Algorithmica
https://doi.org/10.1007/s00453-025-01296-x

Simultaneous Representation of Proper and Unit Interval Graphs

Ignaz Rutter[1] · Darren Strash[2] · Peter Stumpf[1,4,5] · Michael Vollmer[3]

Received: 13 February 2023 / Accepted: 17 January 2025

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.

Ignaz Rutter, Darren Strash, Peter Stumpf, and Michael Vollmer contributed equally to this work.

✉ Peter Stumpf
stumpf@kam.mff.cuni.cz

Ignaz Rutter
rutter@fim.uni-passau.de

Darren Strash
dstrash@hamilton.edu

Michael Vollmer
mi.r.vollmer@gmail.com

1 Faculty of Computer Science and Mathematics, University of Passau, Passau, Germany

2 Department of Computer Science, Hamilton College, Clinton, NY, USA

3 Department of Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

4 Department of Applied Mathematics, Charles University, Prague, Czechia

5 Department of Theoretical Computer Science, Faculty of Information Technology, Czech Technical University in Prague, Prague, Czechia

Published online: 28 February 2025

Keywords Proper interval graphs · Unit interval graphs · Simultaneous representation · Geometric intersection graphs · Recognition

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 Fig. 1a, b.

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 [1, 2]. However, 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 [3]. Likewise, in genome reconstruction, we can ask if a sequence of DNA can be reconstructed from strands that have sequences in common [4].

Simultaneous representations capture this in a very natural way. Given *simultaneous graphs* $G_1, \ldots, G_k$ 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 [5], circle graphs [6], as well as proper and unit interval graphs [5]. For interval graphs, Bläsius and Rutter [7] 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 [8, 9], 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 [10–13]. The notion of simultaneous representation of general intersection graph classes was introduced by Jampani and Lubiw [3], 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

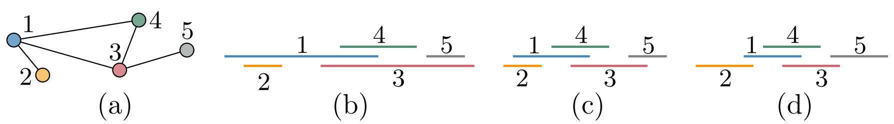

Fig. 1 **a** A graph, with **b** an interval representation, **c** a proper interval representation, **d** a unit interval representation

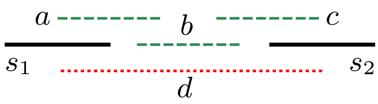

Fig. 2 A simultaneous proper interval representation of a sunflower graph $\mathcal{G}$ consisting of two paths $G_1 = (s_1, a, b, c, s_2)$ (dashed) and $G_2 = (s_1, d, s_2)$ (dotted) with shared start and end s_1, s_2 (bold). They have no simultaneous unit interval representation: The intervals a and 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 1

same subgraph (the *sunflower case*). They further showed that recognizing three or more simultaneous chordal graphs is NP-complete.

Golumbic et al. [14] 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 [3].

We consider simultaneous *proper* and *unit* interval graphs. An interval graph is proper if it has an interval representation where no interval properly contains another one, and it is unit if all intervals have length one; see Fig. 1c, d. Interestingly, while proper and unit interval graphs are the same graph class, as shown by Roberts [15], simultaneous unit interval graphs differ from simultaneous proper interval graphs; see Fig. 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. 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 [16, 17].

Simultaneous interval graphs were first studied by Jampani and Lubiw [18] who gave a $\mathcal{O}(n^2 \log n)$-time recognition algorithm for the special case of two simultaneous graphs. Bläsius and Rutter [7] 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 [19]. However, the complexity for the sunflower case with more than two simultaneous graphs is still open.

Our Results

We settle these problems with the number of input graphs 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 [20–23]. 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. We prove that the general case is NP-complete.

Organization

We begin by introducing basic notation and existing tools throughout Sect. 2. In Sect. 3 we give a characterization of simultaneous proper interval graphs, from which we develop an efficient recognition algorithm. In Sect. 4 we characterize simultaneous proper interval representations of simultaneous unit interval graphs, and then exploit this property to efficiently search for a simultaneous unit interval representation among the set of all simultaneous proper interval graph representations.

2 Preliminaries

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

2.1 Graphs, Orders, and PQ-trees

Let σ be a binary relation (not necessarily a partial order). 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 relation 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 orders 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 $O(|X| + \sum_{C \in \mathcal{C}} |C|)$ [24]. 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 orders 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 [25].

2.2 Interval Graphs, Proper Interval Graphs, and Their Subgraphs

Unless mentioned explicitly, all graphs in this paper are undirected. 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 *([15]) 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 Fig. 2.

We use the well-known characterization of proper interval graphs using *straight enumerations* [20]. Two adjacent vertices $u, v \in V$ are *indistinguishable* if we have $N[u] = N[v]$ where $N[u] = \{v : uv \in E(G)\} \cup \{u\}$ 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 $\sigma = \sigma(R)$ [20]. A *fine enumeration* of a graph G is a linear order η of $V(G)$ such that for $u \in V(G)$ the closed neighborhood $N_G[u]$ is consecutive in η.

Proposition 2 *([20, 26, 27]) (i) A graph is a proper interval graph if and only if it admits a fine enumeration. (ii) A graph is a proper interval graph if and only if it admits a straight enumeration. (iii) The 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. For $\mathcal{G}$, we define the *union graph* $\bigcup \mathcal{G} = \bigcup_{i=1}^{k} G_i$ and set $V = V(\bigcup \mathcal{G})$.

A *simultaneous (proper/unit) interval representation* $\mathcal{R} = (R_1, \ldots, R_k)$ of $\mathcal{G}$ is a tuple of representations such that each R_i 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. Whenever a simultaneous graph $\mathcal{G}$ or a simultaneous representation $\mathcal{R}$ is given, it is implied that $\mathcal{G} = (G_1, \ldots, G_k)$ with $G_i = (V_i, E_i)$ for $i \in \{1, \ldots, k\}$ and $\mathcal{R} = (R_1 \ldots, R_k)$.

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*. Whenever a sunflower graph $\mathcal{G}$ is given, it is implied that its shared graph is denoted by $S = (V_S, V_E)$. 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 $i, j \in \{1, \dots, k\}$. 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 $\mathcal{G}$ with connected union graph.

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

Proof Clearly, a simultaneous (proper/unit) interval representation $\mathcal{R}$ of $\mathcal{G}$ induces a representation for each $\mathcal{G}_i$. Conversely, given simultaneous (proper/unit) interval representations $\mathcal{R}_i$ of $\mathcal{G}_i$ for $i \in \{1, \dots, l\}$, we can combine them such that all intervals in $\mathcal{R}_i$ are placed to the right of all intervals in $\mathcal{R}_{i-1}$ for $i \in \{2, \dots, l\}$ to obtain a simultaneous (proper/unit) interval representation $\mathcal{R}$ of $\mathcal{G}$. □

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 simultaneous graph with a connected union 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 $U \subseteq V$ be a subset of vertices. We call σ *compatible* with a linear order ζ of U if we have for $u, v \in U$ that $u \leq_\zeta v$ implies $B(u, G) \leq_\sigma B(v, G)$.

Lemma 4 *Let $\mathcal{G}$ be a sunflower graph. Then $\mathcal{G}$ admits a simultaneous proper interval representation $\mathcal{R}$ if and only if there exists a linear order ζ of the shared vertices V_S and straight enumerations σ_i for each G_i that are compatible with ζ.*

Proof Assume $\mathcal{R}$ is a simultaneous proper interval representation of $\mathcal{G}$ with corresponding straight enumerations $\sigma_i = \sigma(R_i)$. Let ζ be a linear order of V_S according to their left endpoints in $\mathcal{R}$, breaking ties arbitrarily. We claim that each σ_i is compatible with ζ. If σ_i is not compatible with ζ, there exist vertices $u <_\zeta v$ such that $B(v, G_i) <_{\sigma_i} B(u, G_i)$. By the definition of extracted straight enumerations, this implies that the interval of v has its left endpoint before the interval of u in R_i, which contradicts $u <_\zeta v$.

Conversely, we show how to construct a simultaneous proper interval representation $\mathcal{R}$ of $\mathcal{G}$ using a linear order ζ of V_S and straight enumerations σ_i of each $G_i \in \mathcal{G}$ compatible with ζ. An illustration of the following construction is given in Fig. 3. For each graph G_i with vertex set $V_i = \{v_1, \dots, v_q\}$ let $M_i = \{x_1, y_1, \dots, x_q, y_q\}$ be

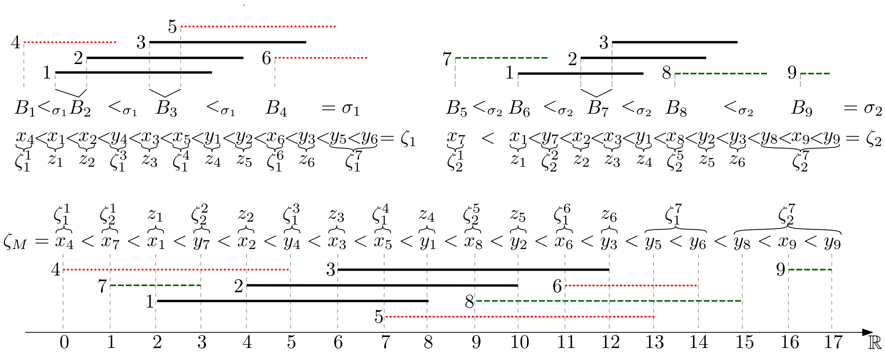

Fig. 3 An illustration of the construction in the proof of Lemma 4. Top: Graph G_1 on the left, Graph G_2 on the right. They share the clique $\{1, 2, 3\}$ and their straight enumerations σ_1, σ_2 are compatible with the order of shared vertices $1 < 2 < 3$. The interval representations of G_1 and G_2 are shown as visualization and are not a determining factor in the construction of ζ_1 and ζ_2. Bottom: The constructed simultaneous interval representation $\mathcal{R} = (R_1, R_2)$ with the linear order σ_M from which $\mathcal{R}$ is derived. The shared intervals are represented by black bold lines, representation R_1 is illustrated by dotted red lines while R_2 is illustrated by dashed green lines

the set of corresponding interval endpoints. We construct a linear order ζ_i of M_i as follows. For any two vertices $v_j, v_l \in V_i$ we define ζ_i as

$$x_j \leq_{\zeta_i} x_l \text{ and } y_j \leq_{\zeta_i} y_l \Leftrightarrow \begin{cases} B(v_j, G_i) <_{\sigma_i} B(v_l, G_i) \text{ or} \\ B(v_j, G_i) = B(v_l, G_i) \text{ and } j \leq l \end{cases} \tag{1}$$

$$\begin{aligned} x_j \leq_{\zeta_i} y_l &\Leftrightarrow B(v_j, G_i) \leq_{\sigma_i} B(v_l, G_i) \text{ or } v_j \in N[v_l] \\ y_l \leq_{\zeta_i} x_j &\Leftrightarrow B(v_l, G_i) <_{\sigma_i} B(v_j, G_i) \text{ and } v_j \notin N[v_l]. \end{aligned} \tag{2}$$

Since σ_i is compatible with ζ, it is clear that the restriction $\zeta' = z_1 <_{\zeta'} \cdots <_{co'} z_{2|V_S|}$ of ζ_i to $\{x_j, y_j \in M_i : w_j \in V_S\}$ is the same for all $i \in \{1, \ldots, k\}$. Therefore, each linear order ζ_i can be represented as $\zeta_i = \zeta_i^1 \circ z_1 \circ \zeta_i^2 \circ z_2 \circ \cdots \circ \zeta_i^{2p} \circ z_{2p} \circ \zeta_i^{2p+1}$, which allows us to combine all ζ_i as follows. Let $\zeta^j = \zeta_1^j \circ \cdots \zeta_k^j$, for $j \in \{1, \ldots, 2p+1\}$. For the union set $M = \bigcup_{i=1}^k M_i$, which contains the interval endpoints of vertices in V_S only once, we construct the linear order $\zeta_M = \zeta^1 \circ z_1 \circ \zeta^2 \circ z_2 \circ \cdots \circ \zeta^{2p} \circ z_{2p} \circ \zeta^{2p+1}$.

Additionally, for each $m \in M$ we define $\#_M(m)$ as the number of elements before m in ζ_M. That is, $\#_M(m) = |\{m' \in M : m' <_{\zeta_M} m\}|$. This allows us to construct the interval representations $R_i = \{[\#_M(x_v), \#_M(y_v)] : v \in V_i\}$, which yields the simultaneous interval representation $\mathcal{R} = (R_1, \ldots, R_k)$. It remains to show that each R_i is a proper interval representation and that it is an interval representation of the respective graph G_i.

Suppose for sake of contradiction that there exists an interval representation R_i that is not a proper interval representation. Then there exist two vertices $v_j \neq v_l \in V_i$ such that the interval of v_j properly contains the interval of v_l in R_i. By construction of R_i this implies $x_j <_{\zeta_i} x_l$ and $y_l <_{\zeta_i} y_j$. Using the construction rules (1), this is only possible if $B(v_j, G_i) = B(v_l, G_i)$ and $j = l$, which contradicts the assumption $v_j \neq v_l$.

To show that each interval representation R_i is a representation of G_i, we show that R_i models exactly the edges of G_i. For any two vertices $v_j, v_l \in V_i$ with $\{v_j, v_l\} \notin E_i$ we have $v_j \notin N[v_l]$ and thus $B(v_j, G_i) \neq B(v_l, G_i)$. Without loss of generality, assume $B(v_j, G_i) <_{\sigma_i} B(v_l, G_i)$. By (2) it follows that $y_j <_{\zeta_M} x_l$, which means that the interval of v_j ends before the interval of v_l begins in R_i. Conversely, let $v_j, v_l \in V_i$ share an edge $\{u_j, v_l\} \in E_i$. By (2) it follows that $x_j \leq_{\zeta_M} y_l$ and $x_l \leq_{\zeta_M} y_j$. This means that both intervals begin before either of them ends in R_i. Therefore each R_i is a proper interval representation of G_i. Note that by construction $\sigma(R_i) = \sigma_i$. □

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 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 anonther straight enumeration.

Theorem 5 *Let $\mathcal{G}$ be a sunflower graph. There exists a simultaneous proper interval representation $\mathcal{R}$ 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 a simultaneous proper interval representation $\mathcal{R}$ with $\sigma(R_i) = \sigma_i$ for $i \in \{1, \ldots, k\}$.*

Proof Let $S = (V_S, E_S)$ be the shared graph of $\mathcal{G}$. If $\mathcal{G}$ is a simultaneous proper interval graph, there exist straight enumerations σ_i of G_i that are compatible with a linear order ζ of V_S by Lemma 4. By definition of compatible, for any vertices $u \leq_\zeta v$ we have $B(u, G_i) \leq_{\sigma_i} B(v, G_i)$ for each $i \in \{1, \ldots, k\}$. Since ζ and each straight enumerations are linear orders, we have $B(u, G_i) \leq_{\sigma_i} B(v, G_i) \Leftrightarrow B(u, G_j) \leq_{\sigma_j} B(v, G_j)$ for any $i, j \in \{1, \ldots, k\}$. As a result, $(\sigma_1, \ldots, \sigma_k)$ is a simultaneous enumeration of $\mathcal{G}$.

Conversely, let $(\sigma_1, \ldots, \sigma_k)$ be a simultaneous enumeration of $\mathcal{G}$. Then $\zeta_P = \{(u, v) \in V_S \times V_S \mid \exists i : B(u, G_i) <_{\sigma_i} B(v, G_i)\}$ is a partial order of V_S. Let ζ be a linear order of V_S extending ζ_P. By definition of simultaneous enumerations and construction of ζ_P, each straight enumeration σ_i is compatible with ζ. Then $\mathcal{G}$ is a simultaneous proper interval graph by Lemma 4. Additionally, the construction in Lemma 4 yields a simultaneous proper interval representation $\mathcal{R} = (R_1, \ldots, R_k)$ 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 $\mathcal{G}$ be a sunflower graph. 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 fine enumerations of S that can be extended to a fine enumeration of each graph G_i. By Theorem 5, $\mathcal{G}$ admits a simultaneous proper interval representation if and only if T is not the null-tree.

In that case, we obtain a simultaneous representation by choosing any order $O \in$ CONSISTENT(T) and constructing a simultaneous representation $\mathcal{S}$ of S. Using the algorithm of Klavík et al. [5], we then independently extend $\mathcal{S}$ to representations R_i of G_i. Since the trees T_i can be computed in time linear in $|V(G_i)| + |E(G_i)|$, and the intersection of two trees takes linear time, the testing algorithm takes time linear in $\sum_{i=1}^{k} |V(G_i)| + |E(G_i)|$. The representation extension of Klavík et al. [5] runs in linear time. We therefore have the following theorem.

Theorem 6 *Given a sunflower graph $\mathcal{G}$, 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 sunflower proper interval graph with $\bigcup G$ connected. Across all simultaneous proper interval representations $\mathcal{R}$ of $\mathcal{G}$, the straight enumeration $\sigma_S(\mathcal{R})$ of the shared graph S is unique up to reversal.*

Proof Let $\mathcal{R}$ be a simultaneous representation of $\mathcal{G}$ and let $\sigma_S(\mathcal{R})$ be the straight enumeration of S induced by $\mathcal{R}$. Since $\bigcup \mathcal{G}$ 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 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 $\sigma_1 \circ \cdots \circ \sigma_k$ be a straight enumeration of G. Then we say the straight enumeration $\sigma' = \sigma_1 \circ \cdots \circ \sigma_{i-1} \circ \sigma_i^r \circ \sigma_{i+1} \circ \cdots \circ \sigma_k$ is *obtained from σ by reversal of C_i*; see Fig. 4. For a sunflower graph $\mathcal{G}$, we call a component $C = (V_C, E_C)$ in G_i *loose*, if all vertices in $V_S \cap V_C$ are in the same block of S. Reversal of loose components is the only "degree of freedom" among the set of simultaneous enumerations, besides full reversal.

Lemma 8 *Let $\mathcal{G}$ be a sunflower proper interval graph with a simultaneous enumeration $(\sigma_1, \dots, \sigma_k)$ where $\bigcup G$ is connected. Then for any simultaneous enumeration*

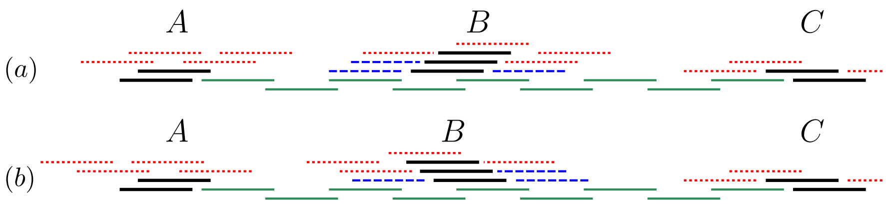

Fig. 4 **a** Simultaneous proper interval representation of G_1 (green solid), G_2 (red dotted), G_3 (blue dashed) with shared graph S (black bold). S has three blocks A, B, C. We denote the component of G_i containing a block D by C_D^i. $C_A^2, C_B^2, C_B^3, C_C^2$ are loose. C_A^2 is independent. (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. In **b** C_A^2 and (C_B^2, C_B^3) are reversed

$(\sigma_1', \dots, \sigma_k')$ of $\mathcal{G}$, the straight enumeration σ_i' can be obtained from σ_i or σ_i^r by reversal of loose components.

Proof By Theorem 5, let $\mathcal{R}$ and $\mathcal{R}'$ be simultaneous proper interval representations of $\mathcal{G}$ with $\sigma(R_i) = \sigma_i$ and $\sigma(R_i') = \sigma_i'$. Then by Lemma 7, $\sigma_S(\mathcal{R}')$ is either equal to $\sigma_S(\mathcal{R})$ or equal to $\sigma_S(\mathcal{R})^r$. Without loss of generality assume $\sigma_S(\mathcal{R}') = \sigma_S(\mathcal{R})$. We now show that σ_i' can be obtained from σ_i by reversal of loose components.

Let $1 \le i \le k$. For each connected component of G_i, its straight enumeration is unique up to reversal by Theorem 2. This means that σ_i and σ_i' can only differ by reversal and reordering of straight enumerations of individual components of G_i. Since $\bigcup \mathcal{G}$ is connected, each component of G_i contains at least one vertex and thus at least one block of S. Since we have $\sigma_S(R_i) = \sigma_S(R_i')$, the order of blocks of S in σ_i and σ_i' is identical. This means that the straight enumerations of components of G_i are ordered identically in σ_i and σ_i' and that no straight enumeration containing vertices from more than one block of S is reversed. As a result, the only difference between σ_i and σ_i' is the reversal of loose components of G_i. □

To obtain a complete characterization, we now introduce additional terms to specify which reversals result in simultaneous enumerations; see Fig. 4. Let $\mathcal{G} = (G_1, \dots, G_k)$ be a sunflower proper interval graph with connected union graph and 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.

Lemma 9 *Let $\mathcal{G}$ be a sunflower proper interval graph with simultaneous enumeration $(\sigma_1, \dots, \sigma_k)$. Let $C \subseteq G_i$ and $C' \subseteq G_j$ be components oriented at a block B of S. Then there exist two vertices $u, v \in B$ such that $B(u, C) <_{\sigma_i} B(v, C)$ and $B(u, C') <_{\sigma_j} B(v, C')$.*

Proof Let $s, t \in B$ be from the "leftmost" and "rightmost" block of σ_i that contain vertices of B, respectively, i.e., for all $x \in B$ we have $B(s, C) \le_{\sigma_i} B(x, C) \le_{\sigma_i} B(t, C)$. Let $s', t' \in B$ be analogous vertices for σ_j and C'. Since C and C' are oriented at B, it follows that $B(s, C) \neq B(t, C)$ and $B(s', C') \neq B(t', C')$. This means that any two vertices $u \in B(s, C) \cap B(s', C')$ and $v \in B(t, C) \cap B(t', C')$ fulfill the lemma. It thus remains to show that $B(s, C) \cap B(s', C') \neq \emptyset$ and $B(t, C) \cap B(t', C') \neq \emptyset$.

If $s' \in B(s, C)$ we have $B(s, C) \cap B(s', C') \neq \emptyset$. Otherwise, we have $s' \notin B(s, C)$ and it follows that $B(s, C) <_{\sigma_i} B(s', C)$. By definition of simultaneous enumerations this implies $B(s, C') \leq_{\sigma_j} B(s', C')$. By definition of s', it follows that $B(s, C') = B(s', C')$ and thus $B(s, C') \cap B(s', C') = B(s, C') \neq \emptyset$. The proof that $B(t, C) \cap B(t', C') \neq \emptyset$ works analogously. □

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*; see Fig. 4. 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 $(\sigma_1, \ldots, \sigma_k)$ and $(\sigma'_1, \ldots, \sigma'_k)$ be tuples of straight enumerations of $G_1, \ldots, G_k$. We say $(\sigma'_1, \ldots, \sigma'_k)$ is *obtained from* $(\sigma_1, \ldots, \sigma_k)$ *by reversal* of *reversible part* $\mathcal{C}(B)$, if $\sigma'_1, \ldots, \sigma'_k$ are obtained by reversal of all components in $\mathcal{C}(B)$. Similarly, $(\sigma_1, \ldots, \sigma_{i-1}, \sigma''_i, \sigma_{i+1}, \ldots, \sigma_k)$ is *obtained from* $(\sigma_1, \ldots, \sigma_k)$ *by reversal* of *independent component* C, if σ''_i is obtained by reversal of C.

Theorem 10 *Let $\mathcal{G} = (G_1, \ldots, G_k)$ be a sunflower graph with connected union graph and shared graph S and simultaneous enumeration $\rho = (\sigma_1, \ldots, \sigma_k)$. Then $\rho' = (\sigma'_1, \ldots, \sigma'_k)$ is a simultaneous enumeration of $\mathcal{G}$ if and only if ρ' can be obtained from ρ or ρ^r by reversal of independent components and reversible parts.*

Proof We first show that ρ' is a simultaneous enumeration if it is obtained from ρ by reversal of an independent component or a reversible part or if $\rho' = \rho^r$. Let $i, j \in \{1, \ldots, k\}$ and $u, v \in V_S$ with $B(u, G_i) \neq B(v, G_i)$. Since ρ is a simultaneous enumeration, we have $B(u, G_i) <_{\sigma_i} B(v, G_i) \Rightarrow B(u, G_j) \leq_{\sigma_j} B(v, G_j)$ and $B(u, G_i) >_{\sigma_i} B(v, G_j) \Rightarrow B(u, G_j) \geq_{\sigma_j} B(v, G_j)$. Let $\rho' = \rho^r$. Then we have $B(u, G_i) <_{\sigma'_i} B(v, G_j) \Rightarrow B(u, G_i) >_{\sigma_i} B(v, G_j) \Rightarrow B(u, G_j) \geq_{\sigma_j} B(v, G_j) \Rightarrow B(u, G_j) \leq_{\sigma'_j} B(v, G_j)$. Next let ρ' be obtained by a reversal of an independent component C in G_i. Then we have for $x, y \in V_S \cap V(C)$ that $B(x, G_i) = B(y, G_i)$. For $u, v \in V_S$ with $\{u, v\} \nsubseteq V(C)$ we have for $j \in \{1, \ldots, k\}$ that $B(u, G_j) \leq_{\rho_j} B(v, G_j) \Leftrightarrow B(u, G_j) \leq_{\rho'_j} B(v, G_j)$ and thus ρ' is also a simultaneous enumeration. Finally let ρ' be obtained by reversal of a reversible part $\mathcal{C}(B)$. For $u, v \in V_S$ with $\{u, v\} \nsubseteq V(B)$ we have as in the previous case for $j \in \{1, \ldots, k\}$ that $B(u, G_j) \leq_{\rho_j} B(v, G_j) \Leftrightarrow (u, G_j) \leq_{\rho'_j} B(v, G_j)$. For $u, v \in V(B)$ we have for $i \in \{1, \ldots, k\}$ with $B(v, G_i) \neq B(u, G_i)$ that $B(u, G_i) <_{\sigma_i} B(v, G_j) \Leftrightarrow B(u, G_i) >_{\sigma'_i} B(v, G_i)$. For $i, j \in \{1, \ldots, k\}$ with $B(v, G_i) \neq B(u, G_i)$ and $B(v, G_j) \neq B(u, G_j)$, we obtain $B(u, G_i) <_{\sigma'_i} B(v, G_i) \Rightarrow B(u, G_i) >_{\sigma_i} B(v, G_i) \Rightarrow B(u, G_j) \leq_{\sigma_j} B(v, G_j) \Rightarrow B(u, G_j) >_{\sigma'_j} B(v, G_j)$. We conclude that ρ' is also a simultaneous enumeration.

It remains to show that every simultaneous enumeration ρ' can actually be obtained from ρ or ρ^r by the provided reversals. By Lemma 8 we obtain ρ' from ρ or ρ^r by reversal of loose components. Without loss of generality assume ρ' can be obtained from ρ by reversal of loose components. First note that the order of two vertices

$u, v \in V_S$ by σ_i with $i \in \{1, \dots, k\}$ and $B(u, G_i) \neq B(v, G_i)$ is affected by the reversal of the component C in G_i containing u and v and by no other reversal. Assume we have two components C, C' of G_i, G_j oriented at a block B of S where C is reversed and C' is not. By Lemma 9 we have two vertices $u, v \in B$ such that $B(u, C) <_{\sigma_i} B(v, C)$ and $B(u, C') <_{\sigma_j} B(v, C')$. Since C is reversed and C' is not, we obtain $B(u, C) >_{\sigma'_i} B(v, C)$ and $B(u, C') <_{\sigma'_j} B(v, C')$ which contradicts ρ' being a simultaneous enumeration. This implies, that for every block B of S either all components oriented at B or none of them are reversed. If one of them is not loose, this implies they are all not contained in a reversible part. If they are all loose, then the reversal of all of them is just the reversal of the reversible part at B. Hence, we actually only reversed 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 Sect. 4.1 we provide a criterion for determining whether a given simultaneous enumeration ζ of $\mathcal{G}$ is realized by a simultaneous unit interval representation of $\mathcal{G}$. Namely, the criterion is the avoidance of a certain configuration in a partial vertex order of $\bigcup \mathcal{G}$ induced by ζ. In Sect. 4.2 we combine these findings to efficiently recognize simultaneous unit interval graphs. To this end, we search among all simultaneous enumerations for one that avoids the forbidden configuration by reversing reversible parts and independent components accordingly.

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 [14].

Lemma 11 *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 a graph H with $V(H) = V$ that contains $G_1, \dots, G_k$ as induced subgraphs and that has a fine enumeration σ such that for $i \in \{1, \dots, k\}$ the straight enumeration ζ_i is compatible with σ on V_i.*

Proof Given a simultaneous unit interval representation R of $\mathcal{G}$ that realizes ζ, one obtains H as the intersection graph of all unit intervals in R with a fine enumeration σ compatible to each ζ_i. On the other hand, a unit interval representation of H corresponding to a fine enumeration σ that is compatible to each ζ_i induces unit interval representations for $G_1, \dots, G_k$ that correspond to $\zeta_1, \dots, \zeta_k$ where S is represented in the same way. □

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 [28] to obtain

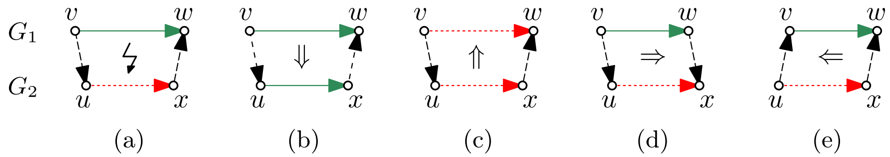

Fig. 5 **a**: The forbidden configuration of Corollary 13. **b**–**e**: The four implications of Corollary 14

four implications that can be used given only partial information on H and σ (as given by Lemma 11); see Fig. 5. For the figures in this section we use arrows to represent a partial order between two vertices. We draw them solid green if their endpoints are adjacent, red dotted if their endpoints are non-adjacent in some graph G_i, and black dashed if they may or may not be adjacent.

Proposition 12 *(Looges and Olariu [28]) A vertex order of a graph H is a fine enumeration if and only if for $v, u, w \in V(H)$ with $v <_\sigma u <_\sigma w$ and $vw \in E(H)$ we have $vu, uw \in E(H)$.*

Corollary 13 *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 $v \leq_\sigma u \leq_\sigma x \leq_\sigma w$ such that $vw \in E$ and $ux \notin E$.*

Proof The condition for three vertices provided by Proposition 12 (*3-vertex condition*) consists of two special cases of the condition for four vertices v, u, x, w of this corollary (*4-vertex condition*) where $v = u$ or $x = w$ while the other three vertices are distinct. On the other hand, if the 4-vertex condition is not met for v, u, x, w, then we have either $vx \in E$ or $vx \notin E$. In the first case the 3-vertex condition is violated by v, u, x and in the second case it is violated by v, x, w. Hence, the 3-vertex condition and the 4-vertex condition are equivalent. Therefore, by Proposition 12 a vertex order is a fine enumeration if and only the 4-vertex condition is satisfied. □

Corollary 14 *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 Fig. 5):*

(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 $\mathcal{G}$ be a sunflower graph with simultaneous enumeration $\zeta = (\zeta_1, \ldots, \zeta_k)$. 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 that $u \leq_\eta v$, 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.

Let $u, v \in V_S$ with $u \neq v$. For $i \in \{1, \ldots, k\}$ a (u, v)-*chain* of *size* $m \in \mathbb{N}$ in (G_i, ζ_i) is a sequence $(u = c_1, \ldots, c_m = v)$ of vertices in V_i with $c_1 <_i \cdots <_i c_m$

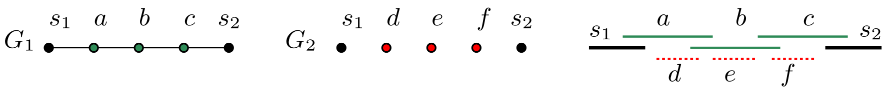

Fig. 6 A sunflower graph $\mathcal{G} = (G_1, G_2)$ with shared vertices s_1, s_2 (black, bold). Let ζ be the simultaneous enumeration realized by the given simultaneous proper interval representation. In (G_1, ζ_1) we have the (s_1, s_2)-chain $C = (s_1, a, b, c, s_2)$ of size 5 (green, solid). In (G_2, 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

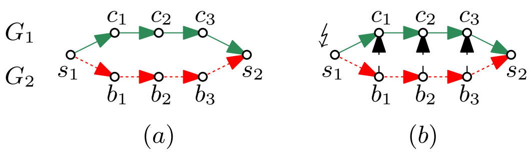

Fig. 7 **a**: A simultaneous enumeration with conflict. **b**: Result with added orders after scouting, starting at s_2 and finding the conflict in s_1

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 Fig. 6. 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_i 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_j has a (u, v)-bar of size l, then the distance between I_u, I_v is greater than $l - 2$. Hence, $\mathcal{G}$ has no conflict. The main result of this section is that the absence of conflicts is not only necessary, but also sufficient for ζ to be realized by a simultaneous unit interval representation.

Theorem 15 *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.*

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 Figs. 7, 8. 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 14 (iv). If there is a conflict, then it is found in this step. Otherwise, in a second sweep (*zipping*) from left to right, we greedily order the vertices by additionally respecting Corollary 14 (iii) to obtain a linear extension where both implications of Corollary 14 (iii), (iv) are satisfied. In the last step we decide which edges H has by respecting Corollary 14 (i), (ii).

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

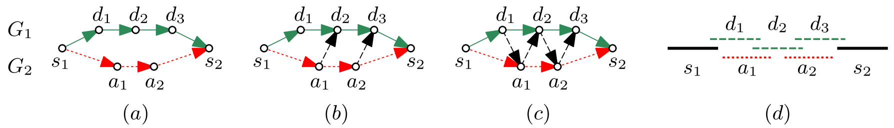

Fig. 8 **a**: A simultaneous enumeration without conflict. **b**: Result with added orders after scouting. **c**: Resulting linear order after zipping. Note that a_1 comes before d_2 in the linear order thanks to scouting. Choosing otherwise would imply a contradiction at s_2. **d**: Resulting unit interval representation for the sandwich graph

represented by the same interval in any simultaneous proper interval representation. Thus, we identify 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 $j \in \{1, \ldots, k\}$. 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 $i \in \{1, \ldots, k\}$. Note that u, v may be ordered even if they are indistinguishable in some input graphs.

We set $E = \left\{(u, v) \in \alpha \mid uv \in \bigcup_{i=1}^{k} E_i\right\}$ and $E^\times = \left\{(u, v) \in \alpha \mid uv \in E(\bigcup_{i=1}^{k} G_i^c)\right\}$ where G_i^c is the complement of G_i, for $i \in \{1, \ldots, k\}$. We call a partial order σ on V *left-closed* if we have

$$\forall v, w, u, x \in V\colon\ (vw \in E \ \wedge\ ux \in E^\times \ \wedge\ x \leq_\sigma w) \ \Rightarrow\ u <_\sigma v. \tag{3}$$

Note that a fine enumeration of a graph H with $G_1, \ldots, G_k$ as induced subgraphs is left-closed by Corollary 14 (iv). We describe the result of the scouting sweep with the following lemma.

Lemma 16 *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 induced by ζ.*

Proof 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 $i \in \{1, \ldots, k-1\}$ the i'th vertex of C and B must be ordered and distinct while the first vertex is shared; see Fig. 7.

Otherwise, the idea is to process the vertices from the right to the left and add for each of them the implied orders (each is considered as vertex x in the definition of left-closed). For each newly ordered pair (u, v) we maintain a pair of a chain and a bar with a common end vertex, which certifies a conflict if $u = v$. We will see that this is the only way the scouting can fail. Formally, we prove by induction, that for $0 \leq m \leq |V|$ there is a partial order $\sigma \supseteq \alpha$ on V and a set $X \subseteq V$ with $|X| = m$ such that:

(S1) For $w \in X$ and $v \in V \backslash X$ we have $v \not>_\sigma w$.

(S2) For $(u, v) \in \sigma \setminus \alpha^\star$ with $u \in V_i$ and $v \in V_j$ there is a vertex $s \in V_S$ such that $u \leq_\alpha s \leq_\alpha v$ or there are a (v, s)-chain $(v = c_l, \ldots, c_1 = s)$ in (G_j, ζ_j) and a (u, s)-bar $(u = b_l, \ldots, b_1 = s)$ in (G_i, ζ_i) with $c_t, b_t \in X$ and $b_t \leq_\sigma c_t$ for $1 \leq t < l$ with $l \geq 2$.

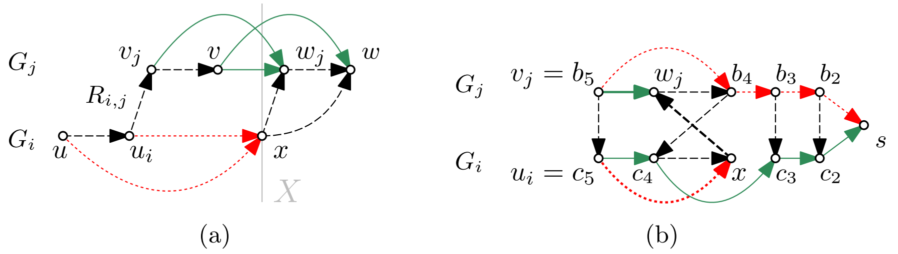

Fig. 9 On the left: The vertices u_i, v_j, w_j as derived from x and X'. We have $R_{i,j} = \{(u_i, v_j)\}$. On the right: The situation for $(v_j, u_i) \in \sigma \setminus \alpha^\star$. We have the (v_j, s)-bar (v_j, b_4, b_3, b_2, s) and the (u_i, s)-chain (u_i, c_4, c_3, c_2, s)

(S3) For $v, w \in V_j, u, x \in V_i$ with $w, x \in X$ as well as $v <_\sigma w$ and $u <_\sigma x$ such that $vw \in E_j$ and $ux \notin E_i$, we have $x \leq_\sigma w \Rightarrow u <_\sigma v$.

Property (S1) means that we actually go from the right to the left along α. Property (S2) ensures that we have pairs of chains and bars for every newly ordered pair of vertices. And Property (S3) ensures σ satisfies the left-closed Property 3 for $w, x \in X$.

We first show the statement for $m = 0$ with $\sigma = \alpha$ and $X = \emptyset$. We have that α is a partial order. For Property (S1) and Property (S3) there is nothing to show, since $X = \emptyset$. For Property (S2) observe that $\alpha \setminus \alpha^\star$ only contains tuples (u, v) obtained by transitivity, which requires some vertex $s \in V_S$ with $u \leq_i s \leq_j v$, where $u \in V_i, v \in V_j$.

Now assume that the statement is true for a partial order σ and a set $X \subsetneq V$. Since $X \neq V$ and σ is a partial order on V, there is a maximal element x in $V \setminus X$. We define $X' = X \cup \{x\}$ and obtain $|X'| = |X| + 1$.

We first consider σ restricted on every pair of graphs G_i, G_j and enhance it according to Property (S3) for X' to a partial order $\tau_{i,j}$ that satisfies all properties. Then we take the union of our obtained orders $\tau_{i,j}$ as τ and show it already is transitive. The remaining properties are then easily derived from the partial orders $\tau_{i,j}$ with $1 \leq i, j \leq k$.

For every pair of graphs G_i, G_j we only need to apply the implication of Property (S3) once if applied at a specific configuration. For $x \in V_i$, we set u_i to be the last vertex in V_i before x, that is not adjacent to x, i.e., we set $u_i = \max_\alpha \left(V_i \setminus (X' \cup N_{G_i}(x))\right)$; see Fig. 9a. We set w_j to be the first vertex in V_j with $x \leq_\sigma w_j$, i.e., we set $w_j = \min_\alpha\{w \in X' \cap V_j \mid x \leq_\sigma w\}$. We further set v_j to be the first vertex in $V_j \setminus X'$ that is adjacent to w_j, i.e., we set $v_j = \min_\alpha(N_{G_j}(w_j) \setminus X')$. Note that u_i, w_j, v_j may not exist. We define $R_{i,j} = \{(u_i, v_j)\}$ or $R_{i,j} = \emptyset$, if (u_i, v_j) does not exist. We set $\tau_{i,j}$ to be the transitive closure of $\{(u, v) \mid u, v \in V_i \cup V_j\} \cup R_{i,j}$. Set $X_{i,j} = X \cap (V_i \cup V_j)$ and $X'_{i,j} = X' \cap (V_i \cup V_j)$.

We first show the relations $\tau_{i,j}$ are partial orders. Since they are by definition reflexive and transitive, it remains to show they are antisymmetric.

$\tau_{i,j}$ is antisymmetric: For $1 \leq i, j \leq k$ it suffices to show $(v_j, u_i) \notin \sigma$ since (u_i, v_j) is the only tuple added before building the transitive closure and σ itself is transitive and antisymmetric. Assume $v_j \leq_h u_i$ with $h \in \{i, j\}$. This implies $i = j$ or $v_j \in V_S$ or $u_i \in V_S$. With Corollary 14 (iii) we obtain $w_j <_h x$ in contradiction to the definition of w_j. Hence, we can assume $(v_j, u_i) \in \sigma \setminus \alpha^\star$. By

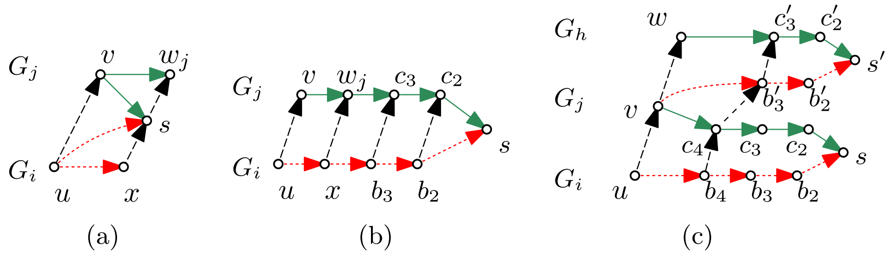

Fig. 10 **a** and **b**: Both cases for the construction of the (v, s)-chain and the (u, s)-bar to satisfy Property (S2) for a tuple $(u, v) \in \tau_{i,j} \backslash \sigma$. (c): The case of chains and bars in the proof of transitivity for τ

Property (S2) we then have that there is a vertex $s \in V_S$ such that there are a (u_i, s)-chain $(u_i = c_l, \dots, c_1 = s)$ and a (v_j, s)-bar $(v_j = b_l, \dots, b_1 = s)$ in (G_j, ζ_j) with $b_{l-1} \leq_\sigma c_{l-1}$ and $l \geq 2$; see Fig. 9b. Especially, we have $v_j <_j b_{l-1}$ and $u_i <_i c_{l-1}$ and $v_j b_{l-1} \in E_j$ and $u_i c_{l-1} \notin E_i$. With Corollary 14 (iii) we obtain $w_j \leq_j b_{l-1}$ and $c_{l-1} \leq_i x$. This yields $x <_\sigma w_j \leq_j b_{l-1} \leq_\sigma c_{l-1} \leq_i x$ which contradicts the antisymmetry of σ. We obtain that $\tau_{i,j}$ is antisymmetric. It is thus a partial order.

$(\tau_{i,j}, X'_{i,j})$ satisfies Property (S1): By choice of x Property (S1) is satisfied for $(\sigma, X'_{i,j})$ Note that for $(u, v) \in \tau_{i,j} \setminus \sigma$ we have that (u, v) is obtained from (u_i, v_j) by transitivity and thus $u \leq_{\tau_{i,j}} u_i <_i x$. Especially, we have $u \notin X'$. Therefore Property (S1) is satisfied for $(\tau_{i,j}, X'_{i,j})$ on $V_i \cup V_j$.

$(\tau_{i,j}, X'_{i,j})$ satisfies Property (S2): Let $u \in V_i$, $v \in V_j$ with $(u, v) \in \tau_{i,j} \backslash \sigma$; see Figs. 10a,b. Then we have $u \leq_\alpha u_i <_\sigma x$ and $v_j \leq_\alpha v$. We further have $v <_\sigma w_j$ since otherwise $u <_\sigma x \leq_\sigma w_j \leq_\sigma v$. By Corollary 14 (i),(ii) we obtain $v w_j \in E_j$ and $ux \notin E_i$. By Property (S2) there is either a vertex $s \in V_S$ such that $x \leq_\sigma s \leq_\sigma w_j$ or there are an (w_j, s)-chain $(w_j = c_l, \dots, c_1 = s)$ in (G_j, ζ_j) and an (x, s)-bar $(x = b_l, \dots, b_1 = s)$ in (G_i, ζ_i) with $c_t, b_t \in X$ and $b_t \leq_\sigma c_t$ for $1 \leq t < l$ with $l \geq 2$. In the first case we have $s \in X'_{i,j}$ and with Property (S1) we obtain $v <_\alpha s \leq_\alpha w_j$. We further have $u \leq_\alpha u_i <_\alpha x \leq_\alpha s$. By Corollary 14 (i),(ii) we obtain $vs \in E_j$ and $us \notin E_i$. In the second case $(v, w_j = c_l, \dots, c_1 = s)$ is a (v, s)-chain in (G_j, ζ_j) and $(u, x = b_l, \dots, b_1 = s)$ is a (u, s)-bar in (G_i, ζ_i) where we also have $w_j, x \in X'_{i,j}$ and $x \leq_\sigma w_j$. We therefore have that (v, s) is a (v, s)-chain of size 2 in (G_j, ζ_j) and that (u, s) is a (u, s)-bar of size 2 in (G_i, ζ_i). Therefore Property (S2) is satisfied for $\tau_{i,j}$.

$(\tau_{i,j}, X'_{i,j})$ satisfies Property (S3): Let $v, w \in V_j$, $u, x' \in V_i$ with $x', w \in X'$ as well as $v <_{\tau_{i,j}} w$ and $u <_{\tau_{i,j}} x'$ such that $vw \in E_j$ and $ux' \notin E_i$. Assume $x \leq_{\tau_{i,j}} w$. Since $\alpha \subseteq \tau_{i,j}$, and relation $\tau_{i,j}$ is antisymmetric, and α is a linear order on V_i and on V_j, we obtain $v <_\alpha w$ and $u <_\alpha x'$. As argued for Property (S2), if $(x', w) \in \tau_{i,j} \backslash \sigma$, then $x' \notin X'$. Therefore we have $x' \leq_\sigma w$. If $x' \neq x$, then we have $x', w \in X$ and thus $u <_\sigma v$ since (σ, X) satisfies Property (S3). Hence, assume $x' = x$; see Fig. 9a. By definition of v_j, w_j, u_i we have $v_j \leq_\alpha v <_\alpha w_j \leq_\alpha w$ and $u \leq_\alpha u_i <_\alpha x$. By Corollary 14 (i),(ii) we obtain $v w_j \in E_j$ and $ux \notin E_i$. This yields $u \leq_\alpha u_i \leq_{R_{i,j}} v_j \leq_\alpha v$ and thus $u \leq_{\tau_{i,j}} v$. It remains to show $u \neq v$. Assume otherwise. If $i = j$, then we have $u <_\alpha w, x$ with $u w_j \in E_i$ and $ux \notin E_i$.

This contradicts $x \leq_\sigma w_j$ (given by choice of w_j) by Corollary 14 since $\alpha_{|V_i} \subseteq \sigma$ is a fine enumeration of G_i. Hence, we have $i \neq j$ and $u = v \in V_S$. This implies $(x, w_j) \notin \alpha^\star$. By Property (S2) we have that there is a vertex $s \in V_S$ such that $x \leq_\sigma s \leq_\sigma w_j$ or there are a (w_j, s)-chain $(w_j = c_l, \dots, c_1 = s)$ in (G_j, ζ_j) and an (x, s)-bar $(x = b_l, \dots, b_1 = s)$ in (G_i, ζ_i) with $c_t, b_t \in X$ and $b_t \leq_\sigma c_t$ for $1 \leq t < l$ with $l \geq 2$. In the first case we have by Corollary 14 that $us \notin E_i$ and $vw_j \in E_j$. We then obtain the conflict consisting of the (u, s)-chain $(u = v, s)$ in E_j and the (u, s)-bar (u, s) in E_i. In the second case we obtain the conflict (C, B) with $(u = v, s)$-chain $C = (v, w_j = c_l, \dots, c_1 = s)$ in (G_j, ζ_j) and (u, s)-bar $B = (u, x = b_l, \dots, b_1 = s)$. Since we assumed there is no conflict we can conclude $u \neq v$.

Finally, we define $\tau = \bigcup_{1 \leq i,j \leq k} \tau_{i,j}$.

τ is antisymmetric:

Since τ is the union of the relations $\tau_{i,j}$ with $1 \leq i, j \leq k$ and those share pairwise at most tuples of a set V_l on which they coincide with α, there are no two distinct vertices $u, v \in V$ with $(u, v), (v, u) \in \tau$.

τ is transitive: Note that $\alpha \subseteq \tau_{i,j} \subseteq \tau$. Let $u \in V_i, v \in V_j, w \in V_h$ and $u \leq_\tau v \leq_\tau w$. If $u \leq_\alpha v$ or $v \leq_\alpha w$, then there is some $\tau_{l,m}$ with $u \leq_{\tau_{l,m}} v \leq_{\tau_{l,m}} w$ and thus $u \leq_\tau w$. Otherwise, we have $(u, v), (v, w) \in \tau \setminus \alpha^\star$ and Property (S2) is satisfied for $(u, v), (v, w)$ with regards to $\tau_{i,j}$ and $\tau_{j,h}$. First assume there is an $s \in V_S$ with $u \leq_\alpha s \leq_\alpha v$. Then if $s \leq_\alpha w$, we have $u \leq_\alpha w$ by transitivity of α and otherwise we have $w \leq_\alpha s \leq_\alpha v$, in contradiction to the antisymmetry of τ. Thus, we have $u \leq_\tau w$. Similarly, we obtain $u \leq_\tau w$ if there is an $s \in V_S$ with $v \leq_\alpha s \leq_\alpha w$.

Hence, assume we have for (u, v) a vertex $s \in V_S$ with a (u, s)-bar $(u = b_l, \dots, b_1 = s)$ in (G_i, ζ_i) and a (v, s)-chain $(v = c_l, \dots, c_1 = s)$ in (G_j, ζ_j) such that $b_{l-1} \leq_\tau c_{l-1}$ and $b_{l-1}, c_{l-1}, \in X'$; see Fig. 10. Further, we assume for (v, w) that there is a vertex $s' \in V_S$ with a (v, s)-bar $(v = b'_t, \dots, b'_1 = s')$ in (G_i, ζ_i) and a (w, s)-chain $(w = c'_t, \dots, c'_1 = s')$ in (G_h, ζ_h) such that $b'_{l-1} \leq_\tau c'_{t-1}$ and $b'_{t-1}, c'_{t-1} \in X'$. This yields $b_{l-1} \leq_\tau c_{l-1}$ and $b'_{l-1} \leq_\tau c'_{t-1}$ and by the definition of chains and bars we have $ub_{l-1} \notin \alpha$ and $vc_{l-1} \in \alpha$ and $vb'_{t-1} \notin \alpha$ and $wc'_{t-1} \in \alpha$. By Corollary 14 (iii) we have $c_{l-1} \leq_j b'_{t-1}$ and thus $b_{l-1} \leq_\tau c_{l-1} \leq_j b'_{t-1} \leq_\tau c'_{t-1}$. This means we have $b_{l-1} \leq_\tau c'_{t-1}$ and $u \leq_i b_{l-1}$ and $w \leq_h c'_{t-1}$ and $ub_{l-1} \notin \alpha$ and $wc'_{t-1} \in \alpha$. By Property (S2) for τ we obtain $u \leq_\tau w$. Hence, τ is transitive. We obtain that τ is a partial order.

(τ, X') satisfies Properties (S1),(S2),(S3): Since τ is the union of the relations $\tau_{i,j}$ with $1 \leq i, j \leq k$ and those satisfy Properties (S1),(S2),(S3), the partial order τ itself also satisfies Properties (S1),(S2),(S3).

We can conclude that there is a partial order $\tau \supseteq \alpha$ that satisfies Property (S3) for $X = V$ and is thus left-closed. ◻

By respecting the order obtained by scouting we avoid wrong decisions when greedily adding vertices to a linear order in the zipping step; see Fig. 8. Indeed, if the scouting did not yield a conflict, the zipping always succeeds.

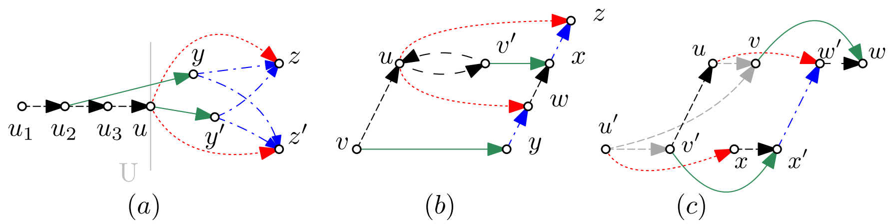

Fig. 11 **a** orders added during a zipping step. Vertices adjacent to U' (namely, y, y') come before those not adjacent to u in the corresponding graph G_i (namely, z, z'). **b** Proof of Observation (4). **c** Proof that τ is left-closed. Arrows for tuples in R are blue and dash dotted

Lemma 17 *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 The former is a special case of the latter. Hence, we only need to show that we obtain a suitable linear extension for a left-closed partial order $\sigma \supseteq \alpha$.

The idea is to process the vertices greedily from the left to the right and add for each of them the orders that are in a sense implied by Corollary 14(iii); see Fig. 11a. This ensures that the left-closed property is preserved. Formally, we prove by induction, that for $0 \leq m \leq |V|$ there is a partial order $\sigma \supseteq \alpha$ on V and a set $U \subseteq V$ with $|U| = m$ such that:

(Z1) For $v \in U$ and $w \in V \backslash U$ we have $v <_\sigma w$.
(Z2) σ is a linear order on U.
(Z3) For $u, v \in U$ and $w, x \in V$ with $vw \in E$, $ux \in E^\times$, we have $v \leq_\sigma u \Rightarrow w <_\sigma x$.
(Z4) σ is left-closed.

Property (Z1) means the processed vertices are to the left and Property (Z3) ensures that Corollary 14(iii) is satisfied. If Property (Z3) holds, then we say σ is *right-closed* on U.

Observe that the partial order τ provided by Lemma 16 is a suitable partial order for σ for $U = \emptyset$.

Now assume that the statement is true for a partial order σ and a set $U \subsetneq V$. Since $U \neq V$ and σ is a partial order on V, there is a minimal element u in $V \setminus U$. We define $U' = U \cup \{u\}$ and obtain $|U'| = |U| + 1$.

We denote the set of all tuples implied by Property (Z1) by Q, i.e., we define $Q = \{u\} \times (V \setminus U)$. We further define the set R to be the set of all tuples implied by Q with Property (Z3), i.e., we set $R = \{(w, x) \in (V \backslash U)^2 \mid \exists v \in U : v \leq_\sigma u \wedge vw \in E \wedge ux \in E^\times\}$. We finally set τ to be the transitive closure of $\sigma \cup Q \cup R$.

By choice of u, with no edge in $Q \cup R$ ending in U, and with $Q \cup R \subseteq \tau$, the Properties (Z1),(Z2),(Z3) are satisfied for τ. Note that we still have to show that τ is antisymmetric.

We will make use of the following observation:

$$\forall (y, w), (x, z) \in R : w \not\leq_\sigma x \tag{4}$$

To see this, assume there are $(y, w), (x, z) \in R$ with $w \leq_\sigma x$; see Fig. 11b. By definition of R there are $v, v' \in V$ such that $vy \in E$ and $uw \in E^\times$ and $v \leq_\sigma u$ as well as $v'x \in E$ and $uz \in E^\times$ and $v' \leq_\sigma u$. Since σ is left-closed, we obtain $u <_\sigma v'$, a contradiction.

τ **is antisymmetric:** Assume there is a cycle C in the graph $(V, \sigma \cup Q \cup R)$. By Property (Z2) and with no edge in $\tau \setminus \sigma$ ending in U, we know that C contains no edge of Q. Assume there are two edges $(y, w), (x, z) \in R\cap$. Without loss of generality, let no other edge of R lie between (y, w) and (x, z) on R. Then by transitivity of σ, we have $w \leq_\sigma x$ in contradiction to Observation (4). Note that this argument also holds for $w = x$ since σ is reflexive. Hence, C contains at most one edge in R. On the other hand, C must contain an edge not in σ, since σ is a partial order. Thus, C contains exactly one edge (y, w) in R and by transitivity of σ we obtain $w \leq_\sigma y$ in contradiction to Observation (4) with $(x, z) = (y, w)$. We conclude that τ is antisymmetric and thus a partial order.

τ **is left-closed** Let $v, w, u', x \in V$ with $vw \in E$ and $u'x \in E^\times$ and $x \leq_\tau w$. If $u' \in U'$ or $v \in U'$, then we have $u' <_\tau v$ or $v <_\tau u'$ by Properties (Z1),(Z2). The latter case contradicts Property (Z3). Especially, we have $u' <_\tau v$ as desired. Otherwise, we have $u', v \in V \setminus U'$ and thus also $x, w \in V \setminus U'$ since $v \leq_\alpha w$ and $u' \leq_\alpha x$. Then there must be a path in $(V, \sigma \cup R)$ from x to w. If $x \leq_\sigma w$, then we obtain $u' <_\sigma v$ with Property (Z4) of σ. Otherwise, we obtain with Observation (4) that there are $x', w' \in V$ with $x \leq_\sigma x' \leq_R w' \leq_\sigma w$; see Fig. 11c. From $(x', w') \in R$ we obtain a vertex $v' \in U$ such that $v' \leq_\sigma u$ and $v'x' \in E$. Since σ is left-closed, we obtain $u' <_\sigma v'$ and $u <_\sigma v$. This yields $u' <_\sigma v$. We conclude that there is a partial order $\tau \supseteq \alpha$ that satisfies Properties (Z2), (Z4) for $U = V$ and is thus linear and left-closed. □

Finally, we construct a graph $H = (V, E')$ for which the obtained linear order τ is a fine enumeration. We do so by adding edges in accordance with Corollary 14 (i).

Lemma 18 *Let $\mathcal{G}$ 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.*

Proof If we have such a graph H, then τ is left-closed by Corollary 14 ((iv)). Hence, let τ be left-closed. Then we set $E' = \{ux \in V^2 \mid \exists vw \in E \colon v \leq_\tau u <_\tau x \leq_\tau w\}$. Clearly, we have $E \subseteq E'$. On the other hand, an edge $ux \in E' \cap E^\times$ would contradict τ being left-closed and an edge $ux \in E' \cap E^{\times r}$ would contradict transitivity of τ. Let $ux \in E'$ with $u \leq_\tau x$. Let $y \in V$ with $u <_\tau y <_\tau x$. By definition of E' there are $v, w \in V$ with $v \leq_\tau u <_\tau y <_\tau x \leq_\tau w$. We obtain $uy, yx \in E'$. Hence, for $v \in V$ the neighborhood $N_H(v)$ is consecutive in τ, and thus τ is a fine enumeration of H. □

Combining Lemmas 11, 16, 17 and 18 we obtain Theorem 15.

Theorem 15 *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 10, 15 we can now efficiently recognize simultaneous unit interval graphs. In this section, we explain how a polynomial running time can be achieved. In Sect. 4.3 we then go into detail about how to achieve a specific running time.

Theorem 19 *Given a sunflower graph $\mathcal{G}$, it can be decided in polynomial time whether $\mathcal{G}$ is a simultaneous unit interval graph.*

Proof By Lemma 3, we can assume that $\bigcup \mathcal{G}$ 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 15, 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 strict conflict. In that case η^r also has no strict conflict. With Theorem 10 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 (u, v)-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 $i \in \{1, \dots, k\}$, $u, v \in V_S$ and each of the four combinations of reversal decisions (reverse or do not reverse) for the corresponding components $C, D \subseteq G_i$ 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 variable that represents whether it is reversed or not (for every other component we have a constant "decision").

For every combination of two reversal decisions that yields a conflict we add a clause that excludes this combination. Note that there are at most $|V|^2$ such pairs. 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 Sect. 4.1. □

4.3 Recognizing Sunflower Unit Interval Graphs in $O(|V| \cdot |E|)$ Time

For a more efficient algorithm, we restrict the conflicts that we have to consider. Let $\mathcal{G}$ be a sunflower proper interval graph with simultaneous enumeration ζ. We call a (u, v)-chain $(u = c_1, \dots, c_m = v)$ in (G_i, ζ_i) *strict* if for $1 < j < m$ we have $c_j \notin V_S$ and it is minimal in the sense that there is no (u, v)-chain with a size smaller than m. For bars we consider a greedy construction as follows. For a vertex v in G_i and a subgraph G' of G_i the *right stop* of v with regards to $G' \in \{G_i, S\}$ is the leftmost vertex v' in G' with $v <_i v'$ that is not adjacent to v. We call a (u, v)-bar $B = (u = b_1, \dots, b_m = v)$ in (G_i, ζ_i) *strict* if, for $2 < l < m$, b_l is in $G_i \setminus S$ and the right stop of b_{l-1} w.r.t. G_i and $v = b_m$ is the right stop of b_{m-1} w.r.t. S. We call a conflict (C, B) *strict* if C and B are strict. It suffices to exclude strict conflicts.

Lemma 20 *Let $\mathcal{G}$ be a simultaneous proper interval graph that has no strict conflict for some simultaneous enumeration ζ. Then $\mathcal{G}$ has no conflict for ζ.*

Proof We show that if there is a conflict for a simultaneous enumeration ζ, then there is a strict conflict for ζ. Let (C, B) be a (u, v)-conflict for ζ with minimum size of C among all chains in conflicts and minimum number of vertices in S between u,v. Let $C = (u = c_1, \ldots, c_m = v)$ and $B = (u = b_1, \ldots, b_m = v) \subseteq G_i$.

Assume C is not strict. Then by the choice of (C, B), there is a $1 < j < m$ with $s = c_j \in V_S$. We obtain the chains $(u = c_1, \ldots, c_j = s)$ and $(s = c_j, \ldots, c_m = v)$ with sizes $j, m - j + 1 < m$. Since C has minimum size among all chains in conflicts, neither $(u = b_1, \ldots, b_{j-1}, s)$ nor $(s, b_{j+1} \ldots, b_m = v)$ may be a bar. Thus, s is adjacent to the three independent vertices b_{j-1}, b_j, b_{j+1}, i.e., G_i contains $K_{1,3}$ as an induced subgraph. This contradicts G_i being a proper interval graph. Hence, C is strict.

If B is strict, we are done. Hence, assume B is not strict. We obtain a strict conflict with a simple exchange argument. Namely, we can iteratively replace for $2 < l < m$ b_l by the right stop of b_{l_1} w.r.t. G_i and finally replace b_m with the right stop of b_{m-1} w.r.t. S. First observe that each iteration results in a new (u, v')-bar of the same size with $v' \leq_i v$ by the definition of right stops. Let B' be the resulting (u, v')-bar $(u = b'_1, \ldots, b'_m)$. Assume that $(u, c_2, \ldots, c_{m-1}, v')$ is not a chain. Then we obtain a smaller (u, v')-chain by removing all elements to the right of v'. Hence, we have a (u, v')-chain and the (u, v')-conflict (C', B') in contradiction to the choice of C. It follows that $(u, c_2, \ldots, c_{m-1}, v')$ is a chain and by the choice of C it follows that $C = C'$ and $v' = v$. It remains to show that B' is strict.

Assume there is an element $s = b'_j$ of B' in S with $1 < j < m$. We obtain the bars $(u = b'_1, \ldots, b'_j = s)$ and $(s = b'_j, \ldots, b_m = v)$ with sizes $j, m - j + 1 < m$ and $u < s < v$. The latter implies that there are $1 \leq p < q \leq m$ such that $(u = c_1, \ldots, c_p, s)$ and $(s, b_q \ldots, b_m = v)$ are chains. Since C has minimum size among all chains in conflicts, they may not form conflicts with the bars $(u = b'_1, \ldots, b'_j = s)$ and $(s = b'_j, \ldots, b_m = v)$. We get $q \leq i \leq p$, a contradiction. Thus, B' and (C, B') are strict. □

We can recognize sunflower unit interval graphs in $O(|V(\bigcup\mathcal{G})| \cdot |E(\bigcup\mathcal{G})|)$ time by considering only strict conflicts. Note that $|V(\bigcup\mathcal{G})|$ and $|E(\bigcup\mathcal{G})|$ count vertices and edges of the shared graph only once.

Lemma 21 *Given a sunflower proper interval graph $\mathcal{G}$ with $\bigcup\mathcal{G}$ connected, we can decide in $O(|V(\bigcup\mathcal{G})| \cdot |E(\bigcup\mathcal{G})|)$ time, whether $\mathcal{G}$ has a simultaneous enumeration η for which $\mathcal{G}$ has no conflicts. In that case η can be computed in the same time.*

Proof We consider $\mathcal{G}$ given as graph $G = (V, E) = \bigcup\mathcal{G}$ where every vertex in V_S labeled with S and every other vertex $v \in V_i$ labeled with G_i. Without loss of generality we can assume that every input graph G_i contains a private vertex $v \notin S$, since otherwise $G_i = S$ will be represented correctly by representing any input graph correctly. With G being connected, we then have $k \leq |V| \leq |E|$.

By Theorem 6 we can test in $O(\sum_{i=1}^{k} |E_i|)$ time whether G is a sunflower proper interval graph and if so obtain a simultaneous enumeration ζ. Otherwise, we can reject.

Note that if a simultaneous enumeration η has no conflict, then also η^r has no conflict. With Theorem 10 we obtain that η or η^r is obtained from ζ by reversals of reversible parts and independent components By Lemma 20, we only need to consider strict conflicts for such simultaneous enumerations.

We compute all independent components and reversible parts as follows. We start by computing the loose components. We call a component of S that is a clique a *candidate block*. We can compute all candidate blocks in $O(|V_S| + |E_S|)$ time. Note that loose components contain a single block of S, which is thus a candidate block.

To efficiently compute the loose components of an input graph G_i, we gather their components that are connected in $G_i \setminus S$ as follows. We use a graph traversal (e.g. a DFS) starting at every (non-visited) vertex in $G_i \setminus S$ that stops at the vertices of S. If a single block B of S is reached and B is a candidate block, all reached vertices of $G_i \setminus S$ are associated to B. Otherwise, all reached vertices belong to a non-loose component, and we mark each reached candidate block. Note that at least one block of S is reached since otherwise G is not connected. The loose components of G_i are then exactly those graphs induced by all vertices associated to a non-marked candidate block. They can thus be computed in $O(|V_i \setminus V(S)| + |E_i \setminus E(S)|)$ time. The independent components and the reversible parts can now be computed by deciding for each loose component C whether it is oriented at its candidate block B. This can be done in $O(|V(C \setminus S)| + |E(C \setminus S)|)$ time. Thus, the independent components and reversible parts can be computed in $O(|V_S| + |V_E| + |V \setminus V_S| + |E \setminus E_S|) = O(|V| + |E|)$ total time.

Our goal is to construct a 2-SAT formula where the variables describe reversal decisions and the clauses forbid reversal decisions and pairs of reversal decisions that result in strict conflicts. To this end, we compute all strict chain lengths and then efficiently compute the strict bar lengths for all reversal decisions.

Since strict chains are shortest paths in $\bigcup \mathcal{G}$ that do not contain shared vertices except for the start and the end, we obtain all sizes of strict chains with start and end in S by breadth-first-searches in $\bigcup \mathcal{G}$ starting at each vertex $u \in V_S$ and stopping at any reached vertex $v \in V_S$ with a total running time in $O(|V_S| \cdot (|E \setminus E_S|))$. Since only the smallest chain between any vertex pair is relevant, we store for each pair of shared vertices u,v the minimum obtained strict (u, v)-chain size in a matrix. Note that these sizes are not affected by reversal decisions and apply for every considered simultaneous enumeration.

For strict bars, the idea is similar. We want to start at every vertex $u \in V_S$ and, for $1 \leq i \leq k$, we iteratively add the next right stop in G_i according to the simultaneous enumeration ζ and note the current length at its right stop in S. However, this would provide the strict bar lengths only for one simultaneous enumeration. We fix this as follows. First note that for two shared vertices u, v, it depends only on the reversal of the components of u and v in G_i whether there is a strict (u, v)-bar of a certain size, since components in-between contribute a maximum independent set irrespective of whether they are reversed or not (and thus the same number of bar-elements). We create a copy of each reversible part and of each independent component and insert them reversed at the same place in ζ, resulting in a partial order σ_i' for each input graph where each vertex has either only one successor or exactly two non-adjacent successors. Label each vertex of a reversible part or independent component with that component and one bit telling whether it is from the original component or its reversed copy. This can be done in $O(\sum_{i=1}^{k}(|V_i|+|E_i|))$ time.

We then construct for each input graph G_i a directed graph $N_i=(V_N,E_N)$ where V_N is the set of vertices of ζ after adding all reversed copies and $E_N=\{(u,v)\in\binom{V}{2}\mid v$ is the right stop of u w.r.t. G_i or $S\}$. This can be done in $O(\sum_{i=1}^{k}(|V_i|+|E_i|))$ total time by iterating for each vertex u of G_i over its (adjacent) successors in σ_i' until finding one or two successors in G_i that are non-adjacent to u. For the right stops w.r.t. S, instead of iterating over all neighbors, we start iterating at the last encountered right stop(s) w.r.t. S to ensure that we visit every vertex in V_S at most once for each input graph G_i.

Then, for every input graph G_i, start a BFS at every shared vertex u in N_i that stops at every shared node v. This can be done in $O(|V_S|\cdot|E\setminus E_S|+|E_S|)$ total time. At every reached shared node v, compare the size of the implied strict (u,v)-bar, which is the current depth, with the strict (u,v)-chain size. If the bar size is not smaller, we have to avoid this strict conflict. We do this by adding a clause that forbids u and v being reversed as stated by their labels. If only one of them has a label, we forbid that reversal decision, and if none of them has a label, then we can reject. Note that the 2-SAT formula contains at most $O(|V_S|\cdot|E\setminus E_S|+|E_S|)$ clauses. By construction, if the 2-SAT formula is not satisfiable, then each simultaneous enumeration has a strict conflict. Otherwise we can make the reversal decisions according to our formula and ensured for every shared vertex that it is not the start of a strict conflict. Hence, this yields a simultaneous enumeration without strict conflicts. Note that a 2-SAT formula can be solved in linear time [29]. □

For the construction of the simultaneous representation, we follow the proofs in Sect. 4.1.

Lemma 22 *Let $\mathcal{G}$ be a simultaneous proper interval graph that has no conflicts for a given simultaneous enumeration η. Then we can compute a simultaneous unit interval representation of $\mathcal{G}$ that realizes η in $O(|V(\bigcup\mathcal{G})|\cdot|E(\bigcup\mathcal{G})|)$.*

Proof Let $G=(V,E)=\bigcup\mathcal{G}$. We first construct a left-closed partial order τ that extends η by following the inductive proof of Lemma 16. We represent the iteratively constructed partial order σ with an edge-minimal directed acyclic graph G' whose transitive closure is σ. Initially, we set G' to be the graph corresponding to the transitive reduction of $\alpha^\star$ (the union of the partial orders on $V_1,\dots,V_k$ corresponding to $\zeta_1,\dots,\zeta_k$). In each step, we need to find the next processed vertex x and the corresponding vertices u_i, v_j, w_j for $w\in V_i$ and $1\le j\le k$. After that, we can just add (u_i,v_j) for $1\le j\le k$ to G'.

To find the next maximal vertex x in constant time, we keep track of the outdegree of each vertex in $G\setminus X$ and maintain a list of vertices with outdegree 0, from which we choose $x\in V\setminus X$ arbitrarily. In each step, we update the outdegree for all neighbors of x in G'. The total running time is then linear in the size of the final graph G' which is in $O(k|V|+|E|)$ since we will only add at most k edges per processed vertex x.

Consider the situation where we want to compute $w_j(x)\in V_j$ for the currently processed vertex $x\in V_i$. The successor x' of x in V_i was already processed and for the corresponding vertex $w_j(x')$ we have $x\le_\sigma w_j(x')$. We can thus compute $w_j(x)$

as $\min_\alpha\{w_j(x'), w \in X \mid xw \in G'\}$ in time linear in $|N_{G'}(x) \cap V_j|$. In total this can be done in $O(k|V| + |E|)$ time.

For the computation of u_i and v_j we can just precompute for each vertex y the first adjacent and last non-adjacent vertex before y in the corresponding input graph with a total running time in $O(\sum_{i=1}^{k}(|V| + |E|))$. Note that we indeed only add at most k edges $(u_i, v_1), \ldots, (u_i, v_k)$ to G' for x.

Next we compute a linear extension τ of σ as in the proof of Lemma 17. A minimal vertex u can be found analogously to a maximal vertex x before by keeping track of the indegree. By adding the vertices of U to a list, we do not need to actually add edges of the form (u, v) with $v \in V \setminus U$ to G'. Instead of adding all edges of R to G', it suffices to compute for each $1 \leq j \leq k$ a maximal vertex $v_j \leq_\sigma u$ and to add the corresponding edge (w_j, x_i) with maximal w_j such that v_j, w_j are adjacent and x_i is the right stop of u w.r.t. G_i. This can be done analogously to finding (u_i, v_j) in the construction of the left-closed partial order.

We finally follow the proof of Lemma 18 to decide adjacency between vertices of different graphs G_i, G_j. This can be done in linear time by going from left to right along our linear order of V as follows. We keep track of the last vertex w adjacent to any vertex visited so far, including the current vertex v. We then set v to be adjacent to w and to all vertices between v and w. This takes $O(|V|^2)$ time in total. We obtain a fine enumeration, from which a unit interval representation of a graph H that has $G_1, \ldots, G_k$ as induced subgraphs can be obtained in linear time. This yields a simultaneous unit interval representation of $\mathcal{G}$. ◻

As a consequence of Lemmas 3, 21 and 22 we obtain the following theorem.

Theorem 23 *Given a sunflower graph $\mathcal{G}$, we can decide in $O(|V(\bigcup \mathcal{G})| \cdot |E(\bigcup \mathcal{G})|)$ time, whether $\mathcal{G}$ is a simultaneous unit interval graph. If it is, then we also provide a simultaneous unit interval representation in the same time.*

5 General Simultaneous Proper and Unit Interval Graphs

In this section we consider the simultaneous representation problem for proper and unit interval representations without the restriction to sunflower graphs. We show that, if the number k of input graphs is part of the input, then these problems are NP-complete. Our reductions are similar to the simultaneous independent work of Bok and Jedličková [19] for simultaneous interval graphs.

Theorem 24 *Recognizing simultaneous proper interval graphs is NP-complete.*

Proof The problem is clearly in NP, as we can guess the order of the endpoints of the intervals in a simultaneous representation and verify (in polynomial time) whether the resulting representation is a simultaneous proper interval representation of the input graphs.

For the NP-hardness, we present a reduction from the NP-hard problem BETWEENNESS [30] which, given a ground set A and a set $\mathcal{T} \subseteq A \times A \times A$ of triplets of A asks whether there exists a linear order σ of A such that for any triple $(a, b, c) \in \mathcal{T}$, we have $a <_\sigma b <_\sigma c$ or $c <_\sigma b <_\sigma a$. We call such an order σ a *betweenness order*.

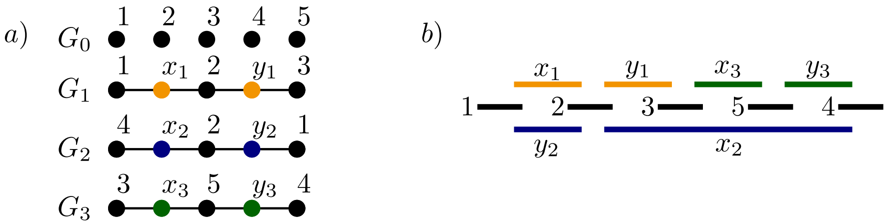

Fig. 12 For BETWEENNESS instance $(A, \mathcal{T})$ with $A = \{1, 2, 3, 4, 5\}$ and $\mathcal{T} = \{(1, 2, 3), (4, 2, 1), (3, 5, 4)\}$ a solution is $\sigma = 1 <_\sigma 2 <_\sigma 3 <_\sigma 5 <_\sigma 4$. (a) The simultaneous graph $\mathcal{G}$ constructed from $(A, \mathcal{T})$. (b) A simultaneous proper interval representation of $\mathcal{G}$

Let $(A, \mathcal{T})$ with $\mathcal{T} = \{T_1, \dots, T_k\}$ be an instance of BETWEENNESS. We construct a simultaneous graph consisting of $k + 1$ graphs $G_0, \dots, G_k$; see Fig. 12. The graph $G_0 = (A, \emptyset)$ contains all elements of A as vertices but no edges. For each triple $T_i = (a_i, b_i, c_i)$, we define the graph G_i as an induced path $a_i x_i b_i y_i c_i$ where x_i and y_i are new vertices. We set $\mathcal{G} = (G_0, G_1, \dots, G_k)$ and claim that $\mathcal{G}$ has a simultaneous proper interval representation $\mathcal{R}$ if and only if $(A, \mathcal{T})$ admits a betweenness order σ.

If $\mathcal{R} = (R_0, R_1, \dots, R_k)$ is a simultaneous interval representation of $\mathcal{G}$, then the representation R_0 defines a linear order σ of A. The fact that R_i is a proper interval representation of an induced path guarantees that b_i is positioned between a_i and c_i in σ for $i = 1, \dots, k$. Therefore σ is betweenness order for $(A, \mathcal{T})$.

Conversely, if σ is a betweenness order of $(A, \mathcal{T})$, we use this order to define a corresponding representation R_0 of G_0. For each triple $T_i = (a_i, b_i, c_i)$, due to the betweenness property, we can add intervals representing x_i and y_i such that we obtain a proper interval representation R_i of R. Altogether, this yields a simultaneous proper interval representation $\mathcal{R} = (R_0, R_1, \dots, R_k)$.

NP-hardness follows, since the instance $\mathcal{G}$ can be constructed in polynomial time from $(A, \mathcal{T})$. □

Theorem 25 *Recognizing simultaneous unit interval graphs is NP-complete.*

Proof The problem is in NP. Namely, we can guess the order of the intervals in the representation of each input graph. Afterwards, a unit interval can be described as the solutions of a straightforward linear program [5].

For the NP-hardness we employ a similar reduction as in the case of proper interval graphs in the proof of Theorem 24. The key difference is that, while the vertices in A can easily be represented as unit intervals, the vertices x_i and y_i may span several vertices of A, and can hence generally not be represented as unit intervals.

We instead replace x_i and y_i by a sequence of vertices $x_i^1, \dots, x_i^{2n}$ and $y_i^1, \dots, y_i^{2n}$. For each $j = 1, \dots, 2n - 1$, there is a graph G_i^j with $V(G_i^j) = \{a_i, b_i, c_i, x_i^j, y_i^j\}$ and edges $x_i^j x_i^{j+1}$ as well as $y_i^j y_i^{j+1}$. The edges $a_i x_i^j$, $c_i y_i^j$ are present only for $j = 1$ and the edges $x_i^{j+1} b_i$, $y_i^{j+1} b_i$ are present only for $j = 2n - 1$; see Fig. 13. Observe that this construction ensures that the vertices $x_i^1, \dots, x_i^{2n}$ all lie between a_i and b_i, and likewise $y_i^1, \dots, y_i^{2n}$ lie between b_i and c_i. The graph G_i^{2n-1} further ensures that they lie on different sides of b_i, i.e., again a simultaneous representation determines a betweenness order.

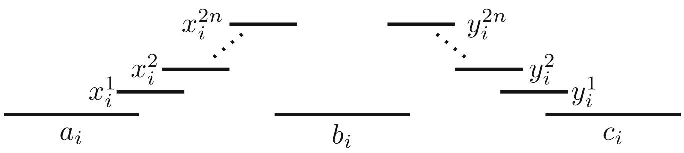

Fig. 13 Illustration of the hardness proof for recognition of simultaneous unit interval graphs

Moreover, the vertices $x_i^1, \dots, x_i^{2n}$ can be put arbitrarily close together or stretched to cover any distance less than $2n$, since the only requirement is that consecutive vertices intersect each other. Thus for any betweenness order of the vertices in A one can construct a corresponding simultaneous unit interval representation of the graphs $G_i, i = 0, \dots, k$ and G_i^j for $i = 1, \dots, k$, $j = 1, \dots, 2n - 1$. $\square$

6 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.

Funding Open Access funding enabled and organized by Projekt DEAL. This work was funded 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".

Data Availability Data sharing not applicable to this article as no datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

1. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs Annals of Discrete Mathematics. Elsevier, Amsterdam (2004)
2. Spinrad, J.P.: Efficient Graph Representations. Fields Institute Monographs, vol. 19. AMS, Providence (2003)
3. Jampani, K., Lubiw, A.: The simultaneous representation problem for chordal, comparability and permutation graphs. J. Graph Algorithms Appl. **16**(2), 283–315 (2012). https://doi.org/10.7155/jgaa.00259
4. Goldberg, P.W., Golumbic, M.C., Kaplan, H., Shamir, R.: Four strikes against physical mapping of DNA. J. Comput. Biol. **2**(1), 139–152 (1995). https://doi.org/10.1089/cmb.1995.2.139
5. Klavík, P., Kratochvíl, J., Otachi, Y., Rutter, I., Saitoh, T., Saumell, M., Vyskočil, T.: Extending partial representations of proper and unit interval graphs. Algorithmica **77**(4), 1071–1104 (2017). https://doi.org/10.1007/s00453-016-0133-z
6. Chaplick, S., Fulek, R., Klavík, P.: Extending partial representations of circle graphs. J. Graph Theory **91**(4), 365–394 (2019). https://doi.org/10.1002/jgt.22436
7. Bläsius, T., Rutter, I.: Simultaneous PQ-ordering with applications to constrained embedding problems. ACM Trans. Algorithms **12**(2), 16–11646 (2015). https://doi.org/10.1145/2738054
8. Bläsius, T., Kobourov, S.G., Rutter, I.: Simultaneous embedding of planar graphs. In: Tamassia, R. (ed.) Handbook on Graph Drawing and Visualization, pp. 349–381. Chapman and Hall/CRC (2013)
9. Brass, P., Cenek, E., Duncan, C.A., Efrat, A., Erten, C., Ismailescu, D.P., Kobourov, S.G., Lubiw, A., Mitchell, J.S.: On simultaneous planar graph embeddings. Comput. Geom. **36**(2), 117–130 (2007). https://doi.org/10.1016/j.comgeo.2006.05.006
10. Gassner, E., Jünger, M., Percan, M., Schaefer, M., Schulz, M.: Simultaneous graph embeddings with fixed edges. In: Fomin, F.V. (ed.) Graph-Theoretic Concepts in Computer Science, 32nd International Workshop, WG. Lecture Notes in Computer Science, vol. 4271, pp. 325–335. Springer, (2006). https://doi.org/10.1007/11917496_29. https://doi.org/10.1007/11917496_29
11. Schaefer, M.: Toward a theory of planarity: Hanani-tutte and planarity variants. J. Graph Algorithms Appl. **17**(4), 367–440 (2013). https://doi.org/10.7155/jgaa.00298
12. Angelini, P., Da Lozzo, G., Neuwirth, D.: On some $\mathcal{NP}$-complete SEFE problems. In: Pal, S.P., Sadakane, K. (eds.) Algorithms and Computation: 8th International Workshop, WALCOM 2014, pp. 200–212. Springer, (2014). https://doi.org/10.1007/978-3-319-04657-0_20
13. Estrella-Balderrama, A., Gassner, E., Jünger, M., Percan, M., Schaefer, M., Schulz, M.: Simultaneous geometric graph embeddings. In: Hong, S.-H., Nishizeki, T., Quan, W. (eds.) Graph Drawing: 15th International Symposium, GD 2007, Sydney. Revised Papers, pp. 280–290. Springer, (2008).https://doi.org/10.1007/978-3-540-77537-9_28
14. Golumbic, M.C., Kaplan, H., Shamir, R.: Graph sandwich problems. J. Algorithms **19**(3), 449–473 (1995). https://doi.org/10.1006/jagm.1995.1047
15. Roberts, F.S.: Indifference graphs. In: Harary, F. (ed.) Proof Techniques in Graph Theory, pp. 139–146. Academic Press, New York (1969)

16. McConnell, R.M., Nussbaum, Y.: Linear-time recognition of probe interval graphs. SIAM J. Discret. Math. **29**(4), 2006–2046 (2015). https://doi.org/10.1137/130930091
17. Nussbaum, Y.: Recognition of probe proper interval graphs. Discret. Appl. Math. **167**, 228–238 (2014). https://doi.org/10.1016/j.dam.2013.11.013
18. Jampani, K.R., Lubiw, A.: Simultaneous interval graphs. In: Cheong, O., Chwa, K., Park, K. (eds.) Algorithms and Computation - 21st International Symposium, ISAAC. Lecture Notes in Computer Science, vol. 6506, pp. 206–217. Springer, (2010). https://doi.org/10.1007/978-3-642-17517-6_20
19. Bok, J., Jedličková, N.: A note on simultaneous representation problem for interval and circular-arc graphs. arXiv preprint arXiv:1811.04062 (2018)
20. Deng, X., Hell, P., Huang, J.: Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. SIAM J. Comput. **25**(2), 390–403 (1996). https://doi.org/10.1137/S0097539792269095
21. Skrien, D.J.: A relationship between triangulated graphs, comparability graphs, proper interval graphs, proper circular-arc graphs, and nested interval graphs. J. Graph Theory **6**(3), 309–316 (1982). https://doi.org/10.1002/jgt.3190060307
22. de Figueiredo, C.M.H., Meidanis, J., de Mello, C.P.: A linear-time algorithm for proper interval graph recognition. Inf. Process. Lett. **56**(3), 179–184 (1995). https://doi.org/10.1016/0020-0190(95)00133-W
23. Heggernes, P., Van't Hof, P., Meister, D., Villanger, Y.: Induced subgraph isomorphism on proper interval and bipartite permutation graphs. Theoretical Computer Science **562**, 252–269 (2015). https://doi.org/10.1016/j.tcs.2014.10.002
24. Booth, K.S., Lueker, G.S.: Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms. J. Comput. Syst. Sci. **13**(3), 335–379 (1976). https://doi.org/10.1016/S0022-0000(76)80045-1
25. Booth, K.S.: PQ tree algorithms. PhD thesis, University of California, Berkeley (1975)
26. Roberts, F.S.: Representations of indifference relations. PhD thesis, Department of Mathematics, Stanford University (1968)
27. Hell, P., Shamir, R., Sharan, R.: A fully dynamic algorithm for recognizing and representing proper interval graphs. SIAM J. Comput. **31**(1), 289–305 (2002). https://doi.org/10.1137/S0097539700372216
28. Looges, P.J., Olariu, S.: Optimal greedy algorithms for indifference graphs. Comput. & Math. Appl. **25**(7), 15–25 (1993). https://doi.org/10.1016/0898-1221(93)90308-I
29. Aspvall, B., Plass, M.F., Tarjan, R.E.: A linear-time algorithm for testing the truth of certain quantified boolean formulas. Inf. Process. Lett. **8**(3), 121–123 (1979). https://doi.org/10.1016/0020-0190(79)90002-4
30. Opatrny, J.: Total ordering problem. SIAM J. Comput. **8**(1), 111–114 (1979)