FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Nikola Jedličková

Algorithmic aspects of intersection-defined graph classes

Department of Applied Mathematics

Supervisor of the master thesis:	prof. RNDr. Jan Kratochvíl, CSc.
Study programme:	Computer Science
Study branch:	Discrete Models and Algorithms

Prague 2019

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

In Prague, Czech Republic, 19th July 2019 Nikola Jedličková

Title: Algorithmic aspects of intersection-defined graph classes

Author: Nikola Jedličková

Department: Department of Applied Mathematics

Supervisor: prof. RNDr. Jan Kratochvíl, CSc., Department of Applied Mathematics

Keywords: graph theory, intersection graphs, geometric representations, computational complexity

Abstract:

Geometrically representable graphs are extensively studied area of research in contemporary literature due to their structural characterizations and efficient algorithms. The most frequently studied class of such graphs is the class of interval graphs. In this thesis we focus on two problems, generalizing the problem of recognition, for classes related to interval graphs.

In the first part, we are concerned with *adjusted interval graphs*. This class has been studied as the right digraph analogue of interval graphs. For interval graphs, there are polynomial algorithms to extend a partial representation by given intervals into a full interval representation. We will introduce a similar problem — the *partial ordering extension* — and we will provide a polynomial algorithm to extend a partial ordering of adjusted interval digraphs.

In the second part, we show two NP-completeness results regarding the *simultaneous representation problem*, introduced by Lubiw and Jampani. The simultaneous representation problem for a given class of intersection graphs asks if some k graphs can be represented so that every vertex is represented by the same object in each representation. We prove that it is NP-complete to decide this for the class of interval and circular-arc graphs in the case when k is a part of the input and graphs are not in sunflower position.

Název: Algoritmické otázky průnikových tříd grafů

Autor: Nikola Jedličková

Katedra: Katedra aplikované matematiky

Supervisor: prof. RNDr. Jan Kratochvíl, CSc., Katedra aplikované matematiky

Klíčová slova: teorie grafů, průnikové třídy grafů, geometrické reprezentace, výpočetní složitost

Abstrakt:

Geometricky reprezentovateľné triedy grafov sú intenzívne študovanou oblasťou výskumu v súčasnej literatúre, a to kvôli ich štrukturálnym charakterizáciám a efektívnym algoritmom. Najštudovanejšou triedou takých grafov je trieda intervalových grafov. V tejto práci sa zameriame na dva problémy, zovšeobecňujúce problém rozpoznávania, pre triedy súvisiace s triedou intervalových grafov.

V prvej časti sa zaoberáme tzv. *zarovnanými intervalovými digrafmi.* Táto trieda bola skúmaná ako správna analógia intervalových grafov. Pre intervalové grafy sú známe algoritmy pre rozširovanie čiastočných reprezentácií daných intervalov na úplnú intervalovú reprezentáciu. My predstavíme podobný problém — *rozširovanie čiastočných usporiadaní* a ukážeme polynomiálny algoritmus pre rozširovanie čiastočných usporiadaní zarovnaných intervalových digrafov.

V druhej časti práce dokážeme NP-úplnosť pre dva špeciálne prípady problému simultánnych reprezentácií grafov, ktorý predstavil Jampani a Lubiw. Problém simultánnych reprezentácií pre danú triedu grafov sa pýta, či k grafov môže byť reprezentovaných tak, že každý vrchol je reprezentovaný rovnakým objektom v každej reprezentácii. Dokázali sme, že tento problém je NP-úplný pre triedu intervalových grafov a prienikových grafov oblúkov na kružnici, kde k je súčasťou vstupu a grafy nie sú v tzv. slnečnicovej pozícii.

Acknowledgments

First of all, I would like to thank my supervisor prof. RNDr. Jan Kratochvíl, CSc. for help and advices throughout my master studies.

I would like to thank to all who supported me during my master studies — family, friends and all staff of Department of Applied Mathematics and Computer Science Institute of Charles University. I am especially grateful to Professor Pavol Hell for an invitation for a research stay at SFU in Canada and a very inspirative collaboration. Some parts of this thesis have been written during this stay.

Last but not least I would like to thank Honza Bok for everything.

Significant parts of Chapter 2 and Chapter 3 of this thesis are based on the following papers done in collaboration.

> Jan Bok and Nikola Jedličková. "A note on simultaneous representation problem for interval and circular-arc graphs." arXiv preprint arXiv:1811.04062 (2018).
>
> Jan Bok, Pavol Hell and Nikola Jedličková. "Extending Partial Orderings of Adjusted Interval Digraphs." In preparation (2019).

Contents

Chapter 1

Introduction

Geometrically represented graphs have been extensively studied in the past few decades. There are two main reasons for that and they are inseparable. The reasons are nice structural characterizations and efficient algorithms for otherwise hard problems. The core of this thesis is to study several algorithmic questions on different classes of geometrically representable graphs. More specifically, we shall make use of existing structural results for these classes to design efficient algorithms or to show that the problem is still hard, despite these elegant and strong characterizations.

The central and unifying notion of this thesis is the notion of *intersection graph*.

Definition 1.1. [62] A graph $G = (V, E)$ has an *intersection representation* by the sets from a family of sets $\mathcal{F}$ if there exists a function $f : V(G) \to \mathcal{F}$ such that two vertices from $V(G)$ are adjacent if and only if their corresponding sets have a non-empty intersection. The function f is called an *$\mathcal{F}$-representation* of the graph G.

A graph G is an *intersection graph* if it has an intersection representation. The class of all graphs with $\mathcal{F}$-representations is called an *intersection class*. In our case, $\mathcal{F}$ is a family of geometric objects. These graphs are known as *geometric intersection graphs*.

Basic structures. In this thesis we will use different generalizations of the usual definition of an undirected graph and so we need to carefully distinguish between them. We will use the notation described in the introduction of the book of Hell and Nešetřil [34]. The following definitions are taken from the book.

A *graph* is an ordered pair (V, E), where V is a set of vertices and E is a set of edges, where edges are two-element sets of vertices. We often write uv instead of $\{u, v\}$ for an edge $\{u, v\} \in E$.

A *digraph* is an ordered pair (V, E), where V is a set of vertices together with a binary relation E on V. We call the elements $(u, v) \in E$ *arcs* or *directed edges*. We often write $u \to v$ for a directed edge (u, v) in a digraph.

A *loop* in a graph is an edge consisting of only one vertex, i.e. $\{v\}$ for some vertex

$v \in V$. By a loop in a digraph we mean an arc (v, v) for a vertex $v \in V$. We say that a graph (or digraph) is *reflexive* if every vertex has a loop. If both (u, v) and (v, u) are arcs, then we say that there is a *double edge* between u and v. In each digraph, we allow at most one arc in the same direction between two vertices and every vertex has at most one loop.

All graphs and digraphs considered in this thesis are finite. By a *simple graph* we mean an undirected graph without loops and multiple edges.

There is a correspondence between graphs and digraphs. For each graph $G = (V, E)$, we can obtain a *corresponding symmetric digraph* by replacing each edge $uv \in E$ by arcs (u, v) and (v, u) and each loop $\{v\}$ by an arc (v, v). Due to this transformation, we can view graphs as symmetric digraphs. See Figure 1.1 for an example.

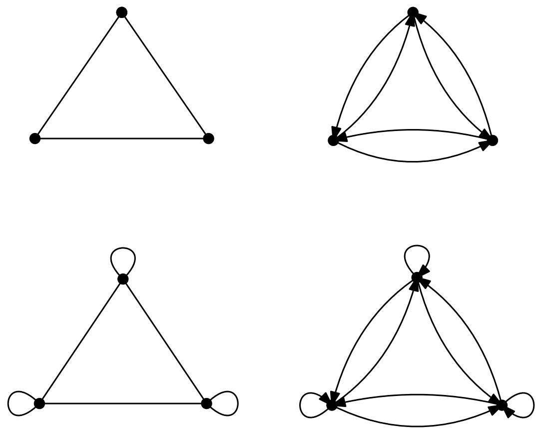


Figure 1.1: Two examples of a graph and its corresponding symmetric digraph.

We usually denote by n the number of vertices, and by m the number of edges, of the graph (resp. digraph) under consideration.

If we do not specify otherwise, we use the notation of West [71].

Structure of this thesis. This thesis is structured as follows.

- In the first chapter, we give an introduction to the theory and algorithmic aspects of geometric intersections graphs and digraphs. More specifically, we are mainly interested in interval graphs, interval digraphs, and classes related to them.
- The second chapter is about extending partial orderings and representations. We briefly summarize the results about extending partial representations, we define the problem of extending partial orderings and we show polynomial time algorithms for solving this problem for proper interval graphs and adjusted interval digraphs. Also, we briefly show a relation to other problems.
- The third chapter is about the simultaneous representation problem. First, we give a summary of recent results and we describe a relation to other

known problems. We prove NP-completeness of the problem for interval and circular-arc graphs in some special cases.

1.1 Interval graphs

Let us start with a mystery story of Claude Berge (written by Golumbic [25]).

> Six professors had been to the library on the day that the rare tractate was stolen. Each had entered once, stayed for some time and then left. If two were in the library at the same time, then at least one of them saw the other. Detectives questioned the professors and gathered the following testimony: Abe said that he saw Burt and Eddie in the library; Burt said that he saw Abe and Ida; Charlotte claimed to have seen Desmond and Ida; Desmond said that he saw Abe and Ida; Eddie testified to seeing Burt and Charlotte; Ida said that she saw Charlotte and Eddie. One of the professors lied!! Who was it?

We know that every professor entered once and stayed for some interval of time. If he saw someone, then in that time they must have been in the library together. Thus, their intervals have to intersect. Before solving this mystery, let us translate it into the language of graph theory. The key to the solution is the definition of an *interval graph*.

One of the most extensively studied classes of intersection graphs are interval graphs. Interval graphs were independently discovered in two completely separate research areas – in combinatorics by Hajos [30] in 1957 and in genetics by Benzer [4] in 1959.

Definition 1.2. A graph G is an *interval graph* if there exists a family of closed intervals on the real line $\{I_v |\ v \in V(G)\}$ such that I_u and I_v intersect if and only if uv is an edge in G.

This assignment of intervals to the vertices is called an *interval representation*. See Figure 1.2 for an example.

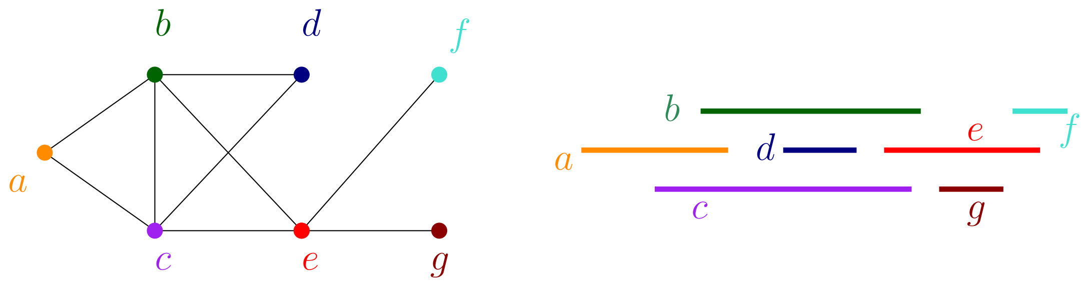


Figure 1.2: An example of an interval graph and its interval representation.

We usually treat interval graphs as simple graphs. However, observe that implicitly every vertex has a loop because it intersects itself. It follows from this observation that every interval graph is reflexive.

Booth and Lueker [8] introduced in 1976 the first linear time algorithm for recognition of interval graphs using PQ-trees. Since then, many algorithms for recognizing interval graphs in linear time were found [10, 20, 46].

Characterizations of interval graphs. Also, several nice structural characterization of interval graphs were proved.

We will need the following definitions. A graph is a *chordal graph* if it does not contain a cycle of length at least four as an induced subgraph. A graph is a *comparability graph* if its edges can be directed such that if $u \to v$ and $v \to w$, then $u \to w$.

We say that three distinct vertices create an *asteroidal triple*, if they are pairwise non-adjacent and for any two of them, there exists a path between them which does not intersect the neighborhood of the third vertex. See Figure 1.3 for an example.

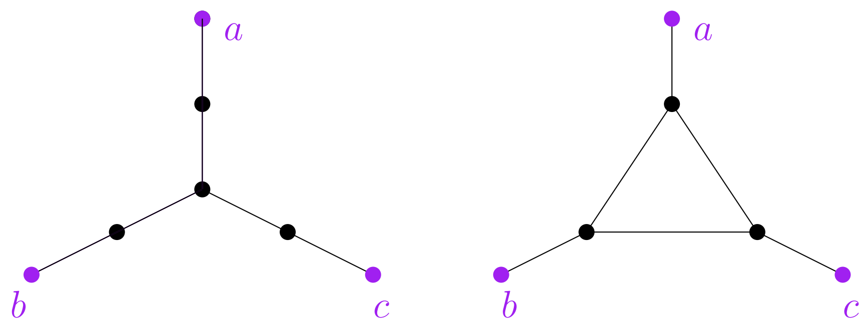


Figure 1.3: Two examples of an asteroidal triple denoted by a, b, c.

The *clique matrix* of an undirected graph G is the incidence matrix in which rows correspond to the maximal cliques of G and columns correspond to the vertices of G. We say that the clique matrix of G satisfies the *consecutive ones property* if the rows can be permuted such that ones appear consecutively in each column.

Theorem 1.1. *The following statements are equivalent for an undirected graph G.*

- *The graph G is an interval graph.*
- *The graph G is a chordal graph and G does not contain an asteroidal triple (Boland and Lekkerkerker, [48]).*
- *The graph G does not contain any graph from Figure 1.4 as an induced subgraph (Boland and Lekkerkerker, [48]).*
- *The graph G does not contain a cycle of length four as an induced subgraph and its complement is a comparability graph. (Gilmore and Hoffman, [22])*
- *There exists a linear ordering of the maximal cliques of G such that for every vertex v of G, the maximal cliques containing v occur consecutively (Gilmore and Hoffman, [22]).*
- *The clique matrix of G has the consecutive ones property for columns (Fulkerson and Gross, [19]).*

- *There exists a linear ordering of the vertices of G such that for every three vertices $u < v < w$, $uw \in E(G)$ implies $uv \in E(G)$ [55, 58]. In other words, there exists a linear ordering of the vertices avoiding the patterns from Figure 1.5.*

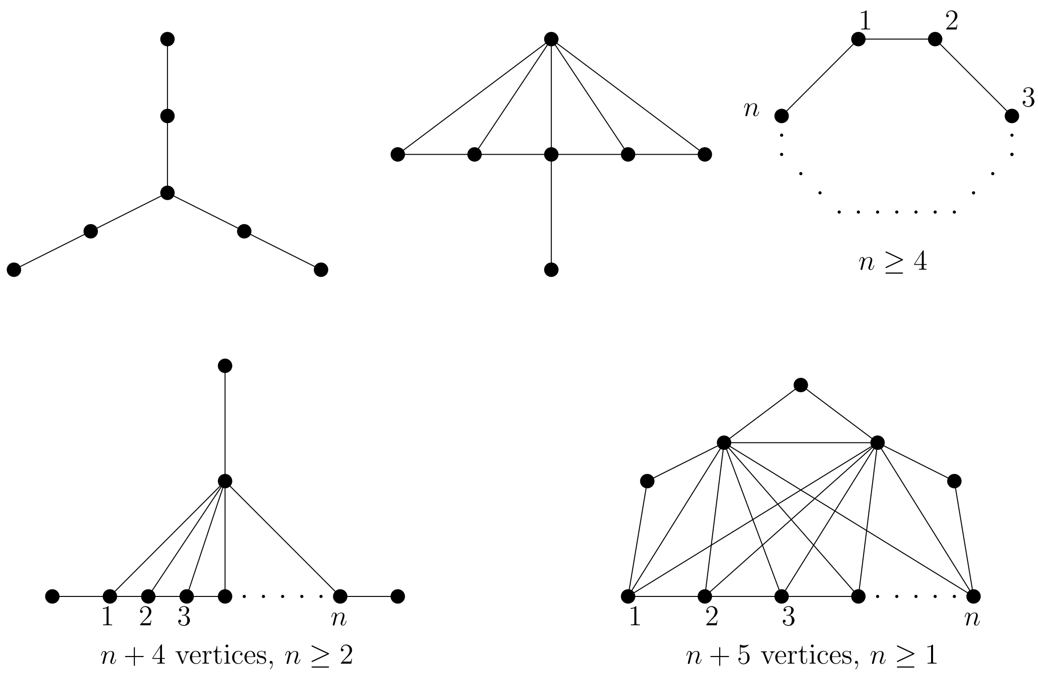


Figure 1.4: Forbidden induced subgraphs for interval graphs.

Figure 1.5: Forbidden patterns from Theorem 1.1. Edges which are not shown are absent.

Some problems which are NP-complete for graphs in general can be solved in polynomial time for the class of interval graphs, for example the maximum independent set, the minimum covering by disjoint completely connected sets or cliques, and the maximum clique (see [29]). Also, some algorithmic results about interval graphs can be found in [53].

For more information about interval graphs, we refer the reader to surveys [23, 25] or books [24, 65].

Solution of the mystery story. Let us recapitulate the story. Each professor entered once and stayed for an interval of time. If he saw someone, then their intervals intersect. By representing each entrance to the library by an interval, we should get an interval graph. However, that is not the case and therefore, someone lies.

Let us represent each professor by a vertex denoted by initials of his name. If professor X saw professor Y, then we add an arc $X \to Y$. The resulting graph G is depicted in Figure 1.6. If there are arcs in both directions between two professors, then, since at least one of them tells the truth, their intervals intersect. However, if there is only a single arc between two professors, then it does not have to be the truth. It follows from Theorem 1.1, that every interval graph is a chordal graph. The graph G has three induced cycles of length four — $ABID, AEID, AECD$. The liar has to be in all these cycles. Otherwise, everyone tells the truth and we still do not have an interval graph. Thus, we have two candidates — Abe and Desmond. If Abe is the liar, then there still remains the cycle $ABID$ because Burt was telling the truth. We conclude that Desmond is the liar!

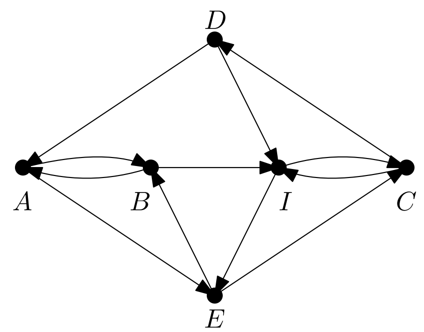


Figure 1.6: A graph of testimonies in Berge's mystery story.

Subclasses of interval graphs. Well known subclasses of interval graphs are *unit interval graphs* and *proper interval graphs*.

Definition 1.3. An interval graph is a *unit interval graph* if there exists an interval representation such that each interval has the same length.

Definition 1.4. An interval graph is a *proper interval graph* if there exists an interval representation such that no interval properly contains another interval.

Roberts [60] proved that these classes are equivalent. Particularly, he showed the following characterization.

Theorem 1.2. *[60] Let G be a simple graph. Then the following statements are equivalent.*

- *The graph G is a unit interval graph.*
- *The graph G is a proper interval graph.*
- *The graph G is an interval graph such that it does not contain a complete bipartite graph $K_{1,3}$ as an induced subgraph.*

A very short and elegant proof of this theorem was given by Bogart and West [7].

Generalization of intervals graphs and related classes. There is another possibility for defining a class of graphs represented by closed intervals on the real line. We can use the relation of *containment* instead of intersections between intervals.

Definition 1.5. [65] A graph G is an *interval containment graph* if there exists a family of intervals $\{I_v | \ v \in V(G)\}$ such that uv is an edge in G if and only if $I_u \subset I_v$ or $I_v \subset I_u$.

Interval containment graphs are in a very close relationship with the class of *permutation graphs* and comparability graphs.

Definition 1.6. A graph G is a *permutation graph* if there exists a family of line segments $\{L_v | \ v \in V(G)\}$ between two fixed parallel lines such that uv is an edge of G if and only if L_u and L_v intersect.

The connection with the notion of permutation is that the vertices of the permutation graph can be viewed as the elements of the ground set of permutation and intersection occurs if and only if the permutation reverses the relative order of the corresponding elements.

Theorem 1.3. *[65] Let G be a graph. The following statements are equivalent.*

- *The graph G is an interval containment graph.*
- *The graph G and its complement are comparability graphs.*
- *The graph G is a permutation graph.*

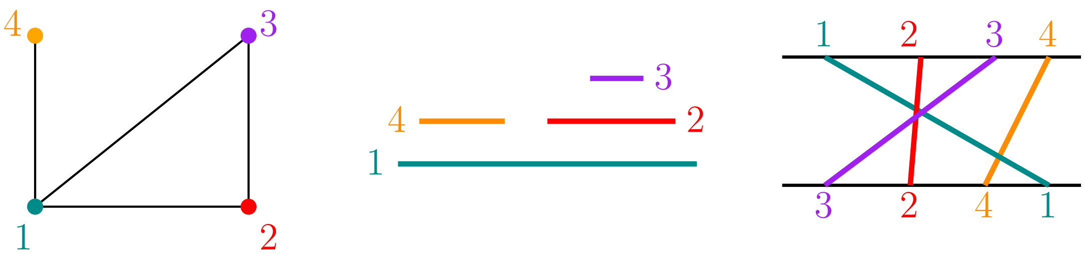


Figure 1.7: An example of a graph with an interval containment representation and a permutation representation.

We can easily see that every permutation graph is a *circle graph*. See Figure 1.8 for an example of a circle graph.

Definition 1.7. A graph G is a *circle graph* if there exists a family of chords $\{C_v | \ v \in V(G)\}$ of a fixed circle such that two chords C_u and C_v intersect if and only if uv is an edge in G.

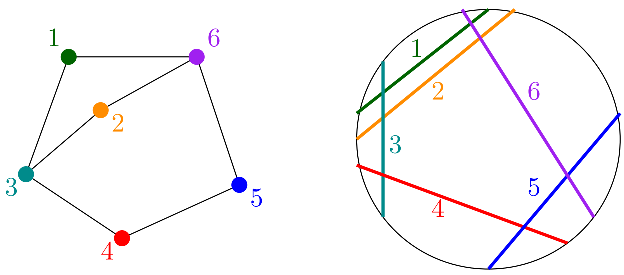


Figure 1.8: An example of a circle graph and its circle representation.

One of the well known superclasses of interval graphs which will be discussed later in this thesis is the class of *circular arc graphs*.

Definition 1.8. A graph G is a *circular arc graph* if there exists a family of arcs $\{A_v | \ v \in V(G)\}$ on a fixed circle such that two arcs A_u and A_v intersect if and only if uv is an edge in G.

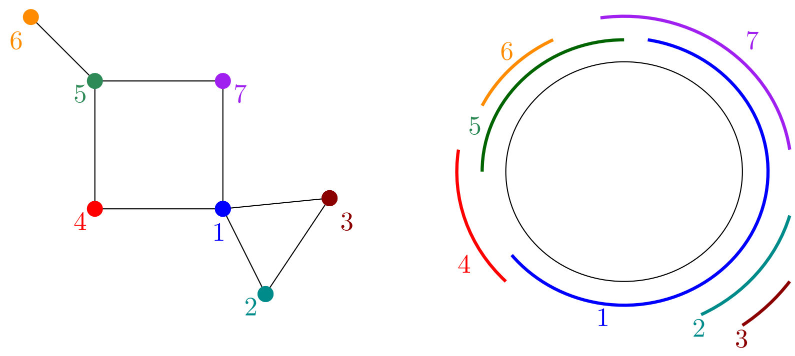


Figure 1.9: An example of a circular arc graph and its circular arc representation.

An example of a circular arc graph is depicted in Figure 1.9. Circular arc graphs were intensively studied, see for example [69, 21, 24]. They can be recognized in linear time as was shown by McConnell [51]. However, before that, some polynomial time algorithms were known [70, 13].

Similarly, as for the interval graphs before, we define two subclasses — *unit circular arc graphs* and *proper circular arc graphs*.

Definition 1.9. A circular arc graph is a *unit circular arc* graph if there exists a circular arc representation such that each arc has the same length.

Definition 1.10. A circular arc graph is a *proper circular arc* graph if there exists a circular arc representation such that no arc properly contains another arc.

For characterizations and recognition algorithms for these subclasses, we refer the reader to survey [50].

1.2 Interval digraphs

Intersection digraphs as an analogue of intersection graphs were first introduced by Beineke and Zamfirescu [3] under the name *connection digraphs*. *Interval digraphs* were introduced by Das et al. [11] in 1989 as an analogue of interval graphs.

Definition 1.11. A digraph H is an *interval digraph* if there exists a family of source intervals $\{I_v | \ v \in V(H)\}$ and a family of sink intervals $\{J_v | \ v \in V(H)\}$ such that the source interval I_u intersects the sink interval J_v if and only if (u, v) is an arc in H.

An example of an interval digraph is depicted in Figure 1.10.

Also, Das et al. obtained a similar characterization for interval digraphs as in [19]. They defined the notion of a *zero-partition* for matrices. A binary matrix has a

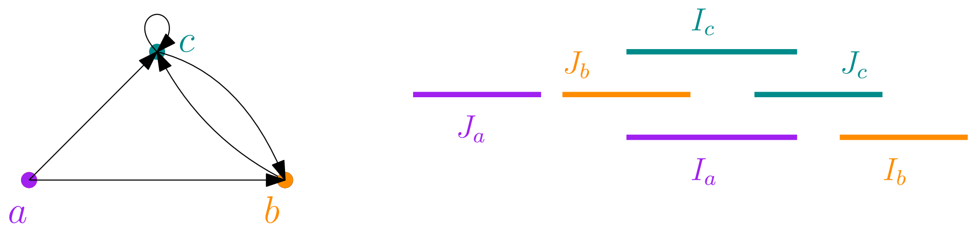


Figure 1.10: An example of an interval digraph.

zero-partition if there exists a permutation of its rows and columns and a labeling of zeros by labels R and C such that every entry to the right of an R is an R and every entry below a C is a C.

Theorem 1.4. *[11] A digraph is an* interval digraph *if and only if its adjacency matrix has a zero-partition.*

Sen et al. [63] found a characterization of unit interval digraphs using the following definition.

A binary matrix has a *monotone consecutive arrangement* (MCA) if there exists a permutation of its rows, a permutation of its columns, and a labeling of zeros by labels R and C such that every entry to the right or above of an R is labeled R and every entry to the left or below a C is labeled C.

Observe, that if a binary matrix has an MCA then it has also a zero-partition.

Theorem 1.5. *[63] A digraph is a unit interval digraph if and only if its adjacency matrix has an MCA.*

In [49], authors describe a hierarchy of classes of interval digraphs characterized by a certain property of adjacency matrices. The largest considered class is the class of interval digraphs and the smallest considered class is the class of unit interval digraphs. Some of the proofs of the adjacency matrix characterizations of interval digraphs, in a simplified form, can be also found in [72].

A representation of an interval digraph is also called a *bi-interval representation*. Observe, that for every two vertices, only the intersection of the source intervals and the sink intervals is important. The intersection of two source intervals or two sink intervals does not matter.

Independently on that, Harary et al. [31] defined a bipartite version of the notion of intersection graphs.

Definition 1.12. A bipartite graph G with parts A and B is called an *interval bigraph* if there exists a family of intervals $\{I_v | \ v \in V(G)\}$ such that for every $a \in A$ and every $b \in B$, it holds ab is an edge in G if and only if I_a and I_b intersect.

Müller [54] proved that there is a strong connection between interval digraphs defined by Das et al. [11] and interval bigraphs defined by Harary [31]. He further showed the following transformation between digraphs and bigraphs and he also found a polynomial recognition algorithm for both of these classes running in time $O(nm^6(n+m)\log n)$.

Let $G = (X \cup Y, E)$ be an interval bigraph with a representation $(\{S_x| \ x \in X\}, \{S_y| \ y \in Y\})$. We define $T_x = \emptyset$ for every $x \in X$ and $S_y = \emptyset$ for every $y \in Y$. By directing all edges from X to Y, we obtain a representation of interval digraph $G' = (X \cup Y, E')$, where $E' = \{(x, y)| \ \{x, y\} \in E, x \in X, y \in Y\}$.

Let $G' = (V, E)$ be an interval digraph with a representation by a family of source intervals $\{S_v| \ v \in V\}$ and a family of sink intervals $\{T_v| \ v \in V\}$. We split every vertex $v \in V$ into two vertices s, t such that s is represented by the interval S_v and t is represented by the interval T_v and we replace every arc v_1v_2 in G' by an edge s_1t_2. By this transformation we obtained an interval bigraph G.

It is clear from this reduction that G is an interval bigraph if and only if G' is an interval digraph.

Interval bigraphs were characterized by Hell and Huang [32].

Theorem 1.6. *[32] Let H be a bipartite graph with bipartition (X, Y). Then the following statements are equivalent.*

- *The graph H is an interval bigraph.*
- *There exists a linear ordering $v_1, v_2, \dots, v_n$ of the vertices of H such that for every three vertices $v_a < v_b < v_c$, the patterns from Figure 1.11 are forbidden. In the figure, the black vertices are from one part and the white vertices are from the other part. The edges which are not shown are absent.*
- *The complement of H is a circular arc graph with a circular arc representation such that no two arcs cover the whole circle.*

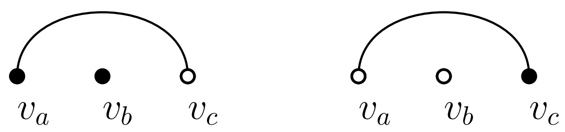


Figure 1.11: The forbidden patterns for interval bigraphs. The edges which are not shown are absent.

Recently, Rafiey [57] found an algorithm for recognizing interval bigraphs in time $O(nm)$ based on this linear ordering characterization and he showed that this algorithm can be also used for recognizing interval digraphs in time $O(nm)$.

Some subclasses of interval bigraphs can be also characterized by a property of their adjacency matrices. The *bipartite adjacency matrix* of a bipartite graph with parts X and Y is the submatrix of its adjacency matrix consisting of the rows for X and the columns for Y.

Theorem 1.7. *[72, 63] The following statements are equivalent for a binary matrix A.*

- *The matrix A is the bipartite adjacency matrix of a unit interval bigraph.*
- *The matrix A is the bipartite adjacency matrix of a proper interval bigraph.*

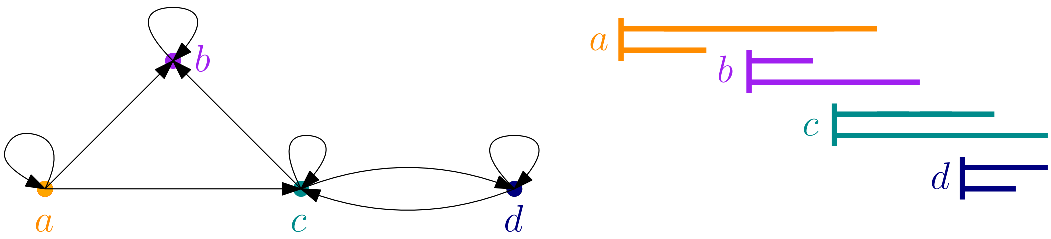


Figure 1.12: An example of an adjusted interval digraph.

- *The matrix A has a monotone consecutive arrangement.*

Some generalizations of both interval graphs and digraphs can be found in [33]. As authors said, they unified several seemingly different graph and digraph classes under one umbrella.

1.3 Adjusted interval digraphs

A more restrictive definition of interval digraphs was given by Feder et al. [17] in 2009 (the full version [18]). This class of digraphs is known under the name *adjusted interval digraphs*.

Definition 1.13. An *adjusted interval digraph* H is an interval digraph such that the intervals I_v and J_v can be chosen to have the same left endpoint for each vertex $v \in V(H)$.

See Figure 1.12 for an example of an adjusted interval digraph. Note that an interval digraph does not have to be reflexive — if I_v and J_v do not intersect, then v does not have a loop. However, adjusted interval digraphs are always reflexive because I_v and J_v always intersect in their common left endpoint.

Characterizations of adjusted interval digraphs. Several structural characterizations of adjusted interval digraphs are known. In this paragraph, we will explain two of them.

We say that two walks $P = p_1, p_2, \dots p_k$ and $Q = q_1, q_2, \dots q_k$ in a digraph H are *congruent* if for every $p_i p_{i+1}$, it holds that:

- $p_i \to p_{i+1}$ if and only if $q_i \to q_{i+1}$, and
- $p_i \leftarrow p_{i+1}$ if and only if $q_i \leftarrow q_{i+1}$.

Let P, Q be two congruent walks. Then P *avoids* Q if there is no edge between p_i and q_{i+1} in the same direction as between p_i and p_{i+1}.

Definition 1.14. [18] Two distinct vertices $u, v \in V(H)$ create an *invertible pair* in H if

- there exist congruent walks P from u to v and Q from v to u such that P avoids Q, and

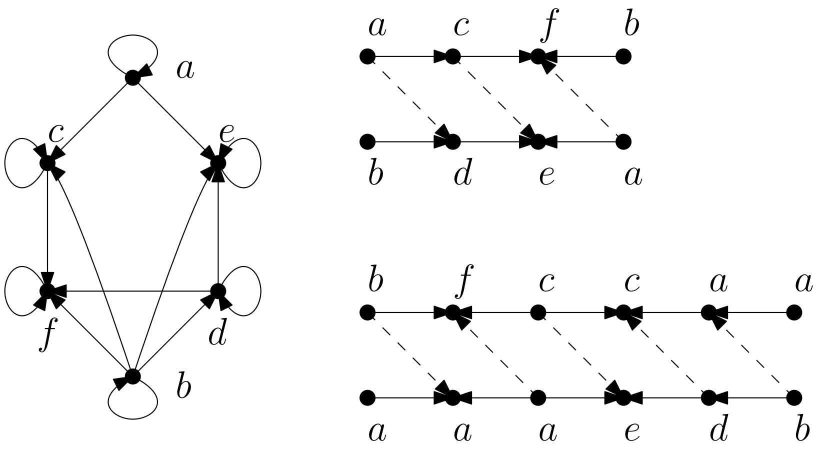


Figure 1.13: An example of an invertible pair a, b.

$$\begin{array}{cc} & \begin{array}{cccc} \quad & v' & v & \quad \end{array} \\ \begin{array}{c} u \\ u' \end{array} & \left[\begin{array}{cccc} \cdots & 0 & 1 & \cdots \\ \cdots & 1 & 1 & \cdots \end{array}\right] \end{array} \qquad \begin{array}{cc} & \begin{array}{cccc} \quad & v' & v & \quad \end{array} \\ \begin{array}{c} u \\ u' \end{array} & \left[\begin{array}{cccc} \cdots & 0 & 1 & \cdots \\ \cdots & 1 & 0 & \cdots \end{array}\right] \end{array}$$

Figure 1.14: Forbidden submatrices in the adjacency matrix of a graph with a min ordering.

- there exist congruent walks P' from v to u and Q' from u to v such that P' avoids Q'.

Theorem 1.8. *[18] Let H be a reflexive digraph. Then H is an adjusted interval digraph if and only if H does not contain an invertible pair.*

In Figure 1.13, there is an example of a digraph containing an invertible pair a, b. Also, there are depicted two pairs of congruent walks between a, b. Thus, it follows from Theorem 1.8 that this digraph is not an adjusted interval digraph.

There is another characterization of adjusted interval digraphs using the so called *min ordering.*

Definition 1.15. A *min ordering* of H is a linear ordering $<$ of $V(H)$ that satisfies the following property. If $(u, v) \in E(H)$, $(u', v') \in E(H)$, $u < u'$, and $v' < v$, then $uv' \in E(H)$.

Hell et al. [33] rephrase this definition in the following way. A digraph H has a min ordering if there exists a linear ordering $<$ of its vertices such that if we order rows and columns in the adjacency matrix with respect to $<$, then the adjacency matrix does not contain a submatrix with rows 01, 11 and a submatrix with rows 01, 10 (see Figure 1.14).

Also, several nice characterizations of a min ordering are known.

Lemma 1.1. *[18] Let H be a reflexive digraph. Then a linear ordering $<$ of $V(H)$ is a min ordering if and only if for any three vertices $i < j < k$, we have*

Figure 1.15: The forbidden patterns for a min ordering (dashed arcs are absent, arcs not shown can be arbitrary).

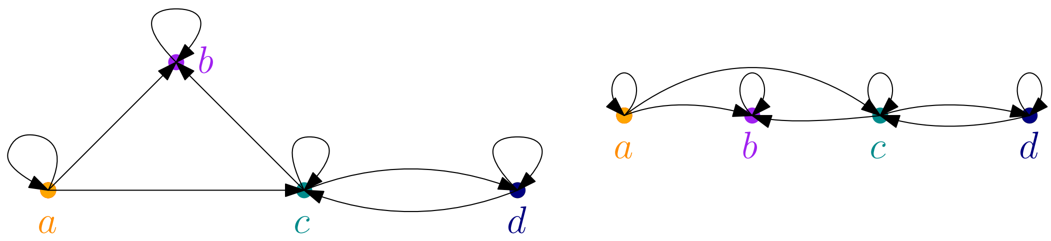


Figure 1.16: An example of an adjusted interval digraph with min a ordering.

- *if $i \to k$ then $i \to j$, and*
- *if $k \to i$ then $j \to i$.*

This lemma characterizes forbidden patterns for a min ordering (see Figure 1.15).

Lemma 1.2. *[18] Let H be a reflexive digraph. Then H has a min ordering if and only if there exists a linear ordering of vertices such that the vertices that follow v in that ordering can be divided into three consecutive groups:*

1. *vertices connected with v by a double edge,*
2. *vertices connected with v by a single edge, and*
3. *vertices not connected with v.*

Theorem 1.9. *[18] A reflexive digraph H is an adjusted interval digraph if and only if H has a min ordering.*

See Figure 1.16 for an example of a min ordering.

Detour for a motivation. Adjusted interval digraphs were defined in the context of studying the *list homomorphism problem.*

PROBLEM:	LHOM(H) – List homomorphism problem for a fixed target graph H.
INPUT:	A graph G with lists $L(v) \subseteq V(H)$ for every $v \in V(G)$.
QUESTION:	Does there exists a mapping $f : V(G) \to V(H)$ such that $f(u)f(v) \in E(H)$ whenever $uv \in E(G)$ and $f(v) \in L(v)$ for every $v \in V(G)$?

The full dichotomy of LHOM(H) was proved [14, 15, 16] for an undirected graph H. For reflexive undirected graphs, the following classification holds.

Theorem 1.10. *[14] Let H be a reflexive graph. If H is an interval graph, then the problem* LHOM*(H) is polynomial-time solvable. Otherwise, the problem* LHOM*(H) is* NP*-complete.*

A dichotomy for digraphs is also known [35]. We will need the following definitions for stating the theorem.

Definition 1.16. [35] A *permutable triple* in a digraph H is a triple of vertices u, v, w and six vertices $s(u), b(u), s(v), b(v), s(w), b(w)$ such that the following condition holds. For any vertex $x \in \{u, v, w\}$ (we denote the other two vertices by y and z), there exists a walk P_1 from x to $s(x)$ and two walks — P_2 from y to $b(x)$ and P_3 from z to $b(x)$, such that both of them are congruent to P_1 and P_1 avoids P_2 and P_3.

Definition 1.17. [35] A *digraph asteroidal triple* (DAT) in a digraph H is a permutable triple in H such that each of $(s(u), b(u)), (s(v), b(v)), (s(w), b(w))$ is an invertible pair in H.

Theorem 1.11. *[35] Let H be any digraph. If H is a DAT-free graph, then the problem* LHOM*(H) is polynomial-time solvable. Otherwise,* LHOM*(H) is* NP*-complete.*

Feder et al. were trying to find a combinatorial characterization of DAT-free graphs and they managed to show the following theorem.

Theorem 1.12. *[18] If H is an adjusted interval digraph, then* LHOM*(H) is polynomial-time solvable.*

They also posed the following conjecture and they proved that it holds if H is a tree.

Conjecture 1. [18] Let H be a reflexive digraph. If H is not an adjusted interval digraph then the problem LHOM(H) is NP-complete.

Recognition of adjusted interval digraphs. Adjusted interval digraphs can be recognized in polynomial time. Feder, Hell, Huang and Rafiey showed the first certifying recognition algorithm running in time $O(m^2 + n^2)$. Subsequently, the algorithm was improved by Takaoka [66] to cubic time – $O(n^3)$.

Both of these algorithms give a certificate whether a digraph is an adjusted interval digraph. Particularly, both of them return a min ordering in case that the answer is positive or return an invertible pair in case that the answer is negative. A decision without a certificate whether a digraph is an adjusted interval digraph can be made in $O(nm)$. We will argue that in Section 2.2.

We will continue in studying adjusted interval digraphs in Section 2.2. We will show some other related results and, most importantly, we will introduce our problem and results for adjusted interval digraphs.

Chapter 2

Extending partial orderings and representations

We study two natural generalizations of the recognition problem — the problem of extending partial representations and the problem of simultaneous representations. A recognition algorithm gives a representation of the input graph, but sometimes we may want to find a representation with some additional properties.

2.1 Extending partial representations

The *partial representation extension problem* was introduced by Klavík, Kratochvíl and Vyskočil [45] in 2011 (journal version appeared in Algorithmica in 2017 [44]).

A *partial representation* R' of a graph G is a representation of an induced subgraph G' of G. The vertices of G' and the sets of R' are called *pre-drawn*. A representation R of G extends a partial representation R' if for every $v \in V(G')$: $R(v) = R'(v)$. The partial representation extension problem for a fixed class of intersection graphs $\mathcal{C}$ is defined as follows.

PROBLEM:	$\textsc{RepExt}(\mathcal{C})$ – Partial representation extension of $\mathcal{C}$.
INPUT:	A graph G and its partial $\mathcal{C}$-representation R'.
QUESTION:	Is there a $\mathcal{C}$-representation R of G extending R'?

An instance of the problem $\textsc{RepExt}(\mathsf{INT})$ with pre-drawn intervals for vertices a, b, e and one of the possible extending representations is shown in Figure 2.1.

The partial representation extension problem was intensively studied and also, there are some very recent results. All results relevant to this problem are comprehensively summarized in PhD thesis of Pavel Klavík from 2017 [40]. We refer the reader to that thesis for further details, explanations and references. In order to avoid duplication, we picked only the results for the graph classes related to this thesis and we summarize them in Table 2.1.

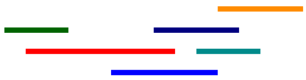
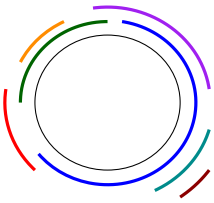
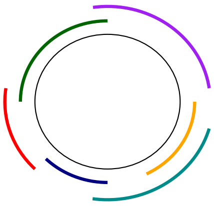
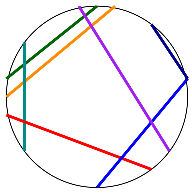
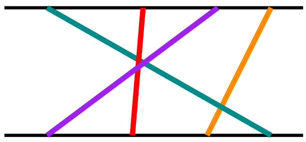
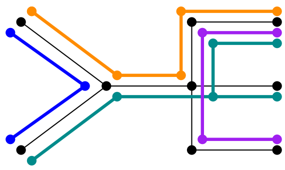
class of graphs $\mathcal{C}$	**time complexity of RepExt($\mathcal{C}$)**
interval graphs 	$O(n+m)$ [45, 44, 6]
unit interval graphs 	$O(n^2)$ [42, 64]
proper interval graphs 	$O(n+m)$ [42]
circular arc graphs 	open
proper circular arc graphs 	polynomial time [2]
circle graphs 	polynomial time [9]
permutation graphs 	$O(nm)$ [41]
chordal graphs 	NP-complete [43]

Table 2.1: A summary of results about RepExt for selected graph classes.

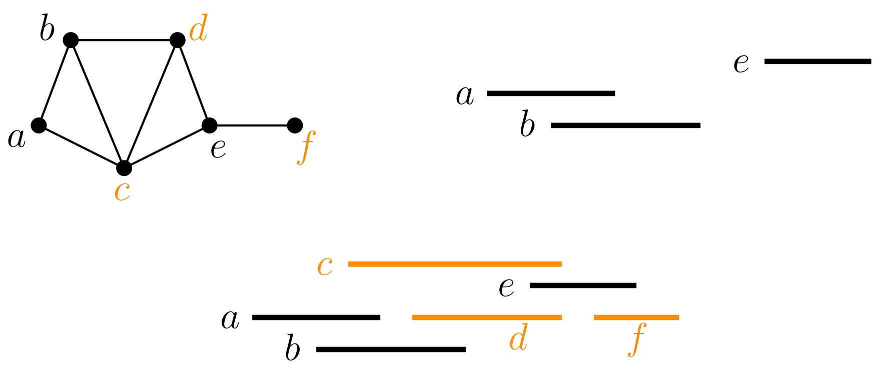


Figure 2.1: An example of an instance of the problem RepExt(INT).

2.2 Extending partial orderings

Several classes of intersection graphs can be characterized by a linear ordering of vertices satisfying some conditions. Also, we can often derive a lot of information about a possible representation from such orderings which further underlines the importance of studying such orderings.

As for extension of partial representation, we may want to find not an arbitrary linear ordering, but a special ordering where some pairs of vertices are pre-ordered. In the following two sections, we will define a new problem for proper interval graphs and adjusted interval digraphs and we will also present our results related to these problems.

2.2.1 Extending partial orderings of proper interval graphs

As the first case, we consider the class of proper interval graphs. This is mainly because of the theorems of Roberts [59] and Deng et al. [12] which describe nicely the structure of possible left endpoint orderings and which basically say that these orderings are very simple.

Let G be an undirected graph. For any vertex u of G, the *neighborhood* of vertex u, denoted by $N(u)$, is defined as the set of vertices adjacent to u in G. A set $N[u] := N(u) \cup \{u\}$ is called the *closed neighborhood* of the vertex u.

Proper interval graphs can be characterized as follows.

Theorem 2.1. *[59] A graph is a proper interval graph if and only if there exists a linear ordering of its vertices such that the closed neighbourhood of each vertex is consecutive.*

With this characterization in hand, we can state our problem.

PROBLEM:	REPORD(PROPER INT) – Extending partial orderings of proper interval graphs.
INPUT:	A graph G, a set of pre-ordered pairs of vertices $P \subseteq \{(u,v) \mid u,v \in V(G), u \neq v\}$.
QUESTION:	Does there exist a linear ordering of vertices $<$ satisfying conditions from Theorem 2.1 such that $u < v$ whenever $(u,v) \in P$?

The following theorem by Deng et al. [12] will be a key part of our algorithm. We will use a simpler reformulation of this theorem and related definitions by Klavík et al. [42].

Let G be an undirected graph. We say that vertices u and v of G are *indistinguishable* if $N[u] = N[v]$ in G. The vertex set of a graph G can be partitioned into *groups* of pairwise indistinguishable vertices. Note that indistinguishability is an equivalence relation and the groups are the equivalence classes.

Theorem 2.2. *[12] For a connected proper interval graph, the ordering $\lhd$ satisfying conditions from Theorem 2.1 is uniquely determined up to local reordering of groups of indistinguishable vertices and complete reversal. This ordering can be found in time $O(n+m)$.*

Algorithm for extending partial ordering of proper interval graphs.

In the following algorithm, we assume that the graph G on the input is connected.

1. Find a linear ordering $\lhd$ of the vertices of G as in Theorem 2.2.
2. Find a pair $(u,v) \in P$ such that u and v are in different groups.
   (a) If there is no such pair, set $< := \lhd$.
   (b) If there is such pair and $u \lhd v$, set $< := \lhd$.
   (c) If there is such pair and $v \lhd u$, set $<$ to be reversal of $\lhd$.
3. For every group B of G, we define a graph G_B with $V(G_B) := B$ and $E(G_B) := \emptyset$.
4. For each pair $(u,v) \in P$, do the following steps.
   (a) If u and v are in different groups and $v > u$, then return that P cannot be extended.
   (b) If u and v are in the same group B, insert the edge $u \to v$ into G_B. If there already was $v \to u$ in G_B, then return that P cannot be extended.
5. For each group B of indistinguishable vertices, do the following steps.
   (a) Find a linear ordering $<_B$ of $V(G_B)$ using a topological sort algorithm such that $u <_B v$ whenever $u \to v$. If it does not exist return that P cannot be extended.

(b) For each $u, v \in B$, define $u < v$ whenever $u <_B v$.

6. Return the ordering $<$.

In the case where we have a graph with at least two connected components, we can argue in a very similar way. Indeed, it suffices to further distinguish pre-ordered pairs consisting of vertices from two different components and insert these pair in form of directed edges into an auxiliary graph. Again, as for the graphs for groups of indistinguishable vertices, we need to topologically sort this graph. Furthermore, the fixation of the linear ordering we get from Theorem 2.2 needs to be done for each connected component separately. We observe that the computational complexity remains the same.

Theorem 2.3. RepOrd(PROPER INT) *can be solved in time* $O(n + m + |P|)$*, i.e. the problem can be solved in time linear to the size of the input.*

Steps 2 and 4 of this algorithm can be made in linear time with regards to the size of the input set P. Clearly, all other steps can be done in $O(n + m)$. Thus, the theorem follows.

Note that the algorithm can get as its input P which is not a partially order set. However, even in this case, our algorithm correctly detects that and outputs that P is not extendable.

Finally, we note that we are aware of the algorithm for RepExt(PROPER INT) from [42]. The problem RepOrd(PROPER INT) shares some similarities due to the structural characterizations of the class of proper interval graphs. However, none of the two algorithms can be directly translated to the other and thus our solution and its analysis is not redundant.

A related problem: Allen's interval algebra. Allen introduced the so-called interval algebra in [1]. The problem asks if there is an interval representation which satisfies given constraints having a form of one of thirteen primitive relations between pairs of intervals. A complete complexity classification of the problem was given by Krokhin et al. in [47]. All of the problems Recog(PROPER INT), RepExt(PROPER INT), and RepOrd(PROPER INT) can be reformulated as a special case of these problems. However, the corresponding general problems are NP-complete in case of RepOrd(PROPER INT) and thus the dichotomy of Krokhin et al. does not imply the polynomial result we obtained.

2.2.2 Extending partial orderings of adjusted interval digraphs

In this section, we introduce the problem of extending partial orderings of adjusted interval digraphs and we will show a polynomial time algorithm for it.

Problem:	Extending partial orderings of adjusted interval digraphs.
Input:	A reflexive digraph H, a set of pre-ordered pairs of vertices $P \subseteq \{(u, v)\mid u, v \in V(H), u \neq v\}$.
Question:	Does there exist a min ordering $<$ of vertices of H such that $u < v$ whenever $(u, v) \in P$?

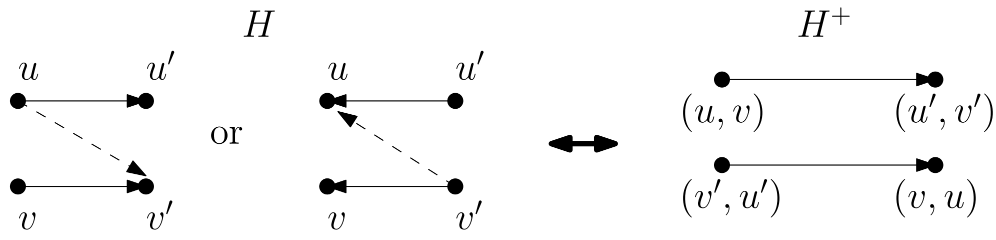


Figure 2.2: An illustration of Definition 2.1. Dashed arcs are absent, arcs which are not shown can be arbitrary.

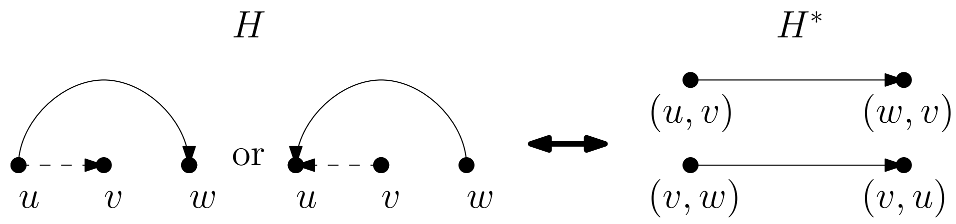


Figure 2.3: An illustration of Definition 2.2. Dashed arcs are absent, arcs which are not shown can be arbitrary.

We say that a min ordering satisfying the above condition is *extending* P. We need to define two auxiliary graphs for our algorithm — a *pair digraph* by Feder et al. [18] and an *implication digraph* by Takaoka [66].

Definition 2.1. [18] Let H be a reflexive digraph. The *pair digraph* H^+ associated with H is defined as $V(H^+) = \{(u,v)|\ u,v \in V(H), u \neq v\}$ and for any vertices u, v, u', v' of H, $(u,v) \to (u',v')$ and $(v',u') \to (v,u)$ in H^+ if and only if

- $(u,u'), (v,v') \in E(H)$ and $(u,v') \notin E(H)$, or
- $(u',u), (v',v) \in E(H)$ and $(v',u) \notin E(H)$.

Observe, that for a directed walk from (u,v) to (v,u) in H^+, there are congruent walks P (from u to v) and Q (from v to u) such that P avoids Q. Also, for two such paths, there exists a directed walk from (u,v) to (v,u) in H^+.

Theorem 2.4. *[18] Let H and H^+ be a reflexive digraph and its pair digraph, respectively. A pair of two vertices $u,v \in V(H)$ is an invertible pair if and only if the vertices (u,v) and (v,u) are in the same strong component of H^+.*

Definition 2.2. [66] Let H be a reflexive digraph. The *implication graph* H^* of H is a digraph such that $V(H^*) = \{(u,v)|u,v \in V(H), u \neq v\}$ and for any three vertices u, v, w of H, $(u,v) \to (w,v)$ and $(v,w) \to (v,u)$ in H^* if and only if

- $(u,w) \in E(H)$ and $(u,v) \notin E(H)$, or
- $(w,u) \in E(H)$ and $(v,u) \notin E(H)$.

Note that H^* is a subgraph of H^+. Moreover, H^* contains exactly these edges $(a,b) \to (c,d)$ from H^+, where either $a = c$ or $b = d$. For every edge $(a,b) \to (c,d)$ in H^+ where a, b, c, d are distinct, there is a directed path $(a,b) \to (a,d) \to (c,d)$ in H^*.

Finally, observe that if there is a directed path from (u, v) to (w, z) in H^* then there is also a directed path from (z, w) to (v, u) in H^*. Also, the same holds for H^+. It follows from these observations that the reachability in H^+ and H^* is the same.

Similarly as for H^+, there is a characterization of invertible pairs in H^*.

Lemma 2.1. *[66] Let H and H^* be a reflexive digraph and its implication graph, respectively. A pair of two vertices $u, v \in V(H)$ is an invertible pair if and only if the vertices (u, v) and (v, u) are in the same strong component of H^*.*

Theorem 2.5. *[18] Let H^+ be the pair digraph associated with a reflexive digraph H. Then the following statements are equivalent.*

1. *The digraph H is an adjusted interval digraph.*
2. *The digraph H has a min ordering.*
3. *The digraph H has no invertible pairs.*
4. *The vertices of H^+ can be partitioned into two sets D, D' such that*
    (a) $(x, y) \in D$ *if and only if* $(y, x) \in D'$,
    (b) *if* $(x, y) \in D$ *and* $(x, y) \to (w, z) \in H^+$ *then* $(w, z) \in D$,
    (c) *if* $(x, y), (y, z) \in D$ *then* $(x, z) \in D$.

We will prove the same result for an implication digraph H^*.

Theorem 2.6. *Let H^* be an implication graph of a reflexive digraph H. Then H is an adjusted interval digraph if and only if the vertices of H^* can be partitioned into two sets D, D' such that the following properties hold.*

1. $(x, y) \in D$ *if and only if* $(y, x) \in D'$,
2. *if* $(x, y) \in D$ *and* $(x, y) \to (w, z) \in H^*$ *then* $(w, z) \in D$,
3. *if* $(x, y), (y, z) \in D$ *then* $(x, z) \in D$.

Proof. Assume that H is an adjusted interval digraph. Then $V(H^+)$ can be partitioned into two sets D, D' as in Theorem 2.5. We will prove that D, D' is the desired partition for $V(H^*)$. The properties (1) and (3) of Theorem 2.6 hold because H^+ and H^* have precisely the same vertex set. Property (2) also holds, since H^* is a subgraph of H^+ and thus there cannot be an edge beginning in D and ending in D'.

For the second part, assume that there exists a partition of $V(H^*)$ into D and D' as in the claim. For a contradiction, we assume that this is not a right partition for H^+. Again, it is easy to see that the properties (1) and (3) of the partition from Theorem 2.5 hold. Thus, property (2) of Theorem 2.5 has to be violated. Let $(a, b) \to (c, d)$ be an edge in H^+ such that $(a, b) \in D$ a $(c, d) \notin D$. Then, a, b, c, d are four distinct vertices. Otherwise, this edge would violate the properties of partition D, D' for H^*.

Now, there always exists a path $(a, b) \to (a, d) \to (c, d)$ in H^*, whenever $(a, b) \to (c, d)$ is in H^+. Therefore, one of the edges on the path has to go between a vertex in D and a vertex in D'. This is a contradiction, since we assumed D, D' is a right partition of $V(H^*)$.

We conclude that D, D' is the right partition for H^+ and by Theorem 2.5, H is an adjusted interval digraph. This finishes our proof. □

For a reflexive digraph H, the implication graph H^* can be constructed in time $O(nm)$. Searching for strongly connected components in H^* takes also $O(nm)$ [68]. Thus, it follows from Lemma 2.1 and Theorem 2.6 that a decision whether a digraph is an adjusted interval digraph can be made in $O(nm)$ time if a certificate is not required.

We showed that H^* has the same properties as H^+. This will be much needed in the next part. Since H^* has a simpler structure, we will use H^* in the rest of the thesis.

We use an auxiliary complete graph K with the vertex set $V(H)$ for a digraph H. We obtain an orientation of K by replacing each edge uv in $E(K)$ with either (u, v) or (v, u). An orientation of K is *acyclic* if it contains no directed cycles.

An orientation of a complete graph is called *tournament*. We say that an orientation T of tournament K is *consistent* with H if for each vertex (u, v) of H^*, it holds that if $u \to v$ in T then $u' \to v'$ in T for each (u', v') such that there is a directed path from (u, v) to (u', v') in H^*.

Lemma 2.2. *[66] Let T be an orientation of K consistent with H. Suppose that T has three vertices u, v, w such that $u \to v, v \to w, w \to u$ in T. If $v' \to w'$ in T and $(v', w') \to (v, w)$ in H^*, then $u \to v', v' \to w', w' \to u$ in T.*

Let T be an orientation of the complete graph K. For each vertex $u \in V(H)$, we define sets

- $E_u := \{(v, w) \in E(T) |\ u \to v, v \to w, w \to u \in T\}$, and
- $E_u^- := \{(w, v) |\ (v, w) \in E_u\}$.

For a given set $S \subseteq \{(u, v) |\ u, v \in V(H), u \neq v\}$ and an orientation T of K, we define a set

$$F_s = \{x \in V(H) |\ \exists y, z \in V(H) \wedge x \to y \in T \wedge y \to z \in T \wedge z \to x \in T \wedge (y, z) \in S\}.$$

Algorithm for extending partial ordering of adjusted interval digraphs.

1. Let $D := P$ and $D' := \{(y, x) |\ (x, y) \in P\}$.
2. Construct the implication graph H^* of H.
3. Repeat until there is no new vertex added to D:
   (a) *(Implication closure.)* While there exists an edge in H^* from any vertex $(x, y) \in D$ to a vertex $(w, z) \in H^* \setminus (D \cup D')$ then add (w, z) to D and (z, w) to D'. If there is an edge from a vertex in D to a vertex in D', then P is not extendable.

(b) *(Transitive closure.)* If there are $(u,v),(v,w) \in D$ and $(u,w) \in D'$, stop the algorithm and output that P is not extendable. Otherwise, while there exist $(u,v),(v,w) \in D$ and $(u,w) \notin D$, add (u,w) to D and (w,u) to D'.

4. Define $\overline{P} := D$.
5. Define a 2CNF formula ϕ_H consisting of the following clauses:
   - x_{vu} for every $(u,v) \in \overline{P}$, and
   - $(x_{uv} \vee x_{vw})$ for every u,v,w such that $(u,w) \in E(H)$ and $(u,v) \notin E(H)$ or $(w,u) \in E(H)$ and $(v,u) \notin E(H)$.
6. If ϕ_H is satisfiable, then find a satisfying truth assignment τ. Otherwise, output that P is not extendable.
7. Define an orientation T of K such that $u \to v$ in T if and only if $\tau(x_{uv}) = 0$.
8. While there is a directed triangle in T:

   (a) For the orientation T and the set $\overline{P}$, find the set $F_{\overline{P}}$.

   (b) Find a vertex $u \in V(H) \setminus F_{\overline{P}}$ with E_u non-empty and replace E_u by E_u^- in T.

   (c) If there is no $u \in V(H) \setminus F_{\overline{P}}$ with E_u non-empty, then P is not extendable.
9. Return the orientation T extending P.

Correctness. It is easy to see that if the algorithm fails in Step 3a or 3b, then P is not extendable. Otherwise, the properties of sets D and D' from Theorem 2.6 would be violated.

Let ϕ_H be the formula from Step 5. There is a clause $(x_{uv} \vee x_{vw})$ in ϕ_H if and only if $(u,v) \to (w,v)$ is an edge in H^*. Observe that the consistency of some orientation T is violated, i.e. $u \to v$ and $v \to w$ are present in T if and only if $x_{uv} = 0$ and $x_{vw} = 0$. Thus ϕ_H is satisfiable if and only if there exists an orientation T of tournament K such that T is consistent with H and for every $(u,v) \in \overline{P}$, there is $u \to v$ in T.

Now suppose that our algorithm constructed an orientation T in Step 7. It follows from constructing this orientation and from the previous observation that T is consistent. However, there is a possibility that it contains a directed cycle. It is a classical result that if a tournament contains a directed cycle, then it must contain a directed triangle.

Lemma 2.3. *A tournament is acyclic if and only if it does not contain a directed triangle.*

Proof. Clearly, a tournament with a directed triangle is not acyclic.

Now suppose that we have a cyclic tournament with a directed cycle C of the shortest length. If the length is 3, we are done. Otherwise, there is a chord in the cycle which, depending on its orientation, completes a directed cycle C' with one of the parts of C which has fewer edges than C, a contradiction with C being minimal. □

We conclude that T is not transitive if and only if it contains a directed triangle.

The algorithm tries to modify the orientation T in Step 8b in order to make it acyclic while still consistent. Two following lemmata will complete the proof of correctness. First, we prove that any failure of the algorithm after the initial successful construction of T means that the given pre-ordering is not extendable. After that, we show that if the algorithm reaches Step 9, we get a consistent acyclic orientation from which we can construct the sought min ordering of H.

Lemma 2.4. *If the algorithm fails in Step 8c, then P is not extendable.*

Proof. If the algorithm fails in Step 8c, then there is no vertex $u \in V(H) \setminus F_{\overline{P}}$ with E_u non-empty and there is still a directed triangle in T. This means that for each of the vertices of the triangle, the edge consisting of the other two vertices of the triangle is in $\overline{P}$. Thus, all the edges of the triangle are in $\overline{P}$ and there cannot exist any orientation T extending P without that triangle. Hence, P is not extendable. □

We prove the following claim similarly to the proof of Lemma 4 in [66].

Lemma 2.5. *If the algorithm outputs an orientation T, then it is acyclic and consistent.*

Proof. In other words, we claim that the algorithm does not make a new directed triangle in any iteration of Step 8b and the resulting orientation is still consistent.

No new triangles. Suppose for a contradiction that there is a new directed triangle $x \to y, y \to z, z \to x$ in T after replacing E_u by E_u^- in some iteration of Step 8b for some vertex u. Clearly, the triangle does not contain the vertex u.

It is easy to see that there cannot be more than one arc from $x \to y, y \to z, z \to x$ in E_u^-. Thus, assume that one of the arcs $x \to y, y \to z, z \to x$ is in E_u^-. Without loss of generality, let $(x, y) \in E_u^-$ and $(y, z), (z, x) \notin E_u^-$. Then there was a directed triangle $u \to y, y \to x, x \to u$ before replacing E_u by E_u^-. We distinguish two cases.

- If $u \to z$ then there is a directed triangle $u \to z, z \to x, x \to u$ and thus $(z, x) \in E_u^-$, a contradiction.
- If $z \to u$ then there is a directed triangle $u \to y, y \to z, z \to u$ and thus $(y, z) \in E_u^-$, a contradiction.

Consistency preservation. Again, suppose for a contradiction that T is not consistent after some iteration of Step 8b. We choose the first such iteration. Then there exist $x \to y$, $y \to z$ in T such that there is $(x, y) \to (z, y)$ in H^*.

Since T was consistent before the iteration of Step 8b, there exists a vertex u such that $(x, y) \in E_u^-$ or $(y, z) \in E_u^-$.

We distinguish three cases.

- If $(x, y) \in E_u^-$ and $(y, z) \notin E_u^-$, then there was a directed triangle $u \to y, y \to x, x \to u$ in T before replacing E_u by E_u^- and there is an edge $(y, z) \to (y, x)$ in H^*. It follows from Lemma 2.2 that there is also a triangle $u \to y, y \to z, z \to u$ in T and thus $(y, z) \in E_u^-$, a contradiction.
- If $(x, y) \notin E_u^-$ and $(y, z) \in E_u^-$, then we can argue analogously to the previous case, we just use the edge $(x, y) \to (z, y)$ instead of the edge $(y, z) \to (y, x)$ in H^* in assumptions of Lemma 2.2.
- If $(x, y) \in E_u^-$ and $(y, z) \in E_u^-$, then there was a directed triangle $u \to y, y \to x, x \to u$ and also a directed triangle $u \to z, z \to y, y \to u$ before replacing E_u by E_u^- which is not possible because the edge between u and y cannot be directed in both ways.

This concludes the proof. □

We showed that the orientation T returned by the algorithm is acyclic and consistent. Note that an acyclic orientation of a tournament directly translates into a linear ordering of its vertices and this linear ordering is a min ordering of vertices of H.

Corollary 2.1. *The algorithm outputs an orientation T extending P if and only if P is extendable.*

Time complexity. We now analyze the time complexity of our algorithm.

Theorem 2.7. *Extending partial ordering of adjusted interval digraph can be made in time $O(n^5)$.*

Proof. Recall that the number of vertices in H^* is $O(n^2)$ and the number of edges is $O(nm)$, with n being the number of vertices in H and m being the number of edges in H.

Complexity of computing all implication closures in Step 3a is $O(nm)$. For transitive closures (Step 3b) we need, in the worst case, run a transitive closure algorithm for every vertex of H^*. If we use repeated depth-first search approach for that, we end up with total time complexity of $O(n^5)$ for this step.

Finding a satisfying assignment of the 2CNF formula takes $O(nm)$ time. The modification of the orientation then takes $O(n^2m)$ time in total. We conclude that the final complexity of the algorithm is $O(n^5)$. □

Chapter 3

Simultaneous representation problem

If we have several related graphs sharing some vertices, we may want to find some way how to represent them simultaneously. For example, graphs can represent a different relation on the same set of vertices or changes in time in case we are considering overlap between an old and a new relation.

The first question of this flavor was how to draw a series of planar graphs sharing some vertices. This problem is well known under the name *simultaneous embedding of planar graphs*. It gained a lot of attention in the literature, see survey [5] for more references.

Motivated by this line of research, Jampani and Lubiw [37] in 2009 introduced and studied a very similar problem, defined for any intersection class of graphs, which they called the *simultaneous representation problem*.

Definition 3.1. Let $\mathcal{C}$ be a class of intersection graphs. Graphs $G_1, \dots, G_k \in \mathcal{C}$ are *simultaneously representable* (or *simultaneous*) if there exist representations $R_1, \dots, R_k$ of $G_1, \dots, G_k$ such that

$$\forall i, j \in \{1, \dots, k\}\ \forall v \in G_i \cap G_j : R_i(v) = R_j(v).$$

The *simultaneous representation problem* for a class $\mathcal{C}$ ($\textsc{SimRep}(\mathcal{C})$) asks if given k graphs $G_1, \dots, G_k \in \mathcal{C}$ are simultaneous.

We distinguish whether k is fixed or if it is a part of the input. The problem can be further divided into two cases, depending on whether the graphs on the input are in *sunflower position* or not.

Definition 3.2. We say that graphs $G_1, \dots, G_k$ are in *sunflower position* if there exists a set of vertices I such that $G_i \cap G_j = I$ for every $i \neq j$. Otherwise, we say that these graphs are in *non-sunflower* position.

Figures 3.1 and 3.2 show some examples of simultaneous and non-simultaneous graphs, respectively.

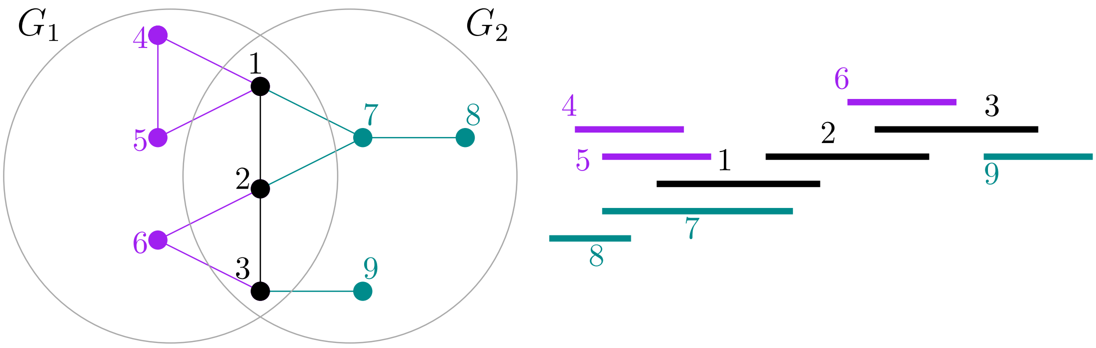


Figure 3.1: An example of two simultaneous interval graphs.

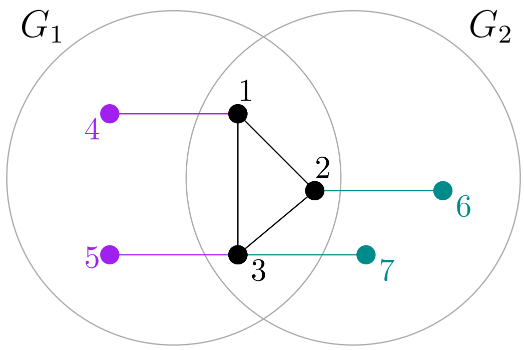


Figure 3.2: An example of two non-simultaneous interval graphs.

3.1 Related problems

The simultaneous representation problem is strongly related to two other problems — *recognizing of probe graphs* and the *graph sandwich problem.*

Graph sandwich problem. The graph sandwich problem was defined by Golumbic and Shamir [28] in 1991.

Definition 3.3. Let $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ be two graphs such that $E_1 \subseteq E_2$. We say that $G = (V, E)$ is a *sandwich graph* for G_1, G_2 if for its set of edges, it holds $E_1 \subseteq E \subseteq E_2$.

PROBLEM:	Graph sandwich problem for a property Γ.
INPUT:	Two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ such that $E_1 \subseteq E_2$.
QUESTION:	Does there exists a sandwich graph G for G_1 and G_2 such that G satisfies the property Γ?

The property is often inclusion into some class of graphs. This problem was first defined for the property "being an interval graph" under the name *interval graph sandwich problem.* It was shown that the interval sandwich problem is NP-complete [28]. Subsequently, the complexity was settled for some other graph classes, for example for unit interval graphs in [26] and for permutation graphs,

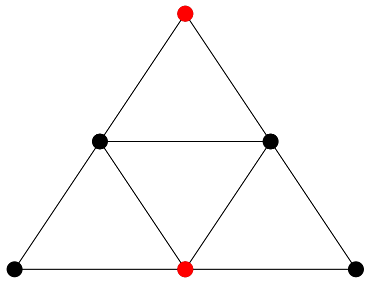


Figure 3.3: An example of a probe interval graph where probes are colored red.

comparability graphs, threshold graphs, split graphs and several other classes in [27].

The graph sandwich problem plays an important role in molecular biology in the problem called the *physical mapping problem of DNA*. For further details and explanation, see [26].

The interval graph sandwich problem is a generalization of the problem SimRep(INT). For given graphs H_1 and H_2 from an instance of SimRep(INT), we define an instance of the graph sandwich problem as follows. Let $G_1 := H_1 \cup H_2$ and $G_2 := G_1 \cup E'$ where $E' = \{\{u, v\}|\ u \in V(H_1) \setminus V(H_2), v \in V(H_2) \setminus V(H_1)\}$. Then H_1 and H_2 are simultaneously representable if and only if there exists an interval sandwich graph for G_1 and G_2.

However, the interval sandwich problem is NP-complete, so this reduction does not say anything about the complexity of SimRep(INT).

Probe graphs. Probe graphs as a new class of graphs were introduced by Zhang [73] in 1994. Similarly as the graph sandwich problem, probe graphs are also connected to molecular biology.

Definition 3.4. Let $G = (V, E)$ be a graph, $P \subseteq V$ be a set of his *probes*. Then G is a *probe interval graph* if there exists a family of intervals $\{I_v|\ v \in V\}$ such that uv is an edge if $u \in P$ or $v \in P$ and I_u and I_v intersect.

Probe graphs can be analogously defined for other graph classes. We call the vertices of $V \setminus P$ *non-probes*. An interval probe graph is a generalization of an interval graph in the sense that we do not have information about intersections of non-probes. The problem of recognizing interval probe graphs can be reformulated as follows. For a given graph $G = (V, E)$ with a set of probes P, we ask if there exists a set of edges $E' \subseteq \{\{u, v\}|\ u, v \in V \setminus P\}$ such that $G' = (V, E \cup E')$ is an interval graph.

The first polynomial time algorithm for recognizing probe interval graphs was presented by Johnson and Spinrad [39]. A few years later, McConnell and Nussbaum [52] provided a linear time algorithm.

Also, the interval graph sandwich problem is a generalization of the problem of recognizing interval probe graphs. Let a graph $G = (V, E)$ with a set of probes P be an instance of the problem of recognizing interval probe graphs. We define graphs $G_1 := (V, E)$ and $G_2 := (V, E \cup E')$, where $E' = \{\{u, v\}|\ u, v \in V \setminus P\}$.

class of graphs	**two graphs**	**k graphs, k not fixed**
chordal graphs	$O(n^3)$	NP-hard
comparability graphs	$O(nm)$	$O(nm)$
permutation graphs	$O(n^3)$	$O(n^3)$

Table 3.1: A summary of results for SimRep for graphs in sunflower position from paper [37].

Subsequently, G is an interval probe graph if and only if there exists an interval sandwich graph for G_1 and G_2.

It follows from this reduction that the interval graph sandwich problem can be solved efficiently (using the algorithm for recognizing probe interval graphs) in case that edges $E(G_2) \setminus E(G_1)$ form a complete graph.

3.2 Recent work

The first paper about the simultaneous representation problem for intersection classes of graphs was by Jampani and Lubiw [37]. They solved the simultaneous representation problem for chordal, comparability and permutation graphs in sunflower position for both k fixed and k as a part of the input.

As comparability graphs do not have an implicit intersection representation, let us explain the definition of the problem. Let G_1, G_2 be two comparability graphs such that $G_1 \cap G_2 = I$. Then G_1 and G_2 are simultaneously representable if there exists a transitive orientation T_1 of G_1 and a transitive orientation T_2 of G_2 such that the orientation of the edges of I is the same in T_1 and T_2.

Chordal graphs are intersection graphs of subtrees of some fixed tree. Two chordal graphs G_1, G_2 are simultaneously representable if there exists a set of *augmentable edges* $A \subseteq \{\{u,v\} \mid u \in V(G_1) \setminus V(G_2), v \in V(G_2) \setminus V(G_1)\}$ such that $G_1 \cup G_2 \cup A$ is a chordal graph.

For k chordal graphs where k is a part of the input, the problem is NP-complete. However, the time complexity of SimRep(CHOR) for fixed $k \geq 3$ is an open problem, even if $k = 3$.

We summarize the results from this paper in Table 3.1.

Subsequently, Jampani and Lubiw [38] found an algorithm for SimRep(INT) for $k = 2$ running in time $O(n^2 \log n)$. This algorithm uses PQ-trees. Later, Bläsius and Rutter [6] found a faster algorithm running in linear time, again using the PQ-tree approach.

Very recently, Rutter et al. [61] gave a linear time algorithm for the simultaneous representation problem for k proper interval graphs in sunflower position where k is a part of the input. They also provided a polynomial time algorithm for the simultaneous representation problem for k unit interval graphs $G_1, \ldots, G_k$ in sunflower position where k is a part of the input, running in time $O(|V(G_1) \cup \ldots \cup V(G_k)| \cdot |E(G_1) \cup \ldots \cup E(G_k)|)$.

Finally, they proved NP-completeness of the problems SIMREP(UNIT INT) and SIMREP(PROPER INT) for the case of k graphs in non-sunflower position where k is a part of the input. Independently on us, they used very similar reductions as our reductions in the next section.

We point the reader to the PhD thesis of Jampani [36] for a broader introduction to the simultaneous representation problem. A short summary can be also found in [67].

3.3 Our results

One of the interesting open problems is determining the complexity of SIMREP(INT) for k graphs in sunflower position where $k \geq 3$ is fixed or it is a part of the input.

There are no results even for the case where $k = 3$ is fixed. We found examples of interval graphs G_1, G_2, G_3 such that each two of them are simultaneously representable, but all three of them together are not simultaneously representable. An example of such graphs G_1, G_2, G_3 is shown in Figure 3.4, together with a simultaneous representation for each pair of them.

Let us explain the situation in the figure. We denote the union of G_1, G_2, G_3 by G. Clearly, G is not an interval graph, since it contains an asteroidal triple on vertices $5, 6, 7$. Now, if there is a subset of all possible edges going between vertices of different non-black colors, then all three graphs are simultaneous. However, by adding any such subset of edges into G, we see that the resulting graph has an induced cycle on at least 4 vertices, which is again a forbidden induced subgraph of interval graphs. We conclude that G_1, G_2, G_3 are non-simultaneous.

Note, that there are a lot of such examples. See Figure 3.5 for some of them. In each of them, the black part is the common part for all of them and non-black vertices and edges always belong to exactly one of the input graphs. We shall not describe what makes these instances non-simultaneous — the principle is very similar to the one in Figure 3.4.

In the rest of this section, we will show that the simultaneous representation problem for k interval or k circular-arc graphs in non-sunflower position where k is a part of the input is NP-complete.

For both NP-completeness results, we use a reduction from the problem called *total ordering*. Opatrny proved that TOTALORDERING is NP-complete [56].

PROBLEM:	TOTALORDERING – Total Ordering Problem
INPUT:	A finite set S and a finite set T of triples from S.
QUESTION:	Does there exist a total ordering $<$ of S such that for all triples $(x, y, z) \in T$, either $x < y < z$ or $x > y > z$?

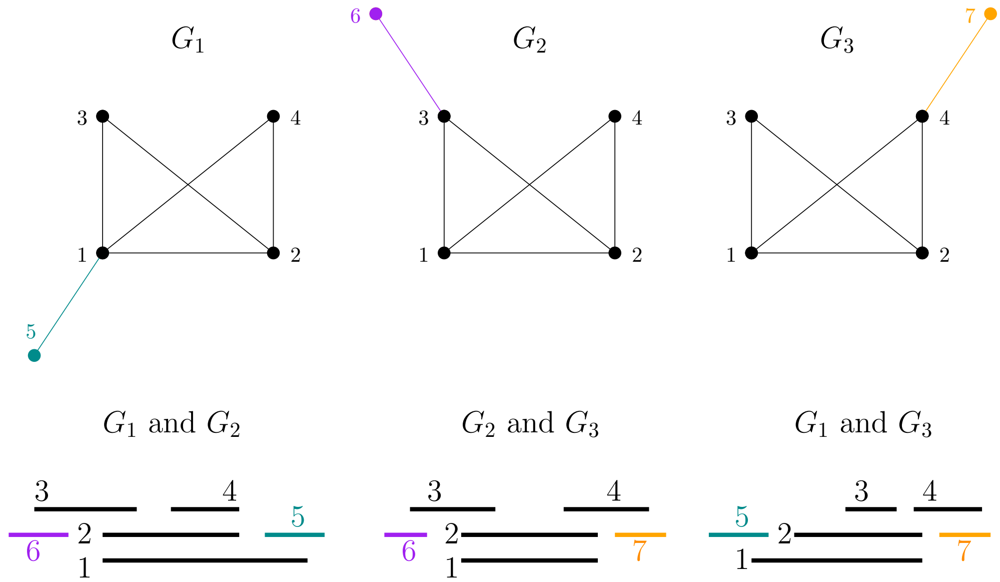


Figure 3.4: An example of pairwise simultaneous interval graphs, but all three of them together are not simultaneous.

3.3.1 Interval graphs

Firstly, we give a precise formulation of the problem.

PROBLEM:	SIMREP(INT) – Simultaneous representation problem for interval graphs
INPUT:	Interval graphs $G_1, \dots, G_k$.
QUESTION:	Do there exist interval representations $R_1, \dots, R_k$ of $G_1, \dots, G_k$ such that $$\forall i, j \in \{1, \dots, k\}\ \forall v \in G_i \cap G_j : R_i(v) = R_j(v)?$$

We are now ready to state and prove the main theorem of this section.

Theorem 3.1. *The problem* SIMREP*(*INT*) for k interval graphs in non-sunflower position where k is not fixed is* NP*-complete.*

Proof. The problem is clearly in NP as we can easily check in polynomial time if given representations are simultaneous.

Now, let I_{TO} be an instance of TOTALORDERING. Let us set s to be $|S|$ and t to be $|T|$. We denote by (x_i, y_i, z_i) each triple for $i \in \{1, \dots, t\}$. We will construct an instance I_S of SIMREP(INT).

We define graphs $G_0, G_1, \dots, G_t$ in the following way.

- $G_0 := (S, \emptyset)$,
- $G_i := (V_i, E_i)$ for each $0 < i \leq t$, where
  - $V_i := \{x_i, y_i, z_i, a_i, b_i, c_i\}$,

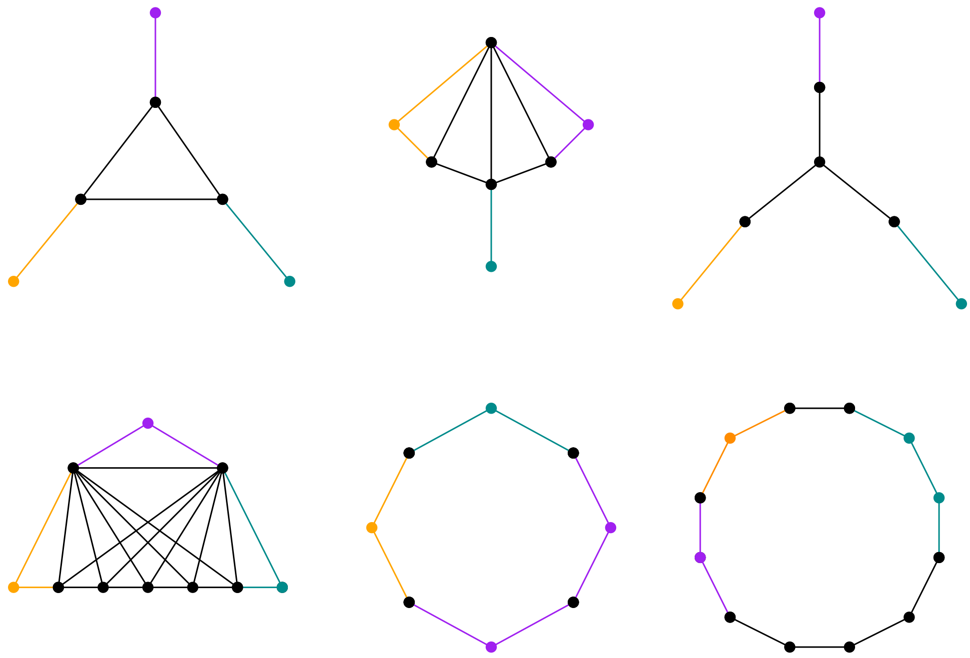


Figure 3.5: Six examples of union of three non-simultaneous graphs which are pairwise simultaneous.

– $E_i := \{x_ib_i, y_ib_i, z_ib_i, x_ia_i, z_ic_i\}$.

We observe that graphs $G_k, k = 0, \ldots, t$ are interval graphs and thus this is indeed an instance of SimRep(INT). See Example 1 and Figure 3.6 for an illustration of this construction.

In every representation of the graph G_i, where $i \in \{1, \ldots, t\}$, vertices x_i, y_i, z_i are represented by disjoint intervals, since these three vertices form an independent set. Furthermore, the vertex y_i is always completely to the right of either x_i or z_i and completely to the left of z_i or x_i, respectively.

Now, we can check that the following properties hold.

- $G_0 \cap G_i = \{x_i, y_i, z_i\}$ for each $1 \leq i \leq t$,
- $G_i \cap G_j = G_i \cap G_j \cap G_0 = \{v \in S | v \in (x_i, y_i, z_i) \wedge v \in (x_j, y_j, z_j)\}$ for each $1 \leq i < j \leq t$.

We observe that these $t + 1$ graphs are simultaneous if and only if the original instance of TotalOrdering has a solution. Since vertices of G_0 form an independent set, we can read the linear ordering $<$ of S from the representation of its corresponding vertices in G_0 by sweeping their left endpoints from the left to the right.

Thus, NP-completeness is established. □

Example 1. For an instance I_{TO} of TotalOrdering, where

$$S = \{1, 2, 3, 4, 5\},$$
$$T = \{(5, 1, 2), (2, 4, 3), (1, 4, 3)\},$$

we build an instance I_S of SimRep(INT) as in Figure 3.6.

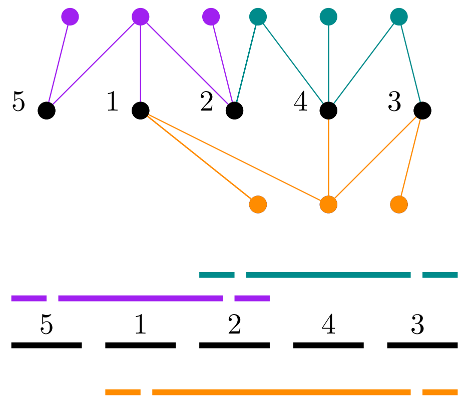


Figure 3.6: In the top: A union of graphs for triples from Example 1. In the bottom: Their simultaneous interval representation.

3.3.2 Circular-arc graphs

PROBLEM:	SIMREP(CA) – Simultaneous representation problem for circular-arc graphs
INPUT:	Circular-arc graphs $G_1, \ldots, G_k$.
QUESTION:	Do there exist circular-arc representations $R_1, \ldots, R_k$ of $G_1, \ldots, G_k$ such that

$$\forall i, j \in \{1, \ldots, k\}\ \forall v \in G_i \cap G_j : R_i(v) = R_j(v)?$$

Theorem 3.2. *The problem* SIMREP(CA) *for k graphs in non-sunflower position where k is not fixed is* NP*-complete.*

Proof. Again, the problem is in NP from obvious reasons.

We will proceed in a similar way as in the proof of previous theorem. We define graphs $G_0, G_1, \ldots, G_t$ in the same way as before and we add one common isolated vertex x to every graph, i.e. $x \in \bigcap_{i=0}^{t} V(G_i)$. The vertex x takes the role of breaking the cycle into a segment. Thus the circular arc representation of $G \setminus \{x\}$ is an interval representation and we can argue the rest as for interval graphs. $\square$

Chapter 4

Conclusion

Geometrically representable graphs are a very active and dynamic area of research. There are a lot of open problems as well. Let us summarize our results, emphasize some interesting open problems and present new problems which have arisen during the writing of this thesis.

Extending partial orderings. We defined and studied a new problem — the extension of partial orderings — motivated by the partial representation extension problem. We have reached the following results.

- We showed an algorithm for solving extending partial ordering of proper interval graphs running in time $O(n + m + |P|)$, where P is the set of pre-ordered pairs on the input.
- We managed to solve the extending partial ordering problem of adjusted interval digraphs in $O(n^5)$ time.

This is the first step to solve the extending partial representation problem of adjusted interval digraphs, which is the current work in progress.

The simultaneous representation problem. We continued in the study of the simultaneous representation problem and we contributed by the following results.

- We showed that $\textsc{SimRep}(\mathsf{INT})$ for k interval graphs in non-sunflower position where k is not fixed is NP-complete.
- Using similar techniques, we proved that $\textsc{SimRep}(\mathsf{CA})$ for k circular arc graphs in non-sunflower position where k is a part of the input is also NP-complete.

Rutter et al. [61] provided polynomial time algorithms for the simultaneous representation problem for k proper interval graphs and for k unit interval graphs in sunflower position where k is a part of the input. However, the problem is still open for k interval graphs in sunflower position for both cases — k fixed (and $k > 2$) and k being a part of the input.

Bibliography

[1] Allen, J. F. Maintaining knowledge about temporal intervals. *Communications of the ACM 26*, 11 (1983).

[2] Bang-Jensen, J., Huang, J., and Zhu, X. Completing orientations of partially oriented graphs. *Journal of Graph Theory 87*, 3 (2018), 285–304.

[3] Beineke, L. W., and Zamfirescu, C. M. Connection digraphs and second-order line digraphs. *Discrete Mathematics 39*, 3 (1982), 237–254.

[4] Benzer, S. On the topology of the genetic fine structure. *Proceedings of the National Academy of Sciences of the United States of America 45*, 11 (1959), 1607.

[5] Bläsius, T., Kobourov, S. G., and Rutter, I. Simultaneous embedding of planar graphs. *arXiv preprint arXiv:1204.5853* (2012).

[6] Bläsius, T., and Rutter, I. Simultaneous PQ-ordering with applications to constrained embedding problems. *ACM Transactions on Algorithms 12*, 2 (2016), 16.

[7] Bogart, K. P., and West, D. B. A short proof that "proper= unit". *arXiv preprint math/9811036* (1998).

[8] Booth, K. S., and Lueker, G. S. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences 13*, 3 (1976), 335–379.

[9] Chaplick, S., Fulek, R., and Klavík, P. Extending partial representations of circle graphs. In *International Symposium on Graph Drawing* (2013), Springer, pp. 131–142.

[10] Corneil, D. G., Olariu, S., and Stewart, L. Asteroidal triple-free graphs. *SIAM Journal on Discrete Mathematics 10*, 3 (1997), 399–430.

[11] Das, S., Sen, M., Roy, A., and West, D. B. Interval digraphs: An analogue of interval graphs. *Journal of Graph Theory 13*, 2 (1989), 189–202.

[12] Deng, X., Hell, P., and Huang, J. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM Journal on Computing 25*, 2 (1996), 390–403.

[13] Eschen, E. M. *Circular-arc graph recognition and related problems.* PhD thesis, Vanderbilt University, 1998.

[14] Feder, T., and Hell, P. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory, Series B 72*, 2 (1998), 236–250.

[15] Feder, T., Hell, P., and Huang, J. List homomorphisms and circular arc graphs. *Combinatorica 19*, 4 (1999), 487–505.

[16] Feder, T., Hell, P., and Huang, J. Bi-arc graphs and the complexity of list homomorphisms. *Journal of Graph Theory 42*, 1 (2003), 61–80.

[17] Feder, T., Hell, P., Huang, J., and Rafiey, A. Adjusted interval digraphs. *Electronic Notes in Discrete Mathematics 32* (2009), 83–91.

[18] Feder, T., Hell, P., Huang, J., and Rafiey, A. Interval graphs, adjusted interval digraphs, and reflexive list homomorphisms. *Discrete Applied Mathematics 160*, 6 (2012), 697–707.

[19] Fulkerson, D., and Gross, O. Incidence matrices and interval graphs. *Pacific Journal of Mathematics 15*, 3 (1965), 835–855.

[20] Galinier, P., Habib, M., and Paul, C. Chordal graphs and their clique graphs. In *International Workshop on Graph-Theoretic Concepts in Computer Science* (1995), Springer, pp. 358–371.

[21] Gavril, F. Algorithms on circular-arc graphs. *Networks 4*, 4 (1974), 357–369.

[22] Gilmore, P. C., and Hoffman, A. J. A characterization of comparability graphs and of interval graphs. *Canadian Journal of Mathematics 16* (1964), 539–548.

[23] Golumbic, M. C. Interval graphs and related topics. *Discrete Mathematics 55*, 2 (1985), 113–121.

[24] Golumbic, M. C. *Algorithmic graph theory and perfect graphs*, vol. 57 of *Annals of Discrete Mathematics.* Elsevier, 2004.

[25] Golumbic, M. C. Algorithmic graph theory and its applications. In *Graph Theory, Combinatorics and Algorithms.* Springer, 2005, pp. 41–62.

[26] Golumbic, M. C., Kaplan, H., and Shamir, R. On the complexity of DNA physical mapping. *Advances in Applied Mathematics 15*, 3 (1994), 251–261.

[27] Golumbic, M. C., Kaplan, H., and Shamir, R. Graph sandwich problems. *Journal of Algorithms 19*, 3 (1995), 449–473.

[28] Golumbic, M. C., and Shamir, R. Complexity and algorithms for reasoning about time: A graph-theoretic approach. *Journal of the ACM 40*, 5 (1993), 1108–1133.

[29] Gupta, U. I., Lee, D.-T., and Leung, J.-T. Efficient algorithms for interval graphs and circular-arc graphs. *Networks 12*, 4 (1982), 459–467.

[30] Hajós, G. Uber eine Art von Graphen. *Internationale mathematische nachrichten* (1957).

[31] Harary, F., Kabell, J. A., and McMorris, F. R. Bipartite intersection graphs. *Commentationes Mathematicae Universitatis Carolinae 23*, 4 (1982), 739–745.

[32] Hell, P., and Huang, J. Interval bigraphs and circular arc graphs. *Journal of Graph Theory 46*, 4 (2004), 313–327.

[33] Hell, P., Huang, J., McConnell, R. M., and Rafiey, A. Interval-like graphs and digraphs. *arXiv preprint arXiv:1804.05258* (2018).

[34] Hell, P., and Nešetřil, J. *Graphs and homomorphisms.* Oxford University Press, 2004.

[35] Hell, P., and Rafiey, A. The dichotomy of list homomorphisms for digraphs. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete Algorithms* (2011), Society for Industrial and Applied Mathematics, pp. 1703–1713.

[36] Jampani, K. R. *Simultaneous Graph Representation Problems.* PhD thesis, University of Waterloo, 2011.

[37] Jampani, K. R., and Lubiw, A. The simultaneous representation problem for chordal, comparability and permutation graphs. In *Workshop on Algorithms and Data Structures* (2009), Springer, pp. 387–398.

[38] Jampani, K. R., and Lubiw, A. Simultaneous interval graphs. In *International Symposium on Algorithms and Computation* (2010), Springer, pp. 206–217.

[39] Johnson, J. L., and Spinrad, J. P. A polynomial time recognition algorithm for probe interval graphs. In *SODA* (2001), vol. 1, pp. 477–486.

[40] Klavík, P. *Extension Properties of Graphs and Structures.* PhD thesis, Charles University, 2017.

[41] Klavík, P., Kratochvíl, J., Krawczyk, T., and Walczak, B. Extending partial representations of function graphs and permutation graphs. In *European Symposium on Algorithms* (2012), Springer, pp. 671–682.

[42] Klavík, P., Kratochvíl, J., Otachi, Y., Rutter, I., Saitoh, T., Saumell, M., and Vyskočil, T. Extending partial representations of proper and unit interval graphs. *Algorithmica 77*, 4 (2017), 1071–1104.

[43] Klavík, P., Kratochvíl, J., Otachi, Y., and Saitoh, T. Extending partial representations of subclasses of chordal graphs. *Theoretical Computer Science 576* (2015), 85–101.

[44] Klavík, P., Kratochvíl, J., Otachi, Y., Saitoh, T., and Vyskočil, T. Extending partial representations of interval graphs. *Algorithmica 78*, 3 (2017), 945–967.

[45] Klavík, P., Kratochvíl, J., and Vyskočil, T. Extending partial representations of interval graphs. In *International Conference on Theory and Applications of Models of Computation* (2011), Springer, pp. 276–285.

[46] Korte, N., and Möhring, R. H. An incremental linear-time algorithm for recognizing interval graphs. *SIAM Journal on Computing 18*, 1 (1989), 68–81.

[47] Krokhin, A., Jeavons, P., and Jonsson, P. Reasoning about temporal relations: The tractable subalgebras of Allen's interval algebra. *Journal of the ACM 50*, 5 (2003), 591–640.

[48] Lekkerkerker, C., and Boland, J. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae 51*, 1 (1962), 45–64.

[49] Lin, I.-J., Sen, M. K., and West, D. B. Classes of interval digraphs and 0, 1-matrices. *Congressus Numerantium* (1997), 201–210.

[50] Lin, M. C., and Szwarcfiter, J. L. Characterizations and recognition of circular-arc graphs and subclasses: A survey. *Discrete Mathematics 309*, 18 (2009), 5618–5635.

[51] McConnell, R. M. Linear-time recognition of circular-arc graphs. *Algorithmica 37*, 2 (2003), 93–147.

[52] McConnell, R. M., and Nussbaum, Y. Linear-time recognition of probe interval graphs. In *European Symposium on Algorithms* (2009), Springer, pp. 349–360.

[53] Möhring, R. H. Algorithmic aspects of comparability graphs and interval graphs. In *Graphs and Order*. Springer, 1985, pp. 41–101.

[54] Müller, H. Recognizing interval digraphs and interval bigraphs in polynomial time. *Discrete Applied Mathematics 78*, 1-3 (1997), 189–205.

[55] Olariu, S. An optimal greedy heuristic to color interval graphs. *Information Processing Letters 37*, 1 (1991), 21–25.

[56] Opatrny, J. Total ordering problem. *SIAM Journal on Computing 8*, 1 (1979), 111–114.

[57] Rafiey, A. Recognizing interval bigraphs by forbidden patterns. *arXiv preprint arXiv:1211.2662* (2012).

[58] Ramalingam, G., and Rangan, C. P. A unified approach to domination problems on interval graphs. *Information Processing Letters 27*, 5 (1988), 271–274.

[59] Roberts, F. S. *Representations of indifference relations.* Department of Mathematics, Stanford University, 1968.

[60] Roberts, F. S. Indifference graphs. *Proof techniques in graph theory* (1969), 139–146.

[61] Rutter, I., Strash, D., Stumpf, P., and Vollmer, M. Simultaneous representation of proper and unit interval graphs. In *EUROCG* (2019).

[62] Scheinerman, E. R. Characterizing intersection classes of graphs. *Discrete mathematics 55*, 2 (1985), 185–193.

[63] Sen, M., and Sanyal, B. K. Indifference digraphs: A generalization of indifference graphs and semiorders. *SIAM Journal on Discrete Mathematics 7*, 2 (1994), 157–165.

[64] Soulignac, F. J. Bounded, minimal, and short representations of unit interval and unit circular-arc graphs. Chapter II: algorithms. *Journal of Graph Algorithms and Applications 21*, 4 (2017), 491–525.

[65] Spinrad, J. P. *Efficient Graph Representations: The Fields Institute for Research in Mathematical Sciences.*, vol. 19. American Mathematical Society, 2003.

[66] Takaoka, A. A recognition algorithm for adjusted interval digraphs. *arXiv preprint arXiv:1810.06162* (2018).

[67] Tamassia, R. *Handbook of graph drawing and visualization.* Chapman and Hall/CRC, 2013.

[68] Tarjan, R. Depth-first search and linear graph algorithms. *SIAM journal on Computing 1*, 2 (1972), 146–160.

[69] Tucker, A. Matrix characterizations of circular-arc graphs. *Pacific Journal of Mathematics 39*, 2 (1971), 535–545.

[70] Tucker, A. An efficient test for circular-arc graphs. *SIAM Journal on Computing 9*, 1 (1980), 1–24.

[71] West, D. B. *Introduction to graph theory*, vol. 2. Prentice hall Upper Saddle River, NJ, 1996.

[72] West, D. B. Short proofs for interval digraphs. *Discrete Mathematics 178*, 1-3 (1998), 287–292.

[73] Zhang, P. Probe interval graph and its applications to physical mapping of DNA. *Manuscript* (1994).

List of Figures