Simultaneous Representation of Proper and Unit Interval Graphs

Ignaz Rutter
Faculty of Computer Science and Mathematics, University of Passau, Germany
rutter@fim.uni-passau.de

Darren Strash
Department of Computer Science, Hamilton College, Clinton, NY, USA
dstrash@hamilton.edu

Peter Stumpf
Faculty of Computer Science and Mathematics, University of Passau, Germany
stumpf@fim.uni-passau.de

Michael Vollmer
Department of Informatics, Karlsruhe Institute of Technology (KIT), Germany
michael.vollmer@kit.edu

Abstract

In a confluence of combinatorics and geometry, simultaneous representations provide a way to realize combinatorial objects that share common structure. A standard case in the study of simultaneous representations is the *sunflower case* where all objects share the same common structure. While the recognition problem for general simultaneous interval graphs – the simultaneous version of arguably one of the most well-studied graph classes – is NP-complete, the complexity of the sunflower case for three or more simultaneous interval graphs is currently open. In this work we settle this question for *proper* interval graphs. We give an algorithm to recognize simultaneous proper interval graphs in linear time in the sunflower case where we allow any number of simultaneous graphs. Simultaneous *unit* interval graphs are much more "rigid" and therefore have less freedom in their representation. We show they can be recognized in time $\mathcal{O}(|V| \cdot |E|)$ for any number of simultaneous graphs in the sunflower case where $G = (V, E)$ is the union of the simultaneous graphs. We further show that both recognition problems are in general NP-complete if the number of simultaneous graphs is not fixed. The restriction to the sunflower case is in this sense necessary.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Intersection Graphs, Recognition Algorithm, Proper/Unit Interval Graphs, Simultaneous Representations

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.80

Related Version A full version of the paper is available at `https://arxiv.org/abs/1908.08882`.

Funding This work was partially supported by grant RU 1903/3-1 of the German Research Foundation (DFG) and by the DFG Research Training Group 2153: "Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation".

1 Introduction

Given a family of sets $\mathcal{R}$, the corresponding *intersection graph* G has a vertex for each set and two vertices are adjacent if and only if their sets have a non-empty intersection. If all sets are intervals on the real line, then $\mathcal{R}$ is an *interval representation* of G and G is an *interval graph*; see Figure 1.

In the context of intersection graph classes, much work has been devoted to efficiently computing a *representation*, which is a collection of sets or geometric objects having an intersection graph that is isomorphic to a given graph. For many well-known graph classes, such as interval graphs and chordal graphs, this is a straightforward task [14, 28]. However,

27th Annual European Symposium on Algorithms (ESA 2019).
Editors: Michael A. Bender, Ola Svensson, and Grzegorz Herman; Article No. 80; pp. 80:1–80:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

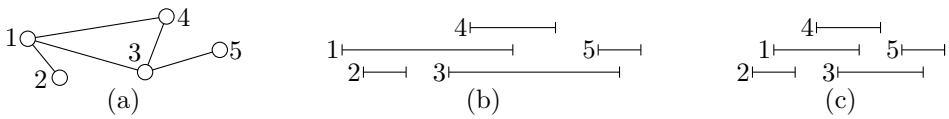


Figure 1 (a) A graph, with (b) an interval representation and (c) proper interval representation.

often it is desirable to consistently represent *multiple* graphs that have subgraphs in common. This is true, for instance, in realizing schedules with shared events, embedding circuit graphs of adjacent layers on a computer chip, and visualizing the temporal relationship of graphs that share a common subgraph [19]. Likewise, in genome reconstruction, we can ask if a sequence of DNA can be reconstructed from strands that have sequences in common [13].

Simultaneous representations capture this in a very natural way. Given *simultaneous graphs* $G_1, G_2, \ldots, G_k$ where each pair of graphs G_i, G_j share some common subgraph, a *simultaneous representation* asks for a fixed representation of each vertex that gives a valid representation of each G_i. This notion is closely related to *partial representation extension*, which asks if a given (fixed) representation of a subgraph can be extended to a representation of the full graph. Partial representation extension has been extensively studied for graph classes such as interval graphs [20], circle graphs [8], as well as proper and unit interval graphs [20]. For interval graphs, Bläsius and Rutter [3] have even shown that the partial interval representation problem can be reduced to a simultaneous interval representation problem on two graphs in linear time.

Simultaneous representations were first studied in the context of embedding graphs [2, 7], where the goal is to embed each simultaneous graph without edge crossings while shared subgraphs have the same induced embedding. Unsurprisingly, many variants are NP-complete [12, 26, 1, 11]. The notion of simultaneous representation of general intersection graph classes was introduced by Jampani and Lubiw [19], who showed that it is possible to recognize simultaneous chordal graphs with two graphs in polynomial time, and further gave a polynomial time algorithm to recognize simultaneous comparability graphs and permutation graphs with two or more graphs that share the same subgraph (the *sunflower case*). They further showed that recognizing three or more simultaneous chordal graphs is NP-complete.

Golumbic et al. [15] introduced the *graph sandwich problem* for a graph class Π. Given a vertex set V and edge sets $E_1 \subseteq E_2 \subseteq \binom{V}{2}$ it asks whether there is an edge set $E_1 \subseteq E \subseteq E_2$ such that the *sandwich graph* $G = (V, E)$ is in Π. Jampani and Lubiw showed that if Π is an intersection graph class, then recognizing k simultaneous graphs in Π in the sunflower case is a special case of the graph sandwich problem where $(V, E_2 \setminus E_1)$ is a k-partite graph [19].

We consider simultaneous *proper* and *unit* interval graphs. An interval graph is proper if in an interval representation no interval properly contains another one (see Figure 1), and it is unit if all intervals have length one. Interestingly, while proper and unit interval graphs are the same graph class as shown by Roberts [25], simultaneous unit interval graphs differ from simultaneous proper interval graphs; see Figure 2. Unit interval graphs are intersection graphs and therefore the graph sandwich paradigm described by Jampani and Lubiw applies. Proper interval graphs are not since in a simultaneous representation intervals of distinct graphs may contain each other which means that the intersection graph of all intervals in the simultaneous representation is not proper.

Sunflower (*unit*) interval graphs are a generalization of *probe* (proper) interval graphs, where each sunflower graph has only one non-shared vertex. Both variants of probe graphs can be recognized in linear time [22, 23].

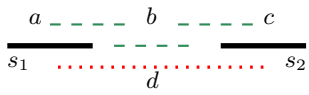


Figure 2 A simultaneous proper interval representation of a sunflower graph $\mathcal{G}$ consisting of two paths $G_1 = (s_1, a, b, c, s_2)$ (dashed) and $G_2 = (s_1, d, s_2)$ (dotted) with shared start and end s_1, s_2 (bold). They have no simultaneous unit interval representation: The intervals a and c enforce that b lies between s_1 and s_2. Interval d therefore includes b in every simultaneous proper interval representation. In particular, not both can have size one.

Simultaneous interval graphs were first studied by Jampani and Lubiw [18] who gave a $\mathcal{O}(n^2 \lg n)$-time recognition algorithm for the special case of two simultaneous graphs. Bläsius and Rutter [3] later showed how to recognize two simultaneous interval graphs in linear time. Bok and Jedličková showed that the recognition of an arbitrary number of simultaneous interval graphs is in general NP-complete [4]. However, the complexity for the sunflower case with more than two simultaneous graphs is still open.

Our Results. We settle these problems with k not fixed for simultaneous *proper* and *unit* interval graphs – those graphs with an interval representation where no interval properly contains another and where all intervals have unit length, respectively [10, 27, 9, 16]. For the sunflower case, we provide efficient recognition algorithms. The running time for proper interval graphs is linear, while for the unit case it is $\mathcal{O}(|V| \cdot |E|)$ where $G = (V, E)$ is the union of the sunflower graphs. In the full version we prove NP-completeness for the non-sunflower case. The reductions are similar to the simultaneous independent work of Bok and Jedličková for simultaneous interval graphs [4].

Organization. We begin by introducing basic notation and existing tools throughout Section 2. In Section 3 we give a characterization of simultaneous proper interval graphs, from which we develop an efficient recognition algorithm. In Section 4 we characterize simultaneous proper interval graphs that can be simultaneous unit interval graphs, and then exploit this property to efficiently search for a representation among simultaneous proper interval graph representations. Proofs of lemmas and theorems marked with $\star$ are omitted.

2 Preliminaries

In this section we give basic notation, definitions and characterizations. Section 2.1 collects basic concepts on graph theory, orderings, and PQ-trees. Section 2.2 introduces (proper) interval graphs and presents relations between the representations of such graphs and their induced subgraphs. Finally, Section 2.3 introduces the definition and notation of simultaneous graphs.

2.1 Graphs, Orderings, and PQ-trees

Unless mentioned explicitly, all graphs in this paper are undirected. For a graph $G = (V, E)$ we denote its size $|G| := |V| + |E|$.

Let σ be a binary relation. Then we write $a_1 \leq_\sigma a_2$ for $(a_1, a_2) \in \sigma$, and we write $a_1 <_\sigma a_2$ if $a_1 \leq_\sigma a_2$ and $a_1 \neq a_2$. We omit the subscript and simply use $<$ and $\leq$ if the ordering it refers to is clear from the context. We denote the *reversal* of a linear order σ by σ^r, and we use $\circ$ to concatenate linear orders of disjoint sets.

A *PQ-tree* is a data structure for representing sets of linear orderings of a ground set X. Namely, given a set $\mathcal{C} \subseteq 2^X$, a *PQ-tree on X for $\mathcal{C}$* is a tree data structure T that represents the set $\textsc{Consistent}(T)$ containing exactly the linear orders of X in which the elements of each set $C \in \mathcal{C}$ are consecutive. The PQ-tree T can be computed in time $O(|X| + \sum_{C \in \mathcal{C}} |C|)$ [6]. Given a PQ-tree T on the set X and a subset $X' \subseteq X$, there exists a PQ-tree T', called the *projection* of T to X', that represents exactly the linear orders of X' that are restrictions of orderings in $\textsc{Consistent}(T)$. For any two PQ-trees T_1 and T_2 on the set X, there exists a PQ-tree T with $\textsc{Consistent}(T) = \textsc{Consistent}(T_1) \cap \textsc{Consistent}(T_2)$, called the *intersection* of T_1 and T_2. Both the projection and the intersection can be computed in $O(|X|)$ time [5].

2.2 Interval Graphs, Proper Interval Graphs, and Their Subgraphs

An *interval representation* $R = \{I_v \mid v \in V\}$ of a graph $G = (V, E)$ associates with each vertex $v \in V$ an interval $I_v = [x, y]$ of real numbers such that for each pair of vertices $u, v \in V$ we have $I_u \cap I_v \neq \emptyset$ if and only if $\{u, v\} \in E$, i.e., the intervals intersect if and only if the corresponding vertices are adjacent. An interval representation R is *proper* if no interval properly contains another one, and it is *unit* if all intervals have length 1. A graph is an *interval graph* if and only if it admits an interval representation, and it is a *proper (unit) interval graph* if and only if it admits a proper (unit) interval representation. It is well-known that proper and unit interval graphs are the same graph class.

▶ **Proposition 1** ([25]). *A graph is a unit interval graph if and only if it is a proper interval graph.*

However, this does not hold in the simultaneous case where every simultaneous unit interval representation is clearly a simultaneous proper interval representation of the same graph, but not every simultaneous proper interval representation implies a simultaneous unit interval representation; see Figure 2.

We use the well-known characterization of proper interval graphs using *straight enumerations* [10]. Two adjacent vertices $u, v \in V$ are *indistinguishable* if we have $N[u] = N[v]$ where $N[u] = \{v \colon uv \in E(H)\} \cup \{u\}$ is the closed neighborhood. Being indistinguishable is an equivalence relation and we call the equivalence classes *blocks* of G. We denote the block of G that contains vertex u by $B(u, G)$. Note that for a subgraph $G' \subseteq G$ the block $B(u, G')$ may contain vertices in $V(G') \setminus B(u, G)$ that have the same neighborhood as u in G' but different neighbors in G. Two blocks B, B' are *adjacent* if and only if $uv \in E$ for (any) $u \in B$ and $v \in B'$. A linear order σ of the blocks of G is a *straight enumeration* of G if for every block, the block and its adjacent blocks are consecutive in σ. A proper interval representation R defines a straight enumeration $\sigma(R)$ by ordering the intervals by their starting points and grouping together the blocks. Conversely, for each straight enumeration σ, there exists a corresponding representation R with $\sigma = \sigma(R)$ [10]. A *fine enumeration* of a graph H is a linear order η of $V(H)$ such that for $u \in V(H)$ the closed neighborhood $N_H[u]$ is consecutive in η.

▶ **Proposition 2** ([24, 10, 17]). *(i) A graph is a proper interval graph if and only if it has a fine enumeration. (ii) A graph is a proper interval graph if and only if it admits a straight enumeration. (iii) A straight enumeration of a connected proper interval graph is unique up to reversal.*

2.3 Simultaneous Graphs

A *simultaneous graph* is a tuple $\mathcal{G} = (G_1, \ldots, G_k)$ of graphs G_i that may each share vertices and edges. Note that this definition differs from the one we gave in the introduction. This way the input for the simultaneous representation problem is a single entity. The size $|\mathcal{G}|$ of a simultaneous graph is $\sum_{i=1}^{k} |G_i|$. We call $\mathcal{G}$ *connected*, if $\bigcup_{i=1}^{k} G_i$ is connected. A *simultaneous (proper/unit) interval representation* $\mathcal{R} = (R_1, \ldots, R_k)$ of $\mathcal{G}$ is a tuple of representations such that $R_i \in \mathcal{R}$ is a (proper/unit) interval representation of graph G_i and the intervals representing shared vertices are identical in each representation. A simultaneous graph is a *simultaneous (proper/unit) interval graph* if it admits a simultaneous (proper/unit) interval representation.

An important special case is that of *sunflower graphs*. The simultaneous graph $\mathcal{G}$ is a *sunflower* graph if each pair of graphs G_i, G_j with $i \neq j$ shares exactly the same subgraph S, which we then call the *shared graph*. Note that, for $\mathcal{G}$ to be a simultaneous interval graph, it is a necessary condition that $G_i \cap G_j$ is an induced subgraph of G_i and G_j for $i, j = 1, \ldots, k$. In particular, in the sunflower case the shared graph S must be an induced subgraph of each G_i. The following lemma allows us to restrict ourselves to instances whose union graph $\bigcup_{\mathcal{G}} = \bigcup_{i=1}^{k} G_i$ is connected.

▶ **Lemma 3** ($\star$). *Let $\mathcal{G} = (G_1, \ldots, G_k)$ be a simultaneous graph and let $C_1, \ldots, C_l$ be the connected components of $\bigcup_{\mathcal{G}}$. Then $\mathcal{G}$ is a simultaneous (proper) interval graph if and only if each of the graphs $\mathcal{G}_i = (G_1 \cap C_i, \ldots, G_k \cap C_i)$, $i = 1, \ldots, l$ is a simultaneous (proper/unit) interval graph.*

3 Sunflower Proper Interval Graphs

In this section, we deal with simultaneous proper interval representations of sunflower graphs. We first present a combinatorial characterization of the simultaneous graphs that admit such a representation. Afterwards, we present a simple linear-time recognition algorithm. Finally, we derive a combinatorial description of all the combinatorially different simultaneous proper interval representations of a connected simultaneous graph, which is a prerequisite for the unit case.

3.1 Characterization

Let $G = (V, E)$ be a proper interval graph with straight enumeration σ and let $V_S \subseteq V$ be a subset of vertices. We call σ *compatible* with a linear order ζ of V_S if, we have for $u, v \in V_S$ that $u \leq_\zeta v$ implies $B(u, G) \leq_\sigma B(v, G)$.

▶ **Lemma 4.** *Let $\mathcal{G} = (G_1, \ldots, G_k)$ be a sunflower graph with shared graph $S = (V_S, E_S)$. Then $\mathcal{G}$ admits a simultaneous proper interval representation $\mathcal{R}$ if and only if there exists a linear order ζ of V_S and straight enumerations σ_i for each G_i that are compatible with ζ.*

Proof Sketch. For a given representation $\mathcal{R}$ the straight enumerations $\sigma_i = \sigma(R_i)$ and linear order ζ of V_S given by their left endpoints in $\mathcal{R}$ clearly satisfy the lemma. Conversely we build a linear order of interval endpoints from each σ_i that equals a proper interval representation. As each σ_i is compatible with ζ, all endpoint orderings allow the same ordering for vertices in S, thus permitting one global ordering of all endpoints. Drawing the intervals according to this ordering then yields a simultaneous representation $\mathcal{R}$ since it extends each individual ordering. ◀

Let $\mathcal{G} = (G_1, \ldots, G_k)$ be a sunflower graph with shared graph $S = (V_S, E_S)$ and for each $G_i \in \mathcal{G}$ let σ_i be a straight enumeration of G_i. We call the tuple $(\sigma_1, \ldots, \sigma_k)$ a *simultaneous enumeration* if for any $i, j \in \{1, \ldots, k\}$ and $u, v \in V_S$ we have $B(u, G_i) <_{\sigma_i} B(v, G_i) \Rightarrow B(u, G_j) \leq_{\sigma_j} B(v, G_j)$. That is, the blocks containing vertices of the shared graph are not ordered differently in any straight enumeration.

▶ **Theorem 5** ($\star$)**.** *Let $\mathcal{G} = (G_1, \ldots, G_k)$ be a sunflower graph. There exists a simultaneous proper interval representation $\mathcal{R} = (R_1, \ldots, R_k)$ of $\mathcal{G}$ if and only if there is a simultaneous enumeration $(\sigma_1, \ldots, \sigma_k)$ of $\mathcal{G}$. If $(\sigma_1, \ldots, \sigma_k)$ exists, there also exists $\mathcal{R}$ with $\sigma(R_i) = \sigma_i$ for each $R_i \in \mathcal{R}$.*

3.2 A Simple Recognition Algorithm

In this section we develop a very simple recognition algorithm for sunflower graphs that admits a simultaneous proper interval representation based on Theorem 5.

Let $\mathcal{G} = (G_1, \ldots, G_k)$ be a sunflower graph with shared graph $S = (V_S, E_S)$. By Proposition 2, for each graph G_i, there exists a PQ-tree T_i' that describes exactly the fine enumerations of G_i. We denote by $T_i = T_i'|S$ the projection of T_i to the vertices in S. The tree T_i thus describes all proper interval representations of S that can be extended to a proper interval representation of G_i. Let T denote the intersection of $T_1, \ldots, T_k$. By definition, T represents all proper interval representations of S that can be extended to a proper interval representation of each graph G_i. Thus, $\mathcal{G}$ admits a simultaneous proper interval representation if and only if T is not the null-tree.

If T is not the null-tree, we can obtain a simultaneous representation by choosing any ordering $O \in \textsc{Consistent}(T)$ and constructing a simultaneous representation $\mathcal{S}$ of S. Using the algorithm of Klavík et al. [20], we can then independently extend $\mathcal{S}$ to representations R_i of G_i. Since the trees T_i can be computed in time linear in the size of the graph G_i, and the intersection of two trees takes linear time, the testing algorithm takes time linear in the total size of the G_i. The representation extension of Klavík et al. [20] runs in linear time. We therefore have the following theorem.

▶ **Theorem 6.** *Given a sunflower graph $\mathcal{G} = (G_1, \ldots, G_k)$, it can be tested in linear time whether $\mathcal{G}$ admits a simultaneous proper interval representation.*

3.3 Combinatorial Description of Simultaneous Representations

Let $\mathcal{G}$ be a sunflower proper interval graph with shared graph S and simultaneous representation $\mathcal{R}$. Then, each representation $R \in \mathcal{R}$ uses the same intervals for vertices of S and implies the same straight enumeration $\sigma_S(\mathcal{R}) = \sigma_S(R) = \sigma(\{I_v \in R : v \in V(S)\})$.

▶ **Lemma 7.** *Let $\mathcal{G}$ be a connected sunflower proper interval graph with shared graph S. Across all simultaneous proper interval representations $\mathcal{R}'$ of $\mathcal{G}$, the straight enumeration $\sigma_S(\mathcal{R})$ of S is unique up to reversal.*

Proof. Let $\mathcal{R}$ be a simultaneous representation of $\mathcal{G}$ and $\sigma_S(\mathcal{R})$ the straight enumeration of S induced by $\mathcal{R}$. Since $\mathcal{G}$ is connected, for any two blocks B_i and B_{i+1} of S consecutive in $\sigma_S(\mathcal{R})$, there exists a graph $G \in \mathcal{G}$ such that B_i and B_{i+1} are in the same connected component of G. Since S is an induced subgraph of G, for any two vertices $u, v \in V(S)$ with $B(u, S) \neq B(v, S)$ we have $B(u, G) \neq B(v, G)$. This means that a straight enumeration of G implies a straight enumeration of S. Additionally, the straight enumeration of each connected component of G is unique up to reversal by Proposition 2. As a result, for any

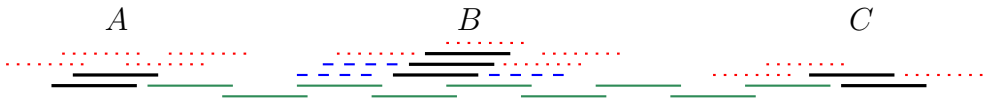


Figure 3 Simultaneous proper interval representation of G_1 (green solid), G_2 (red dotted), G_3 (blue dashed) with shared graph S (black bold). S has three blocks A, B, C. We denote the component of G_i containing a block D by C_D^i. C_A^2, C_B^2, C_B^3, C_C^2 are loose. C_A^2 is independent. (C_B^2, C_B^3) is a reversible part. (C_C^2) is not a reversible part, since C_C^1 is aligned at C and not loose.

proper interval representation R of G, the blocks B_i and B_{i+1} are consecutive in $\sigma_S(R)$. This holds for any two consecutive blocks in σ, which means that the consecutivity of all blocks of S is fixed for all simultaneous representations of $\mathcal{G}$. As a consequence $\sigma_S(\mathcal{R})$ is fixed up to complete reversal. ◀

Let G be a proper interval graph consisting of the connected components $C_1, \dots, C_k$ with straight enumerations $\sigma_1, \dots, \sigma_k$. Let $\sigma_1 \circ \dots \circ \sigma_k$ be a straight enumeration of G. Then we say the straight enumeration $\sigma' = \sigma_1 \circ \dots \circ \sigma_{i-1} \circ \sigma_i^r \circ \sigma_{i+1} \circ \dots \circ \sigma_k$ is *obtained from* σ *by reversal of* C_i. For a sunflower graph $\mathcal{G}$ containing G with shared graph $S = (V_S, E_S)$, we call a component $C = (V_C, E_C)$ of G *loose*, if all vertices $V_S \cap V_C$ are in the same block of S. Reversal of loose components is the only "degree of freedom" among simultaneous enumerations, besides full reversal, and is formally shown in the full version.

To obtain a complete characterization, we now introduce additional terms to specify which reversals result in simultaneous enumerations (see Figure 3). Let $\mathcal{G} = (G_1, \dots, G_k)$ be a connected sunflower proper interval graph with shared graph S. We say a component C of a graph in $\mathcal{G}$ *aligns* two vertices $u, v \in S$ if they are in different blocks of C, i.e., $B(u, C) \neq B(v, C)$. If in addition u and v are in the same block B of S, we say C *is oriented at* B. If there is another component C' among graphs in $\mathcal{G}$ oriented at B, the orientation of their straight enumerations in a simultaneous enumeration of $\mathcal{G}$ are dependent; that is, they cannot be reversed independently. This is shown formally in the full version.

For each block B of S, let $\mathcal{C}(B)$ be the connected components among graphs in $\mathcal{G}$ oriented at B. Since a component may contain B without aligning vertices, we have $0 \leq |\mathcal{C}(B)| \leq k$. If $\mathcal{C}(B)$ contains only loose components, we call it a *reversible part*. Note that a reversible part $\mathcal{C}(B)$ contains at most one component of each graph G_i. Additionally, we call a loose component C *independent*, if it does not align any two vertices of S. Let $(\sigma_1, \dots, \sigma_k)$ and $(\sigma'_1, \dots, \sigma'_k)$ be tuples of straight enumerations of $G_1, \dots, G_k$. We say $(\sigma'_1, \dots, \sigma'_k)$ is *obtained from* $(\sigma_1, \dots, \sigma_k)$ *through reversal* of reversible part $\mathcal{C}(B)$, if $\sigma'_1, \dots, \sigma'_k$ are obtained by reversal of all components in $\mathcal{C}(B)$.

▶ **Theorem 8** ($\star$). *Let $\mathcal{G} = (G_1, \dots, G_k)$ be a connected sunflower graph with shared graph S and simultaneous enumeration $\rho = (\sigma_1, \dots, \sigma_k)$. Then $\rho' = (\sigma'_1, \dots, \sigma'_k)$ is a simultaneous enumeration of $\mathcal{G}$ if and only if ρ' can be obtained from ρ or ρ^r through reversal of independent components and reversible parts.*

4 Sunflower Unit Interval Graphs

In the previous section we characterized all simultaneous enumerations for a sunflower proper interval graph $\mathcal{G}$. We say a simultaneous proper/unit interval representation of a sunflower graph $\mathcal{G}$ *realizes* a simultaneous enumeration $\zeta = (\zeta_1, \dots, \zeta_k)$ of ζ, if for $i \in \{1, \dots, k\}$ the representation of G_i corresponds to the straight enumeration ζ_i. In Section 4.1 we provide a criterion which determines for a given simultaneous enumeration ζ of $\mathcal{G}$ whether there is a

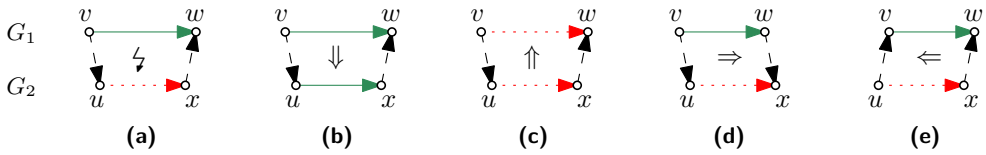


Figure 4 (a): The forbidden configuration of Corollary 11. (b)–(e): The four implications of Corollary 12.

simultaneous unit interval representation of $\mathcal{G}$ that realizes ζ. Namely, the criterion is the avoidance of a certain configuration in a partial vertex order of $\bigcup_{\mathcal{G}}$ *induced by* ζ. In Section 4.2 we combine these findings to efficiently recognize simultaneous unit interval graphs.

4.1 Simultaneous Enumerations of Sunflower Unit Interval Graphs

We first obtain a combinatorial characterization by reformulating the problem of finding a representation as a restricted graph sandwich problem [15].

▶ **Lemma 9** ($\star$). *A sunflower graph $\mathcal{G}$ has a simultaneous unit interval representation that realizes a simultaneous enumeration $\zeta = (\zeta_1, \dots, \zeta_k)$ if and only if there is some graph H with $V(H) = V(\mathcal{G})$ that contains the graphs $G_1, \dots, G_k$ as induced subgraphs and has a fine enumeration σ such that for $i \in \{1, \dots, k\}$ straight enumeration ζ_i is compatible with σ on V_i.*

Our approach is to obtain more information on what graph H and the fine enumeration σ must look like. We adapt a characterization of Looges and Olariu [21] to obtain four implications that can be used given only partial information on H and σ (as given by Lemma 9); see Figure 4. For the figures in this section we use arrows to represent a partial order between two vertices. We draw them solid green if they are adjacent, red dotted if they are non-adjacent in some graph G_i, and black dashed if they may or may not be adjacent.

▶ **Theorem 10** (Looges and Olariu [21]). *A vertex order of a graph $H = (V, E)$ is a fine enumeration if and only if for $v, u, w \in V$ with $v <_\sigma u <_\sigma w$ and $vw \in E$ we have $vu, uw \in E$.*

▶ **Corollary 11** ($\star$). *A vertex order of a graph $H = (V, E)$ is a fine enumeration if and only if there are no four vertices $v, u, x, w \in V$ with $v \leq_\sigma u \leq_\sigma x \leq_\sigma w$ and $vw \in E$ and $ux \notin E$.*

▶ **Corollary 12.** *Let $H = (V, E)$ be a graph with fine enumeration σ. Let $v, u, x, w \in V$ and $u \leq_\sigma x$ as well as $v \leq_\sigma w$. Then we have (see Figure 4):*

(i) $vw \in E \wedge v \leq_\sigma u \wedge x \leq_\sigma w \Rightarrow ux \in E$
(ii) $ux \notin E \wedge v \leq_\sigma u \wedge x \leq_\sigma w \Rightarrow vw \notin E$
(iii) $vw \in E \wedge ux \notin E \wedge v \leq_\sigma u \Rightarrow w <_\sigma x$
(iv) $vw \in E \wedge ux \notin E \wedge x \leq_\sigma w \Rightarrow u <_\sigma v$.

Now we introduce the forbidden configurations for *simultaneous* enumerations of sunflower unit interval graphs. Throughout this section let $\mathcal{G} = (G_1, \dots, G_k)$ be a sunflower graph with shared graph S and simultaneous enumeration $\zeta = (\zeta_1, \dots, \zeta_k)$. Furthermore, let $V_i = V(G_i)$ and $E_i = E(G_i)$, for $i \in \{1, \dots, k\}$. Finally, let $V = V_1 \cup \dots \cup V_k$. For a straight enumeration η of some graph H we say for $u, v \in V(H)$ that $u <_\eta v$, if u is in a block before v, and we say $u \leq_\eta v$, if $u = v$ or $u <_\eta v$. We call $\leq_\eta$ the *partial order on $V(H)$ corresponding to η*. Note that for distinct u, v in the same block we have neither $u >_\eta v$ nor $u \leq_\eta v$. We write $u \leq_i v$ and $u <_i v$ instead of $u \leq_{\zeta_i} v$ and $u <_{\zeta_i} v$, respectively.

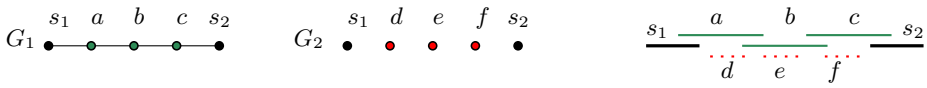
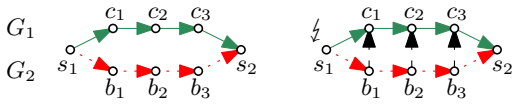


Figure 5 A sunflower graph $\mathcal{G} = (G_1, G_2)$ with shared vertices s_1, s_2 (black, bold). Let ζ be the simultaneous enumeration realized by the given simultaneous proper interval representation. In (G_1, ζ_1) we have the (s_1, s_2)-chain $C = (s_1, a, b, c, s_2)$ of size 5 (green, solid). In (G_2, ζ_2) we have the (s_1, s_2)-bar $B = (s_1, d, e, f, s_2)$ of size 5 (red, dotted). Hence, sunflower graph $\mathcal{G}$ has the conflict (C, B) for the simultaneous enumeration ζ.



(a) (b)

Figure 6 (a): A simultaneous enumeration with conflict. (b): Result with added orderings after scouting, starting at s_2 and finding the conflict in s_1.

Let $u, v \in V(S)$ with $u \neq v$. For $i \in \{1, \dots, k\}$ a (u, v)*-chain* of *size* $m \in \mathbb{N}$ in (G_i, ζ_i) is a sequence $(u = c_1, \dots, c_m = v)$ of vertices in V_i with $c_1 <_i \cdots <_i c_m$ that corresponds to a path in G_i. A (u, v)*-bar* between u and v of *size* $m \in \mathbb{N}$ in (G_i, ζ_i) is a sequence $(u = b_1, \dots, b_m = v)$ of vertices in V_i with $b_1 <_i \cdots <_i b_m$ that corresponds to an independent set in G_i. An example is shown in Figure 5. If there is a (u, v)-chain C in G_i of size $l \geq 2$ and a (u, v)-bar B in (G_j, ζ_j) of size at least l, then we say that (C, B) is a (u, v)*-(chain-bar-)conflict* and that $\mathcal{G}$ *has conflict* (C, B) *for* ζ. Note that one can reduce the size of a larger (u, v)-bar by removing intervals between u and v. Thus, we can always assume that in a conflict, we have a bar and a chain of the same size $l \geq 2$. Assume $\mathcal{G}$ has a simultaneous unit interval representation realizing ζ. If a graph $G \in \mathcal{G}$ has a (u, v)-chain of size $l \geq 2$, then the distance between the intervals I_u, I_v for u, v is smaller than $l - 2$. On the other hand, if a graph $G \in \mathcal{G}$ has a (u, v)-bar of size l, then the distance between I_u, I_v is greater than $l - 2$. Hence, sunflower graph $\mathcal{G}$ has no conflict. The result of this section is that the absence of conflicts is not only necessary, but also sufficient.

▶ **Theorem 13.** *Let $\mathcal{G}$ be a sunflower proper interval graph with simultaneous enumeration ζ. Then $\mathcal{G}$ has a simultaneous unit interval representation that realizes ζ if and only if $\mathcal{G}$ has no conflict for ζ.*

Recall that $V_i = V(G_i)$ for $i \in \{1, \dots, k\}$ and $V = V_1 \cup \cdots \cup V_k$. Let $\alpha^\star$ be the union of the partial orders on $V_1, \dots, V_k$ corresponding to $\zeta_1, \dots, \zeta_k$. Set α to be the transitive closure of $\alpha^\star$. We call α the *partial order on V induced by ζ*. The rough idea is that the partial order on V induced by the simultaneous enumeration ζ is extended in two sweeps to a fine enumeration of some graph H that contains $G_1, \dots, G_k$ as induced subgraphs; see Figures 6, 7. For $(u, v) \in \alpha$ we consider u to be to the left of v. The first sweep (*scouting*) goes from the right to the left and makes only necessary extensions according to Corollary 12 (iv). If there is a conflict, then it is found in this step. Otherwise, we can greedily order the vertices on the way back by additionally respecting Corollary 12 (iii) (*zipping*) to obtain a linear extension where both implications are satisfied. In the last step we decide which edges H has by respecting Corollary 12 (i).

For $h \in \{1, \dots, k\}$ we say two vertices $u, v \in V_h$ are *indistinguishable in* $\mathcal{G}$ if we have $N_{G_i}(u) = N_{G_i}(v)$ for all $i \in \{1, \dots, k\}$ with $u, v \in V_i$. In that case u, v can be represented by the same interval in any simultaneous proper interval representation. Thus, we identify

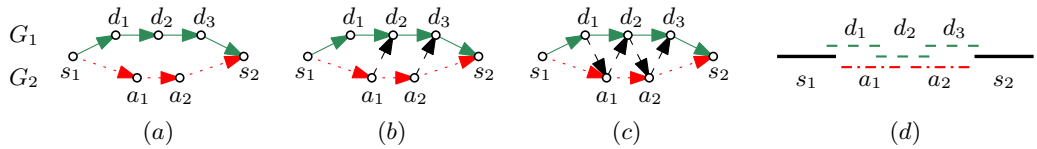


Figure 7 (a): A simultaneous enumeration without conflict. (b): Result with added orderings after scouting. (c): Resulting linear order after zipping. Note that a_1 comes before d_2 in the linear order thanks to scouting. Choosing otherwise would imply a contradiction at s_2. (d): Resulting unit interval representation for the sandwich graph.

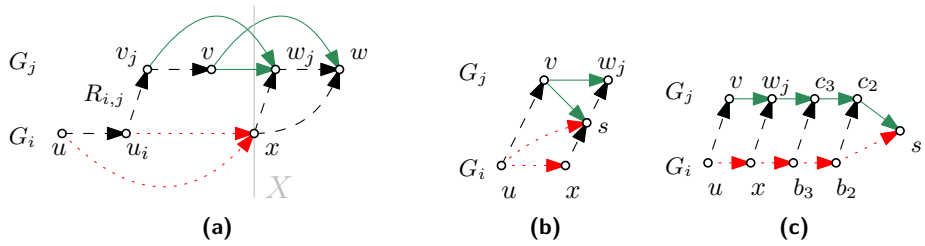


Figure 8 (a): The vertices u_i, v_j, w_j as derived from x and X. The introduced ordering (u_i, v_j) is marked with $R_{i,j}$. (b),(c): Both cases of a chain-bar pair for u and v.

indistinguishable vertices. If $u, v \in V_h$ are not indistinguishable, then we have $N_{G_j}(u) \neq N_{G_j}(v)$ for some $j \in \{1, \dots, k\}$. In that case u, v are ordered by ζ_j and therefore by α. That is, we can assume α to be a linear order on V_i for $i \in \{1, \dots, k\}$. Note that u, v may be ordered even if they are indistinguishable in some input graphs.

For $i \in \{1, \dots, k\}$, let $G_i^c = (V_i, \binom{V_i}{2} \setminus E_i)$ be the complement of G_i. We set $E = \{(u, v) \in \alpha \mid uv \in E_1 \cup \dots \cup E_k\}$ and $F = \{(u, v) \in \alpha \mid uv \in E(G_1^c \cup \dots \cup G_k^c)\}$. We call a partial order σ on V *left-closed* if we have

$$\forall v, w, u, x \in V\colon \ (vw \in E \ \wedge \ ux \in F \ \wedge \ x \leq_\sigma w) \ \Rightarrow \ u <_\sigma v. \tag{1}$$

Note that a fine enumeration of a graph H with $G_1, \dots, G_k$ as induced subgraphs is left-closed by Corollary 12 (iv). We describe the result of the first sweep with the following lemma.

▶ **Lemma 14.** *A sunflower graph $\mathcal{G}$ has no conflict for a simultaneous enumeration ζ if and only if there is a left-closed partial order τ that extends the partial order on $V(\mathcal{G})$ induced by ζ.*

Proof Sketch. If there is a conflict (C, B), then the partial order α induced by ζ cannot be extended to be left-closed since then for $i \in \{1, \dots, k-1\}$ the i'th vertex of C and B must be ordered and distinct while the first vertex is shared; see Figure 6.

Otherwise, we process the vertices from the right to the left and add for each of them the implied orderings (each is considered as vertex x in the definition of left-closed). First consider the case of just two input graphs G_1, G_2. Let X be the set of already processed vertices and let σ be the current partial order. We next process a maximal vertex $x \in V \setminus X$. Let $x \in V_i$. Then we choose u_i to be the rightmost vertex in V_i with $u_i x \in F$ and for $j \neq i$ we choose w_j to be the leftmost vertex in V_j with $x \leq w_j$ and v_j to be the leftmost vertex in V_j with $v_j w_j \in E$; see Figure 8a. Each of u_i, v_j, w_j may not exist. If they do, we extend σ to σ' by adding the ordering $u_i \leq_{\sigma'} v_j$. The other implied orderings are exactly those obtained by transitive closure.

Two vertices $u \in V_1, v \in V_2$ are only ordered by α if there is a shared vertex s with $u \leq_\alpha s \leq_\alpha v$ or $v \leq_\alpha s \leq_\alpha u$. The key observation is that if u is ordered before v due to a necessary extension, then there is a shared vertex s and a (v, s)-chain and a (u, s)-bar of equal

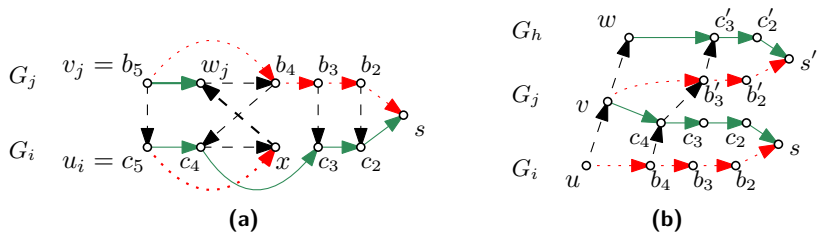


Figure 9 (a): Example situation for $(v_j, u_i) \in \tau_{i,j} \setminus \alpha^\star$. We have the (v_j, s)-bar (v_j, b_4, b_3, b_2, s) and the (u_i, s)-chain (u_i, c_4, c_3, c_2, s) and obtain $x \leq_{\tau_{i,j}} w_j \leq_{\tau_{i,j}} b_4 \leq_{\tau_{i,j}} c_4 \leq_{\tau_{i,j}} x$. (b): Example situation for the transitivity of τ where we have a chain-bar pair for u, v as well as for v, w. We obtain $b_4 \leq_{\tau_{i,j}} c_4 \leq_{\tau_{i,j}} b'_3 \leq_{\tau_{i,j}} c'_3$ and since $u <_\alpha b_4$ and $w <_\alpha c'_3$ we get $b_4 \leq_{\tau_{i,h}} c'_3$ in an appropriate induction and with $\tau_{i,h}$ being left-closed we obtain $u \leq_{\tau_{i,h}} w$. (The base cases for the induction involve shared vertices and thereby only two input graphs.)

size (*chain-bar pair*): If we have $x \leq_\alpha w_j$, then there is a shared vertex $x \leq_\alpha s \leq_\alpha w_j$ and by Theorem 10 we obtain $us \in F$ and $vs \in E$, which yields a chain-bar pair; see Figure 8b. Otherwise we have a chain-bar pair for x and w_j that can be extended by u and v; see Figure 8c. With the absence of conflicts this ensures that vertices ordered according to the left-closed property are actually distinct.

Assume a new extension would violate the property of antisymmetry. This would mean we already had $v_j <_\sigma u_i$, which would imply a cyclic ordering of x, w_j with elements of the (necessary) chain-bar pair for v_j, u_i in a prior step; see Figure 9a. Finally, for more than two input graphs we obtain a corresponding ordering $\tau_{i,j}$ for each pair of input graphs G_i, G_j. Let $\tau = \bigcup_{i,j \in \{1,\dots,k\}} \tau_{i,j}$ be their union. For $u <_{\tau_{i,j}} v <_{\tau_{j,h}} w$ we can prove $u <_{\tau_{i,h}} w$ by using chain-bar pairs and induction; see Figure 9b. Hence, τ is already transitive and the other properties are easy to verify. ◀

By respecting the orderings obtained by scouting we avoid wrong decisions when greedily adding vertices to a linear ordering in the zipping step; see Figure 7.

▶ **Lemma 15.** *Let $\mathcal{G}$ be a sunflower graph with a simultaneous enumeration ζ. There is a left-closed linear order τ that extends the partial order α on $V(\mathcal{G})$ induced by ζ if and only if there is a left-closed partial order $\sigma \supseteq \alpha$.*

Proof Sketch. Given σ we process the vertices from the left to the right. We add in each step a leftmost vertex u of the remaining vertices to a set U of the processed vertices that are linearly ordered. We denote the current order by σ'. Vertex u is then ordered before all other vertices in $V \setminus U$. To avoid that the left-closed property is violated when adding such orderings for another vertex, we ensure our extended order $\sigma'' \supseteq \sigma'$ is *right-closed on U* meaning that

$$\forall u, v \in U, w, x \in V \colon (vw \in E \wedge ux \in F \wedge v \leq u) \Rightarrow w < x. \tag{2}$$

To this end, we consider the current vertex u as vertex u in the definition of right-closed and add all implied orderings in σ''. This means for each vertex $y \in Y = \{y \in V \mid \exists u' \in U \colon uy \in E\}$ and each vertex $z \in Z = \{z \in V \mid uz \in F\}$ we set $y \leq_{\sigma''} z$; see Figure 10a. We further extend σ'' to be transitive. Note that there are no two vertices $y \in Y, z \in Z$ with $y \leq_\sigma z$, since σ is left-closed and for $u' \in U$ we have $u' \leq_\sigma u$. With this observation we can verify that σ'' is antisymmetric and left-closed; see Figure 10b. ◀

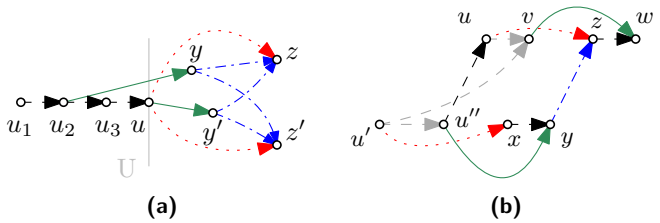


Figure 10 (a): orderings added during a zipping step (blue dash-dotted). All vertices in $Y = \{y, y'\}$ are ordered before those in $Z = \{z, z'\}$. (b): The case for σ'' being left-closed where we have $x \leq_{\sigma''} w$ due to transitivity. This means there is some ordering $(y, z) \in Y \times Z$ with $x \leq_{\sigma'} y \leq_{\sigma''} z \leq_{\sigma'} w$. We further have a vertex $u'' \in U$ with $u''y \in E$ and $uz \in F$. Given vertices $u', v \in V$ with $u'x \in F$ and $vw \in E$ we obtain $u' <_{\sigma'} u''$ and $u <_{\sigma'} v$ since σ' is left-closed. This yields $u' <_{\sigma''} v$.

Finally, we construct a graph $H = (V, E')$ for which the obtained linear order τ is a fine enumeration. We do so by setting $E' = \{ux \in V^2 \mid \exists vw \in E\colon v \leq_\tau u <_\tau x \leq_\tau w\}$ in accordance with Corollary 12 (i).

▶ **Lemma 16** $(\star)$. *Let $\mathcal{G} = (G_1, \ldots, G_k)$ be a sunflower graph with a simultaneous enumeration ζ. A linear order τ that extends the partial order on $V(\mathcal{G})$ induced by ζ is a fine enumeration for some graph H that has $G_1, \ldots, G_k$ as induced subgraphs if and only if τ is left-closed.*

Combining Lemmas 9, 14, 15 and 16 we obtain Theorem 13.

▶ **Theorem 13.** *Let $\mathcal{G}$ be a sunflower proper interval graph with simultaneous enumeration ζ. Then $\mathcal{G}$ has a simultaneous unit interval representation that realizes ζ if and only if $\mathcal{G}$ has no conflict for ζ.*

4.2 Recognizing Simultaneous Unit Interval Graphs in Polynomial Time

With Theorems 8 and 13 we can now efficiently recognize simultaneous unit interval graphs.

▶ **Theorem 17.** *Given a sunflower graph $\mathcal{G} = (G_1, \ldots, G_k)$, we can decide in $O(|V| \cdot |E|)$ time, whether $\mathcal{G}$ is a simultaneous unit interval graph, where $(V, E) = G^\star = G_1 \cup \cdots \cup G_k$. If it is, then we also provide a simultaneous unit interval representation in the same time.*

Proof Sketch. Here we establish polynomial time recognition, and the stated time is proven in the full version. As discussed earlier, we can assume that $G^\star$ is connected. With Theorem 6 we obtain a simultaneous enumeration ζ of $\mathcal{G}$, unless $\mathcal{G}$ is not a simultaneous proper interval graph. By Theorem 13, the sunflower graph $\mathcal{G}$ is a simultaneous unit interval graph if and only if there is a simultaneous enumeration η for which $\mathcal{G}$ has no conflict. In that case η^r also has no conflict. With Theorem 8 we have that η or η^r is obtained from ζ by reversals of reversible parts and independent components. Hence, we only need to consider simultaneous enumerations obtained that way.

Since every single graph G_i is proper, it has no conflict and we only need to consider (u, v)-conflicts with $u, v \in V(S)$, where S is the shared graph. The minimal (u, v)-chains for G_i are exactly the shortest paths in G_i and thus independent from reversals. On the other hand, for the maximal size of (u, v)-bars in G_i only the reversals of the two corresponding components C, D of u, v are relevant, while components in-between always contribute their maximum independent set regardless of whether they are reversed. We can thus compute

for $i, j \in \{1, \ldots, k\}$, $u, v \in V(S)$ and each of the four combinations of reversal decisions (reverse or do not reverse) for the corresponding components C, D of u, v, whether they yield a conflict at (u, v). We can formulate a corresponding 2-SAT formula $\mathcal{F}$: For every independent component and every reversible part we introduce a literal that represents whether it is reversed or not. For every combination of two reversal decisions that yields a conflict we add a clause that excludes this combination. If $\mathcal{F}$ is not satisfiable, then every simultaneous enumeration yields a conflict. Otherwise, a solution yields a simultaneous enumeration without conflict. We obtain a simultaneous unit interval representation by following the construction in Section 4.1. ◀

5 Conclusion

We studied the problem of simultaneous representations of proper and unit interval graphs. We have shown that, in the sunflower case, both simultaneous proper interval graphs and simultaneous unit intervals can be recognized efficiently. While the former can be recognized by a simple and straightforward recognition algorithm, the latter is based on the three ingredients: 1) a complete characterization of all simultaneous proper interval representations of a sunflower simultaneous graph, 2) a characterization of the simultaneous proper interval representations that can be realized by a simultaneous unit interval representation and 3) an algorithm for testing whether among the simultaneous proper interval representations there is one that satisfies this property.

Future Work. While our algorithm for (sunflower) simultaneous proper interval graphs has optimal linear running time, we leave it as an open problem whether simultaneous unit interval graphs can also be recognized in linear time.

Our main open question is about the complexity of sunflower simultaneous interval graphs. Jampani and Lubiw [18] conjecture that they can be recognized in polynomial time for any number of input graphs. However, even for three graphs the problem is still open.

References

1. Patrizio Angelini, Giordano Da Lozzo, and Daniel Neuwirth. On Some $\mathcal{NP}$-complete SEFE Problems. In Sudebkumar Prasant Pal and Kunihiko Sadakane, editors, *Algorithms and Computation: 8th International Workshop, WALCOM 2014, Chennai, Proceedings*, pages 200–212. Springer, 2014. doi:10.1007/978-3-319-04657-0_20.
2. Thomas Bläsius, Stephen G. Kobourov, and Ignaz Rutter. Simultaneous Embedding of Planar Graphs. *CoRR*, abs/1204.5853, 2012. arXiv:1204.5853.
3. Thomas Bläsius and Ignaz Rutter. Simultaneous PQ-Ordering with Applications to Constrained Embedding Problems. *ACM Trans. Algorithms*, 12(2):16:1–16:46, 2015. doi:10.1145/2738054.
4. Jan Bok and Nikola Jedličková. A note on simultaneous representation problem for interval and circular-arc graphs. *arXiv preprint*, 2018. arXiv:1811.04062.
5. Kellogg S. Booth. *PQ Tree Algorithms*. PhD thesis, University of California, Berkeley, 1975.
6. Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1.
7. Peter Brass, Eowyn Cenek, Cristian A Duncan, Alon Efrat, Cesim Erten, Dan P Ismailescu, Stephen G Kobourov, Anna Lubiw, and Joseph SB Mitchell. On simultaneous planar graph embeddings. *Computational Geometry*, 36(2):117–130, 2007. doi:10.1016/j.comgeo.2006.05.006.

8 Steven Chaplick, Radoslav Fulek, and Pavel Klavík. Extending Partial Representations of Circle Graphs. In Stephen Wismath and Alexander Wolff, editors, *Graph Drawing: 21st International Symposium, GD 2013, Bordeaux, Revised Selected Papers*, pages 131–142. Springer, 2013. doi:10.1007/978-3-319-03841-4_12.

9 Celina M. Herrera de Figueiredo, João Meidanis, and Célia Picinin de Mello. A linear-time algorithm for proper interval graph recognition. *Information Processing Letters*, 56(3):179–184, 1995. doi:10.1016/0020-0190(95)00133-W.

10 Xiaotie Deng, Pavol Hell, and Jing Huang. Linear-Time Representation Algorithms for Proper Circular-Arc Graphs and Proper Interval Graphs. *SIAM J. Comput.*, 25(2):390–403, 1996. doi:10.1137/S0097539792269095.

11 Alejandro Estrella-Balderrama, Elisabeth Gassner, Michael Jünger, Merijam Percan, Marcus Schaefer, and Michael Schulz. Simultaneous Geometric Graph Embeddings. In Seok-Hee Hong, Takao Nishizeki, and Wu Quan, editors, *Graph Drawing: 15th International Symposium, GD 2007, Sydney. Revised Papers*, pages 280–290. Springer, 2008. doi:10.1007/978-3-540-77537-9_28.

12 Elisabeth Gassner, Michael Jünger, Merijam Percan, Marcus Schaefer, and Michael Schulz. Simultaneous Graph Embeddings with Fixed Edges. In Fedor V. Fomin, editor, *Graph-Theoretic Concepts in Computer Science: 32nd International Workshop, WG 2006, Bergen, Revised Papers*, pages 325–335. Springer, 2006. doi:10.1007/11917496_29.

13 Paul W. Goldberg, Martin C. Golumbic, Haim Kaplan, and Ron Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995. doi:10.1089/cmb.1995.2.139.

14 Martin Charles Golumbic. *Algorithmic Graph Theory and Perfect Graphs (Annals of Discrete Mathematics, Vol 57)*. North-Holland Publishing Co., 2004.

15 Martin Charles Golumbic, Haim Kaplan, and Ron Shamir. Graph sandwich problems. *Journal of Algorithms*, 19(3):449–473, 1995. doi:10.1006/jagm.1995.1047.

16 Pinar Heggernes, Pim van 't Hof, Daniel Meister, and Yngve Villanger. Induced Subgraph Isomorphism on proper interval and bipartite permutation graphs. *Theoretical Computer Science*, 562:252–269, 2015. doi:10.1016/j.tcs.2014.10.002.

17 Pavol Hell, Ron Shamir, and Roded Sharan. A Fully Dynamic Algorithm for Recognizing and Representing Proper Interval Graphs. *SIAM J. Comput.*, 31(1):289–305, 2002. doi:10.1137/S0097539700372216.

18 Krishnam Raju Jampani and Anna Lubiw. Simultaneous Interval Graphs. In Otfried Cheong, Kyung-Yong Chwa, and Kunsoo Park, editors, *Algorithms and Computation: 21st International Symposium, ISAAC 2010, Jeju Island, Proceedings, Part I*, pages 206–217. Springer, 2010. doi:10.1007/978-3-642-17517-6_20.

19 Krishnam Raju Jampani and Anna Lubiw. The Simultaneous Representation Problem for Chordal, Comparability and Permutation Graphs. *Journal of Graph Algorithms and Applications*, 16(2):283–315, 2012. doi:10.7155/jgaa.00259.

20 Pavel Klavík, Jan Kratochvíl, Yota Otachi, Ignaz Rutter, Toshiki Saitoh, Maria Saumell, and Tomáš Vyskočil. Extending Partial Representations of Proper and Unit Interval Graphs. *Algorithmica*, 77(4):1071–1104, April 2017. doi:10.1007/s00453-016-0133-z.

21 Peter J Looges and Stephan Olariu. Optimal greedy algorithms for indifference graphs. *Computers & Mathematics with Applications*, 25(7):15–25, 1993. doi:10.1016/0898-1221(93)90308-I.

22 Ross M McConnell and Yahav Nussbaum. Linear-time recognition of probe interval graphs. In Amos Fiat and Peter Sanders, editors, *Proceedings of the 17th Annual European Symposium on Algorithms (ESA'09)*, volume 5757 of *Lecture Notes in Computer Science*, pages 349–360. Springer, 2009. doi:10.1007/978-3-642-04128-0_32.

23 Yahav Nussbaum. Recognition of probe proper interval graphs. *Discrete Applied Mathematics*, 167:228–238, 2014. doi:10.1016/j.dam.2013.11.013.

24 Fred S Roberts. *Representations of indifference relations*. PhD thesis, Department of Mathematics, Stanford University, 1968.

25 Fred S. Roberts. Indifference Graphs. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, New York, 1969.

26 Marcus Schaefer. Toward a Theory of Planarity: Hanani-Tutte and Planarity Variants. In Walter Didimo and Maurizio Patrignani, editors, *Graph Drawing: 20th International Symposium, GD 2012, Redmond, Revised Selected Papers*, pages 162–173. Springer, 2013. doi:10.1007/978-3-642-36763-2_15.

27 Dale J. Skrien. A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs. *Journal of Graph Theory*, 6(3):309–316, 1982. doi:10.1002/jgt.3190060307.

28 Jeremy P. Spinrad. *Efficient Graph Representations*. Fields Institute Monographs. AMS, 2003.