrm{Aut}(X) \cong \mathbb{S}_3 \times \mathbb{S}_2 \times \mathbb{S}_3$, but $\mathrm{Aut}(T) \cong \mathbb{S}_2 \times \mathbb{S}_3$. We fix this by adding copies of an asymmetric path A which has the trivial automorphism group.

Theorem 3, so the operations (b) and (c) are sufficient. For an asymmetric Q-node, $\mathrm{Aut}(M)$ is the direct product of the automorphism groups of its subtrees. For a symmetric Q-node, we apply the operation (d) where G_1 corresponds to the automorphisms of the left part of the Q-node, G_2 to the middle part and G_3 to the right part. The semidirect product with $\mathbb{Z}_2$ corresponds to reversing the Q-node. ◀

This lemma connects $\mathrm{Aut}(\mathsf{INT})$ and the geometrical structure of an interval representation. The operation (b) applies to non-isomorphic independent parts of the representation, (c) to isomorphic parts which can be arbitrary permuted, and (d) to parts which can only be reflected vertically.

Proof of Theorem 1(i). It easily follows from Lemma 4 that $\mathrm{Aut}(\mathsf{INT}) = \mathrm{Aut}(\mathsf{TREE})$. We show that (d) can be expressed using (b) and (c). Assuming $G_1 \cong G_3$, we get

$$(G_1 \times G_2 \times G_3) \rtimes_\varphi \mathbb{Z}_2 \cong (G_1 \times G_3) \rtimes_\varphi \mathbb{Z}_2 \times G_2 \cong G_1 \wr \mathbb{Z}_2 \times G_2.$$

An alternative proof shows that the automorphism groups of trees are closed under (d). Suppose that $G_1, G_2, G_3 \in \mathrm{Aut}(\mathsf{TREE})$ and $G_1 \cong G_3$. Then there exist trees T_1, T_2 and T_3 such that $\mathrm{Aut}(T_i) \cong G_i$ and $T_1 \cong T_3$. We construct a tree T by attaching T_1, T_2, and T_3 to a path by the roots, as shown in Fig. 4b. ◀

From Interval Graphs to Trees. We solve the open problem of Hanlon [17]. For an interval graph X, we construct a tree T such that $\mathrm{Aut}(X) \cong \mathrm{Aut}(T)$. Consider the MPQ-tree M for X. We know that $\mathrm{Aut}(M) \cong \mathrm{Aut}(X)$ and we just need to encode the structure of M into T. We do this inductively.

Suppose a P-node P is in the root. Then its subtrees can be encoded by trees and we just attach them to a common root. Further, if $\mathrm{sec}(P)$ is non-empty, we attach a star with $|\mathrm{sec}(P)|$ leaves to the root. As before, we possibly need to modify this by subdivision, and we get $\mathrm{Aut}(T) \cong \mathrm{Aut}(M)$.

Let a Q-node Q be in the root. If Q is asymmetric, we attach the trees corresponding to the subtrees of Q and stars corresponding to the vertices of equal sections of Q to an asymmetric path. If Q is symmetric, then $\mathrm{Aut}(M) \cong (G_1 \times G_2 \times G_3) \rtimes \mathbb{Z}_2$ and we just attach trees T_1, T_2 and T_3 to a path as in Fig. 4b. In both cases, $\mathrm{Aut}(T) \cong \mathrm{Aut}(M)$.

From Trees to Interval Graphs. For a rooted tree T, we construct an interval graph X such that $\mathrm{Aut}(T) \cong \mathrm{Aut}(X)$ as follows. We place the intervals by copying the structure of T, as shown in Fig. 5. Each interval is contained exactly in the intervals of its ancestors. If T contains a vertex with only one child, then $\mathrm{Aut}(T) < \mathrm{Aut}(X)$. This can be fixed by adding asymmetric paths, as in Fig. 5.

Automorphism Groups of Proper Interval Graphs. As an application of the previously derived characterization of Aut(INT), we show that the automorphism groups of connected proper interval graphs are the same as the automorphism groups of caterpillars. First, we derive a characterization of Aut(CATERPILLAR).

▶ **Lemma 5.** *Let X be a caterpillar graph and let P be the central path.*

(i) *If no automorphism swaps the path P, then the group* $\mathrm{Aut}(X)$ *is isomorphic to a direct product of symmetric groups.*

(ii) *If there exists an automorphism of X that swaps the path P, then*

$$\mathrm{Aut}(X) \cong (G_1 \times G_2 \times G_3) \rtimes_\varphi \mathbb{Z}_2,$$

where G_2 is isomorphic to $\mathbb{S}_k$, $G_1 \cong G_3$ are isomorphic to a direct product of symmetric groups, and φ is the homomorphism defined as $\varphi(0) = \mathrm{id}$ and $\varphi(1) = (g_1, g_2, g_3) \mapsto (g_3, g_2, g_1)$.

Proof (Sketch). The root of an MPQ-tree M representing a caterpillar graph X is a Q-node. All twin classes are trivial, since X is a tree. Each child of the root is either a P-node, or a leaf. All children of every P-node are leaves. If there exist an automorphism that swaps the central path P, then the root is symmetric, otherwise it is asymmetric. We can determine $\mathrm{Aut}(M)$ similarly as in the proof of Lemma 4. ◀

Proof of Theorem 1(ii). According to Corneil [5], the MPQ-tree representing a connected proper interval graph contains only one Q-node with the maximal cliques attached to it. It is possible that the sections of this Q-node are nontrivial. This equality of automorphism groups follows by Lemma 5 and the proof of Lemma 4. ◀

3 Automorphism Groups of Circle Graphs

In this section, we prove Theorem 1(iii). We start by introducing *split decomposition* (used for recognizing circle graphs) which is described by a *split-tree*. Similarly as in Section 2, we show for a split-tree S that $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$. From now on, we focus on connected circle graphs and we want to establish that their automorphism groups are the same as the automorphism groups of pseudotrees (PSEUDOTREE).

Split Decomposition. A *split* of X is a partition of the set $V(X)$ into four parts A, B, A' and B' such that:

- For every $a \in A$ and every $b \in B$, we have $ab \in E(X)$.
- There is no edge between A' and $B \cup B'$, and between B' and $A \cup A'$.
- Both sides have at least two vertices: $|A \cup A'| \geq 2$ and $|B \cup B'| \geq 2$.

The split decomposition takes *any* split of X, and replaces X by graphs X_A and X_B. The graph X_A is induced by $A \cup A' \cup \{m_A\}$, where m_A is a *marker vertex* adjacent exactly to the vertices in A. The graph X_B is defined similarly for B, B' and m_B; see Fig. 6a. The decomposition is then applied recursively on X_A and X_B. Graphs containing no splits are called *prime graphs*. According to [11], every prime circle graph has a unique circle representation up to rotations and reflections. It is standard to stop the split decomposition also on *degenerate graphs* which are K_n and $K_{1,n}$ (which clearly are circle graphs). The reason is that these graphs have many splits but are very simple. The fundamental property is that a graph X is a circle graph if and only if X_A and X_B are circle graphs.

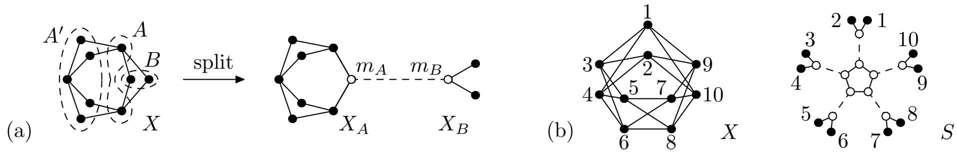


Figure 6 (a) An example of a split. The marker vertices are depicted in white. (b) The split-tree S with dashed tree-edges. We have $\mathrm{Aut}(S) \cong \mathbb{Z}_2^5 \rtimes \mathbb{D}_5$.

Split-tree. We encode the steps of the split decomposition by a tree structure. If X contains a split (A, B, A', B'), then we replace X by the graphs X_A and X_B, and connect the marker vertices m_A and m_B by a *tree-edge*. We repeat this recursively on X_A and X_B. The resulting graph is called a *split-tree*, since tree-edges connect prime and degenerate graphs in a tree pattern; see Fig. 6b. Each prime or degenerate graph is a *node* of the split-tree.

In [12], split-trees are defined in terms of graph-labeled trees. However, our definition is more suitable for working with automorphism groups. Cunningham [6] proved that the split-tree S for a graph X is uniquely determined. Clearly, a graph is a circle graph if and only if each node of its split-tree is a circle graph. The following lemma says that the split-tree S captures the adjacencies in X; we omit the proof.

▶ **Lemma 6.** *The vertices $x, y \in V(X)$ are adjacent if and only if there exists an alternating path $x, m_1, m_2, \dots, m_k, y$ in the split-tree S such that each m_i is a marker vertex, each $m_{2i-1}m_{2i}$ is a tree-edge and the remaining edges belong to $E(X)$.*

Automorphisms of a Split-tree. The split-tree S is a labeled graph where some vertices are labeled as marker vertices and some edges are labeled as tree-edges. An automorphism of S is required to preserve these labels, so it maps marker vertices only to marker vertices and tree-edges only to tree-edges. We show that the automorphism group of S is isomorphic to $\mathrm{Aut}(X)$.

▶ **Lemma 7.** *Let S be a split-tree representing X. Then $\mathrm{Aut}(S) \cong \mathrm{Aut}(X)$.*

Proof. First, we show that each $\sigma \in \mathrm{Aut}(S)$ induces a unique automorphism α of X. We define $\alpha = \sigma \restriction_{V(X)}$. By Lemma 6, two vertices $x, y \in V(X)$ are adjacent if and only if there exists an alternating path in S connecting them. Since σ is an automorphism, the existence of this alternating path is preserved between x and y and between $\sigma(x)$ and $\sigma(y)$. Therefore $xy \in E(X) \iff \alpha(x)\alpha(y) \in E(X)$.

For the converse, we show that $\alpha \in \mathrm{Aut}(X)$ induces a unique automorphism $\sigma \in \mathrm{Aut}(S)$. On the non-marker vertices, σ is determined. On the marker vertices, we define σ recursively. Let (A, B, A', B') be a split in X. This split is mapped by α to another split (C, D, C', D'), i.e., $\alpha(A) = C$, $\alpha(A') = C'$, $\alpha(B) = D$, and $\alpha(B') = D'$. By applying the split decomposition to the first split, we get the graphs X_A and X_B with the marker vertices $m_A \in V(X_A)$ and $m_B \in V(X_B)$. Similarly, for the second split we get X_C, X_D with $m_C \in V(X_C)$ and $m_D \in V(X_D)$. Since α is an automorphism, we have that $X_A \cong X_C$ and $X_B \cong X_D$. It follows that the unique split-trees of X_A and X_C are isomorphic, and similarly for X_B and X_D. Therefore, we define $\sigma(m_A) = m_C$ and $\sigma(m_B) = m_D$, and we finish the rest recursively. ◀

▶ **Lemma 8.** *A connected circle graph X has* $\mathrm{Aut}(X) \in \mathrm{Aut}(\mathsf{PSEUDOTREE})$.

Proof (Sketch). We begin by proving the following characterization:

$$\mathrm{Aut}(\mathsf{PSEUDOTREE}) = \bigcup_{n \geq 1} \mathrm{Aut}(\mathsf{TREE}) \rtimes \mathbb{D}_n \cup \mathrm{Aut}(\mathsf{TREE}) \rtimes \mathbb{Z}_n.$$

Suppose a pseudotree Y contains a cycle, otherwise $\mathrm{Aut}(Y) \in \mathrm{Aut}(\mathsf{TREE}) \rtimes \mathbb{Z}_1$. Then $\mathrm{Aut}(Y)$ preserves the cycle. The subgroup of $\mathrm{Aut}(Y)$ fixing the cycle belongs to $\mathrm{Aut}(\mathsf{FOREST}) = \mathrm{Aut}(\mathsf{TREE})$, and $\mathrm{Aut}(Y)$ acts on the cycle as a dihedral or cyclic group. This can be described by a semidirect product, and so $\mathrm{Aut}(Y) \in \mathrm{Aut}(\mathsf{TREE}) \rtimes \mathbb{D}_n$ or $\mathrm{Aut}(\mathsf{TREE}) \rtimes \mathbb{Z}_n$.

Let X be a connected circle graph and S a split-tree for X. By Lemma 7 we have that $\mathrm{Aut}(X) \cong \mathrm{Aut}(S)$. Since X is a circle graph, each node of S is a prime or degenerate graph. The automorphism group of a degenerate graph is isomorphic to $\mathbb{S}_n$. According to [11], each circle graph that is prime has a unique circle representation, up to rotations and reflections. It follows that the automorphism group of a prime circle graph is a subgroup of $\mathbb{D}_n$.

The split tree S consists of prime and degenerate graphs connected by tree-edges. The center of the split-tree is a node or a tree-edge. In the latter case, we subdivide the tree-edge by creating two new marker vertices and connecting them by a normal edge. So, we assume that the center is a node C. Every automorphism of S maps C to C.

We root S by C. Let $N \neq C$ be a node of S. If N is a degenerate graph, then we can arbitrarily permute its isomorphic children. If N is a prime graph, then we can only reverse the order of its children. This is because the vertex of N which is connected by a tree-edge with the parent of N has to be fixed. The subgroup of $\mathrm{Aut}(S)$ that fixes C is in $\mathrm{Aut}(\mathsf{FOREST}) = \mathrm{Aut}(\mathsf{TREE})$, similarly as for interval graphs.

If the center C is a degenerate graph, then $\mathrm{Aut}(S) \in \mathrm{Aut}(\mathsf{TREE})$ since it closed under (b) and (c) of Lemma 4. Otherwise, C is a prime graph and $\mathrm{Aut}(S)$ acts on C as a subgroup of a dihedral group. Therefore, $\mathrm{Aut}(S) \in \mathrm{Aut}(\mathsf{PSEUDOTREE})$. ◀

The above lemma geometrically describes automorphisms of circle graphs. The center C corresponds to the essential geometrical structure of X, and it can be rotated and possibly reflected. The remainder of X is attached to C via the structure of S, so it is less free. We note that the automorphism groups $\mathrm{Aut}(\mathsf{PSEUDOFOREST})$ can be constructed from $\mathrm{Aut}(\mathsf{PSEUDOTREE})$ by Theorem 3.

We are ready to prove that $\mathrm{Aut}(\mathsf{CIRCLE}) = \mathrm{Aut}(\mathsf{PSEUDOFOREST})$:

Proof of Theorem 1(iii). Each connected circle graph X has $\mathrm{Aut}(X) \in \mathrm{Aut}(\mathsf{PSEUDOTREE})$ according to Lemma 8. Since every pseudotree is a connected circle graph, these two classes have the same automorphism groups. Circle graphs and pseudotrees are closed under disjoint unions, hence the equality follows. ◀

4 Automorphism Groups Acting on Intersection Representations

We denote by $\mathfrak{Rep}$ the set of all intersection representations of a graph X. Every automorphism $\pi \in \mathrm{Aut}(X)$ creates from $\mathcal{R} \in \mathfrak{Rep}$ another representation $\mathcal{R}'$ such that $R'_{\pi(u)} = R_u$; so π swaps the labels of the sets of $\mathcal{R}$. We denote $\mathcal{R}'$ as $\pi(\mathcal{R})$, and $\mathrm{Aut}(X)$ acts on $\mathfrak{Rep}$.

The general set $\mathfrak{Rep}$ is too large. Therefore it is more convenient to define a suitable equivalence relation $\sim$. We factorize $\mathfrak{Rep}$ by $\sim$ and we work with $\mathfrak{Rep}/\sim$, which contains exactly one representation from every equivalence class. It is reasonable to assume that $\sim$ is a congruence with respect to the action of $\mathrm{Aut}(X)$, which means that for every $\mathcal{R} \sim \mathcal{R}'$ and $\pi \in \mathrm{Aut}(X)$, we have $\pi(\mathcal{R}) \sim \pi(\mathcal{R}')$. We consider the induced action of $\mathrm{Aut}(X)$ on $\mathfrak{Rep}/\sim$.

We assume that stabilizer of $\mathcal{R} \in \mathfrak{Rep}/\sim$ is a normal subgroup $\mathrm{Aut}(\mathcal{R})$ of $\mathrm{Aut}(X)$ which describes automorphisms inside this representation. The quotient $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R})$ describes all morphisms which change one representation in the orbit of $\mathcal{R}$ into another one. Our strategy for understanding $\mathrm{Aut}(X)$ is by decomposing it geometrically into $\mathrm{Aut}(\mathcal{R})$,

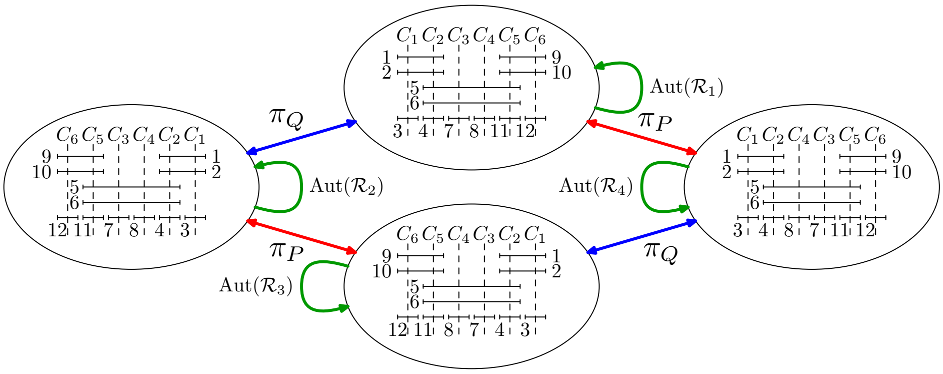


Figure 7 An interval graph with four non-equivalent representations. Its MPQ-tree M, depicted in Fig. 3, has one Q-node and one P-node. The graph has three classes of twin vertices of size two, so $\mathrm{Aut}(\mathcal{R}) \cong \mathbb{S}_2^3$. The quotient group $\mathrm{Aut}(T)$ is generated by two automorphism: π_Q corresponding to flipping the Q-node, and π_P corresponding to permuting the P-node. We have $\mathrm{Aut}(T) \cong \mathbb{Z}_2^2$.

which is mostly very simple, and $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R})$, for which we need to understand the structure of all representations.

This approach is inspired by well-known results in map theory. A *map* $\mathcal{M}$ is a 2-cell embedding of a graph; i.e, aside vertices and edges, it prescribes a rotation scheme for the edges incident with each vertex. One defines $\mathrm{Aut}(\mathcal{M})$ as the subgroup of $\mathrm{Aut}(X)$ which preserves/reflects the rotational schemes. Unlike $\mathrm{Aut}(X)$, we know that $\mathrm{Aut}(\mathcal{M})$ is always small and can be easily determined in polynomial time. But the quotient $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{M})$ describes morphisms between different maps and can be very complicated.

Interval Graphs. For an interval graph X, the set $\mathfrak{Rep}$ consists of all assignments of closed intervals which define X. It is natural to consider two interval representations equivalent if one can be transformed into the other by continuous shifting of the endpoints of the intervals while preserving the correctness of the representation. Then each representation of $\mathfrak{Rep}/\sim$ corresponds to a different ordering of the maximal cliques from left to right. Figure 7 depicts an interval graph with four different non-equivalent representations in $\mathfrak{Rep}/\sim$.

We interpret our results of Section 2 in terms of the action of $\mathrm{Aut}(X)$ on $\mathfrak{Rep}$. We proved that $\mathrm{Aut}(X) \cong \mathrm{Aut}(M)$ where M is the MPQ-tree. If an automorphism is in the stabilizer, then it fixes the ordering of the maximal cliques and it can only permute twin vertices. Therefore $\mathrm{Aut}(\mathcal{R})$ is a product of symmetric groups, one for each equivalence class of twin vertices. In the description using MPQ-trees, each equivalence class corresponds to a set of vertices which are contained in the same sections. Every stabilizer is the same and every orbit of the action of $\mathrm{Aut}(X)$ is isomorphic. Different orderings of the maximal cliques correspond to different reorderings of the PQ-tree. The defined $\mathrm{Aut}(T)$ describes morphisms of representations belonging to one orbit of the action of $\mathrm{Aut}(X)$, so these representations are the same up to the labeling of the intervals. It is the quotient group $\mathrm{Aut}(M)/\mathrm{Aut}(\mathcal{R})$ which is isomorphic to $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R})$.

Circle Graphs. For a circle graph X, the set $\mathfrak{Rep}$ consists of all assignments of chords of a circle which define X. Two representations are considered equivalent if one can be transformed into other by (i) continuos shifting of chords while preserving the representation and (ii) swapping two chords with the same neighbors such that there is no other endpoint in between them. We call two vertices x and y *semi-twin vertices* if $N(x) = N(y)$. They again form equivalence classes, and two representation are equivalent if they have the same circular ordering of chords up to permuting semi-twin vertices.

We interpret the results of Section 3 in terms of the action of $\mathrm{Aut}(X)$. It follows that $\mathrm{Aut}(\mathcal{R})$ is a direct product of symmetric groups, corresponding to permuting semi-twin vertices. It consists of all automorphisms which fix marker vertices of the split tree S. The quotient $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R})$ describes all structural transformations of the split tree. For the central node C, rotation/reflection is possible, so we get a subgroup of $\mathbb{D}_n$. If $N \neq C$ is a prime graph, we can only apply the geometric reflection with the axis perpendicular to the chord of the marker vertex, so their symmetries are trivial or $\mathbb{Z}_2$. For a degenerate graph $N \neq C$, one can arbitrary permute isomorphic subtrees, so it is a direct product of wreath products with symmetric groups. So $\mathrm{Aut}(X)/\mathrm{Aut}(\mathcal{R}) \in \mathrm{Aut}(\mathsf{PSEUDOTREE})$.

5 Algorithms for Computing Automorphism Groups

We have described the structure of automorphism groups of interval and circle graphs. In this section, we briefly explain algorithmic implications of our results which allow to compute automorphism groups in terms of basic groups $\mathbb{Z}_n$, $\mathbb{D}_n$ and $\mathbb{S}_n$, and their group products. This description is much better than just outputting permutations generating $\mathrm{Aut}(X)$. Many tools of the computational group theory are devoted to getting better understanding of an unknown group, described by generators (permutations, matrices) or relators (presentations). Our description gives this structural understanding of $\mathrm{Aut}(X)$ for free.

For interval graphs, a linear-time algorithm follows from the standard tools and techniques. The MPQ-tree M is computed in time $\mathcal{O}(n+m)$. We can compute $\mathrm{Aut}(T)$ in a similar manner as the automorphism group of a rooted tree. Therefore, we get a recursive description in terms of group products, and we can describe their generators.

For circle graphs, our description easily leads to a polynomial-time algorithm, by computing the split tree and understanding its symmetries. The best algorithm for computing split-trees runs in almost linear time [13]. With a careful implementation and checking all details, one can likely match this time for computing $\mathrm{Aut}(X)$ using our results.

6 Open Problems

We conclude this paper with several open problems concerning automorphism groups of other intersection-defined classes of graphs; for an overview see [15, 31].

We do not describe $\mathrm{Aut}(\mathsf{PERM})$. But our results and the inclusions $\mathsf{CATERPILLAR} \subsetneq \mathsf{PERM} \subsetneq \mathsf{CIRCLE}$ imply that they are non-universal, between $\mathrm{Aut}(\mathsf{CATERPILLAR})$ and $\mathrm{Aut}(\mathsf{CIRCLE})$. We believe that our techniques can be applied.

▶ Problem 1. What is $\mathrm{Aut}(\mathsf{PERM})$?

Circular-arc graphs (CIRCULAR-ARC) are intersection graphs of circular arcs and they naturally generalize interval graphs. Surprisingly, this class is very complex and more different from interval graphs than it seems. The paper of Hsu [20] relates circular-arc graphs to circle graphs. It easily follows that $\mathrm{Aut}(\mathsf{CIRCULAR\text{-}ARC}) \supseteq \mathrm{Aut}(\mathsf{PSEUDOTREE})$.

▶ Problem 2. What is $\mathrm{Aut}(\mathsf{CIRCULAR\text{-}ARC})$? Is it equal to $\mathrm{Aut}(\mathsf{PSEUDOTREE})$?

Figure 2 depicts two infinite hierarchies of graph classes, one between INT and CHOR, and the other one between PERM and FUN. In both cases, the bottom graph class has non-universal automorphism groups and the top one has universal automorphism groups.

Let Y be any fixed graph. The class Y-GRAPH consists of all intersections graphs of connected subgraphs of a subdivision of Y. Observe that K_2-GRAPH = INT and

$$\bigcup_{T \in \mathsf{TREE}} T\text{-}\mathsf{GRAPH} = \mathsf{CHOR}.$$

The infinite hierarchy between INT and CHOR is formed by T-GRAPH for which INT $\subseteq$ T-GRAPH $\subsetneq$ CHOR. If Y contains a cycle, then Y-GRAPH is no longer contained in CHOR. The simplest of these classes are circular-arc graphs which are equal to K_3-GRAPH.

▶ Conjecture 1. For every fixed graph Y, the class Y-GRAPH is non-universal.

The hierarchy between PERM and FUN is defined using the Dushnik-Miller dimension of partially ordered sets. Every poset is equal to the intersection of some linear orderings, and this dimension is the least number of these linear orderings. The complement of every function graph can be transitively oriented, and its dimension is the least dimension of all its transitive orientations. We denote the class of all function graphs of the dimension at most k by k-DIM. It follows that 1-DIM are all complete graphs, 2-DIM = PERM, and

$$\bigcup_{k \in \mathbb{N}} k\text{-}\mathsf{DIM} = \mathsf{FUN}.$$

We note that recognition of k-DIM is NP-complete for $k > 2$ [33].

▶ Problem 3. What are Aut(k-DIM)? Are they non-universal for every $k \in \mathbb{N}$?

References

1 L. Babai. Automorphism groups of planar graphs II. In *Infinite and finite sets (Proc. Conf. Keszthely, Hungary)*, 1973.
2 K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and planarity using PQ-tree algorithms. *J. Comput. System Sci.*, 13:335–379, 1976.
3 N. Carter. *Visual group theory.* MAA, 2009.
4 C. J. Colbourn and K. S. Booth. Linear times automorphism algorithms for trees, interval graphs, and planar graphs. *SIAM J. Comput.*, 10(1):203–225, 1981.
5 Derek G Corneil, Hiryoung Kim, Sridhar Natarajan, Stephan Olariu, and Alan P Sprague. Simple linear time recognition of unit interval graphs. *Information Processing Letters*, 55(2):99–104, 1995.
6 W.H. Cunningham. Decomposition of directed graphs. *SIAM Journal on Algebraic Discrete Methods*, 3:214–228, 1982.
7 P. Erdös, S. Fajtlowicz, and A. J. Hoffman. Maximum degree in graphs of diameter 2. *Networks*, 10(1):87–90, 1980.
8 J. Fiala, P. Klavík, J. Kratochvíl, and R. Nedela. Algorithmic aspects of regular graphs covers with applications to planar graphs. In *Lecture Notes in Computer Science, Automata, Languages, and Programming ICALP 2014*, volume 8572, pages 489–501, 2014.
9 R. Frucht. Herstellung von graphen mit vorgegebener abstrakter gruppe. *Compositio Mathematica*, 6:239–250, 1939.
10 D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pac. J. Math.*, 15:835–855, 1965.
11 C. P. Gabor, K. J. Supowit, and W.-L. Hsu. Recognizing circle graphs in polynomial time. *Journal of the ACM (JACM)*, 36(3):435–473, 1989.
12 E. Gioan and C. Paul. Split decomposition and graph-labelled trees: Characterizations and fully dynamic algorithms for totally decomposable graphs. *Discrete Appl. Math.*, 160(6):708–733, 2012.

13 E. Gioan, C. Paul, M. Tedder, and D. Corneil. Practical and efficient circle graph recognition. *Algorithmica*, pages 1–30, 2013.
14 C. D. Godsil and G. Royle. *Algebraic graph theory*, volume 207. Springer New York, 2001.
15 M. C. Golumbic. *Algorithmic graph theory and perfect graphs*, volume 57. Elsevier, 2004.
16 M. Grohe and D. Marx. Structure theorem and isomorphism test for graphs with excluded topological subgraphs. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 173–192, 2012.
17 P. Hanlon. Counting interval graphs. *Transactions of the American Mathematical Society*, 272(2):383–426, 1982.
18 Z. Hedrlín, A. Pultr, et al. On full embeddings of categories of algebras. *Illinois Journal of Mathematics*, 10(3):392–406, 1966.
19 A. J. Hoffman and R. R. Singleton. On moore graphs with diameters 2 and 3. *IBM Journal of Research and Development*, 4(5):497–504, 1960.
20 W. L. Hsu. $\mathcal{O}(M \cdot N)$ algorithms for the recognition and isomorphism problems on circular-arc graphs. *SIAM Journal on Computing*, 24(3):411–439, 1995.
21 C. Jordan. Sur les assemblages de lignes. *Journal für die reine und angewandte Mathematik*, 70:185–190, 1869.
22 N. Korte and R. H. Möhring. An incremental linear-time algorithm for recognizing interval graphs. *SIAM J. Comput.*, 18(1):68–81, 1989.
23 G. S. Lueker and K. S. Booth. A linear time algorithm for deciding interval graph isomorphism. *Journal of the ACM (JACM)*, 26(2):183–195, 1979.
24 E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25(1):42–65, 1982.
25 R. Mathon. A note on the graph isomorphism counting problem. *Information Processing Letters*, 8(3):131–136, 1979.
26 B. D. McKay, M. Miller, and J. Širáň. A note on large graphs of diameter two and given maximum degree. *J. Combin. Theory Ser. B*, 74(1):110–118, 1998.
27 M. Miller and J. Širáň. Moore graphs and beyond: A survey of the degree/diameter problem. *Electronic Journal of Combinatorics*, 61:1–63, 2005.
28 Fred S Roberts. Indifference graphs. *Proof techniques in graph theory*, 139:146, 1969.
29 J. J. Rotman. *An introduction to the theory of groups*, volume 148. Springer, 1995.
30 U. Schöning. Graph isomorphism is in the low hierarchy. *Journal of Computer and System Sciences*, 37(3):312–323, 1988.
31 J. P. Spinrad. *Efficient Graph Representations.: The Fields Institute for Research in Mathematical Sciences.*, volume 19. American Mathematical Soc., 2003.
32 H. Whitney. Nonseparable and planar graphs. *Trans. Amer. Math. Soc.*, 34:339–362, 1932.
33 Mihalis Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic Discrete Methods*, 3(3):351–358, 1982.
34 S. Zhou. A class of arc-transitive Cayley graphs as models for interconnection networks. *SIAM Journal on Discrete Mathematics*, 23(2):694–714, 2009.