FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

MASTER THESIS

Jana Novotná

Computational and structural apects of interval graphs and their variants

Department of Applied Mathematics

Supervisor of the master thesis:	prof. RNDr. Jan Kratochvíl, CSc.
Study programme:	Computer Science
Study branch:	Computational Linguistics

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 .... date .... signature of the author

I would like to thank my advisor, prof. Jan Kratochvíl, for his time and guidance. Furthermore, I would like to thank everyone who contributed to the pleasant, self-developing, and motivating atmosphere at the computer science section of the faculty during my entire studies. Big thanks belong also to my family, especially to my father, who supported me not only financially, and to my mother, for encouraging me in studying what I want. Last but not least, I would like to thank Tomáš Masařík for his patience and all the support he gave me.

Title: Computational and structural apects of interval graphs and their variants

Author: Jana Novotná

Department: Department of Applied Mathematics

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

Abstract: Interval graphs, intersection graphs of segments on a real line (intervals), play a key role in the study of algorithms and special structural properties. Unit interval graphs, their proper subclass, where each interval has a unit length, has also been extensively studied. We study mixed unit interval graphs—a generalization of unit interval graphs where each interval has still a unit length, but intervals of more than one type (open, closed, semi-closed) are allowed. This small modification captures a much richer class of graphs. In particular, mixed unit interval graphs are not claw-free, compared to unit interval graphs.

Heggernes, Meister, and Papadopoulos defined a representation of unit interval graphs called the bubble model which turned out to be useful in algorithm design. We extend this model to the class of mixed unit interval graphs. The original bubble model was used by Boyaci, Ekim, and Shalom for proving the polynomiality of the MaxCut problem on unit interval graphs. However, we found a significant mistake in the proof which seems to be hardly repairable. Moreover, we demonstrate advantages of such model by providing a subexponential-time algorithm solving the MaxCut problem on mixed unit interval graphs using our extended version of the bubble model. In addition, it gives us a polynomial-time algorithm for specific mixed unit interval graphs; that improves a state-of-the-art result even for unit interval graphs. We further provide a better algorithmic upper-bound on the clique-width of mixed unit interval graphs. Clique-width is one of the most general stuctural graph parameters, where a large group of natural problems is still solvable in the tracktable time when an efficient representation is given. Unfortunately, the exact computation of the clique-width representation is NP-hard. Therefore, good upper-bounds on the size of clique-width are highly appreciated, in particular, when such a bound is algoritmic.

Keywords: interval graph, unit interval graph, computational complexity, clique-width, maximum cut

Contents

Introduction

A graph G is an intersection graph if there exists a family of nonempty sets $\mathcal{F} = \{S_1, \dots, S_n\}$ such that for each vertex v_i in G, a set $S_i \in \mathcal{F}$ is assigned in a way that there is an edge v_iv_j in G if and only if $S_i \cap S_j \neq \emptyset$. Any graph can be represented as an intersection graph since per each vertex, we can use the set of its incident edges. However, many important graph classes can be described as intersection graphs with a restricted family of sets. Depending on the geometrical representation, different types of intersection graphs are defined, for instance, interval, circular-arc, disk graphs, etc.

Interval graphs are intersection graphs of segments of the real line, called intervals. They have been a well known and widely studied class of graphs from both the theoretical and the algorithmic points of view since 1957. They were first mentioned independently in combinatorics (Hajos, 1957[19, 6]) and in genetics (Benzer, 1959[1]).

Interval graphs have a nice structure, they are chordal and, therefore, also perfect which provides a variety of graph decompositions and models. Such properties are often useful tools for the algorithm design—the most common algorithms on them are based on dynamic programming. Therefore, many classical NP-hard problems are polynomial-time solvable on interval graphs, for instance Hamiltonian cycle (Keil 1985[23]), Graph isomorphism (Booth, 1976[3]) or Colorability (Golumbic, 1980 [16]) in linear time. Surprisingly, the complexity of some well-studied problems is still unknown despite an extensive research, e.g. the MaxCut problem, the $L_{2,1}$-labeling problem, or the packing coloring problem. Interval graphs have many real applications in diverse areas including genetics[1], economics, and archaeology[28, 29]. According to Golumbic[16], many real-world applications involve solving problems on graphs which are either interval graphs themselves or are related to interval graphs in a natural way.

An important subclass of interval graphs is the class of *proper interval graphs*, graphs which can be represented by such an interval representation that no interval properly contains another one. Another interval representation is a representation with intervals (of the same type) of only unit lengths, graphs which have such a representation are called *unit interval graphs*. Roberts proved in 1969 [27] that a graph is a proper interval graph if and only if it is a unit interval graph. Later, Gardi came up with a constructive combinatorial proof [14].

The mentioned results do not specifically care about what types of intervals (open, closed, semi-closed) are used in the interval representation. Note that as far as there are no restrictions on lengths of intervals, it does not matter which types of intervals are used. The same applies if there is only one type of intervals in the interval representation. However, this is not true when all intervals in the interval representation have unit length and at least two types of intervals are used. Recently, it has been observed that a restriction on different types of intervals in the unit interval representation leads to several new subclasses of interval graphs. By allowing all types of intervals of the unit length, a new proper superclass of unit interval graphs is created, we call the class *mixed unit interval graphs*.

Although there are 16 different combinations of types of unit intervals, it was shown in [10, 26, 31, 21, 32] in years 2012–2018 that they form only four different classes of mixed unit interval graphs. In particular, one of those classes is the class of unit interval graphs and none of them is the class of interval graphs itself.

There are lots of characterizations of interval and unit interval graphs. Among many of the characterizations, we single out a matrix-like representation called the *bubble model* [20] where vertices of a unit interval graph G are placed into a "matrix" where each matrix field may contain more vertices as well as it can be empty. Edges of G are represented implicitly with quite strong conditions: each column forms a clique; and in addition, there is an edge between two vertices u and v from different columns if and only if v occurs in a higher row than u in the column one to the right from u. In particular, there are no edges between non-consecutive columns. This representation can be computed and stored in linear space given a proper interval ordering representation.

1.1 Our Results

We introduce a similar representation, called the *$\mathcal{U}$-bubble model*, for mixed unit interval graphs. The representation has almost the same structure as the original bubble model, except that edges are allowed in the same row under specific conditions. We show that a graph is a mixed unit interval graph if and only if it can be represented by a $\mathcal{U}$-bubble model. This characterization allows us to work with a well-described structure of the $\mathcal{U}$-bubble model which turns out useful both for proving desired properties and for finding counterexamples. In addition, we show properties of our model, such as the relation of the size of a maximum independent set or maximum clique, and the size of the model.

Since mixed unit interval graphs lie between unit interval graphs and interval graphs, studying them can help us to characterize the finer differences between those two classes and possibly to understand the boundary between effectively solvable problems and hard problems. However, to my best knowledge, there is currently no problem known to be NP-hard on interval graphs and polynomial-time solvable on unit interval graphs. Moreover, no problem is known to be NP-hard or polynomial-time solvable on (unit) interval graphs without knowing it on both those classes.

Despite those facts, there are some partial results on unit interval graphs which can be generalized to mixed unit interval graphs. The problem I have chosen to demonstrate the techniques upon is the MaxCut problem. Given a graph G, the MaxCut problem is a problem of finding a partition of vertices of G into two sets S and $\overline{S}$ such that the number of edges with one endpoint in S and the other one in $\overline{S}$ is maximum among all partitions.

The MaxCut problem seemed to be an ideal candidate since it was claimed to be polynomial-time solvable in unit interval graphs [4] whereas its time complexity is unknown on interval graphs. There were two results about polynomiality of the MaxCut problem in unit interval graphs in the past years; the first one by Bodlaender, Kloks, and Niedermeier in 1999 [2], the second one by Boyaci, Ekim, and Shalom which has been published in 2017 [4]. Surprisingly, the first paper was disproved by authors themselves a few years later. In the second paper, authors used a bubble model for proving the polynomiality. However, we realised that their algorithm is not correct, as well. Moreover, it seems to be hardly repairable. We provide an example where their algorithm fails and also further discussion about the proof.

Using the $\mathcal{U}$-bubble model, we obtain at least a subexponential-time algorithm for MaxCut in mixed unit interval graphs. Furthermore, we obtain a polynomial-time algorithm if the given graph has a $\mathcal{U}$-bubble model with a constant number of columns. This extends a result by Boyaci, Ekim and Shalom [5] who showed a polynomial-time algorithm for MaxCut on unit interval graphs which have a bubble model with two columns (also called co-bipartite chain graphs). The question whether the MaxCut problem is polynomial-time solvable or NP-hard in unit interval graphs still remains open.

The third part of the thesis is devoted to clique-width, one of graph parameters that is used to measure the complexity of a graph. Many NP-hard problems can be solved efficiently on graphs with bounded clique-width [8]. In general, it is NP-complete to compute the exact value of clique-width. Furthermore, it is NP-complete even to decide if the graph has clique-width at most k for a given number k, see [12].

Although, unit interval graphs are known to have unbounded clique-width, in [17], there are results indicating its upper-bound. It follows from results by Fellows et all., and Kaplan and Shamir [11, 22] that the clique-width of (mixed) unit interval graphs is upper-bounded by ω (the maximum size of their clique) $+1$. This result was improved for unit interval graphs using the bubble model by Heggernes et all. [20]; the upper-bound there is a minimum of α (the maximum size of an independent set) $+ 1$, and a parameter related to the bubble model representation which is in the worst case $\omega+1$. We use similar ideas to extend these bounds to mixed unit interval graphs using the $\mathcal{U}$-bubble model. In particular, we obtain that the upper-bound on clique-width is the minimum of the analogously defined parameter for a $\mathcal{U}$-bubble model and $2\alpha + 3$. The upper-bound is still in the worst case $\omega + 1$. We further discuss the relation between the size of the $\mathcal{U}$-bubble model and upper-bounds on clique-width.

1.2 Structure of the paper

Chapter 2 is devoted to a summary of selected known results for the class of mixed unit interval graphs and its subclasses. The remaining chapters address our own results. The construction of $\mathcal{U}$-bubble model is shown in Chapter 3 together with the main theorem that a graph is a mixed unit interval graph if and only if it has a $\mathcal{U}$-bubble model. In Section 3.4, basic properties of $\mathcal{U}$-bubble model with respect to the size of a maximum clique and a maximum independent set are posted. Next, Chapter 4 stands for the upper-bounds on clique-width of mixed unit interval graphs. Finally, the mistake in the original algorithm for the MaxCut problem in unit interval graphs is discussed in Chapter 5 as well as the subexponential-time algorithm for the MaxCut problem in mixed unit interval graphs.

1.3 Preliminaries and Notation

By a *graph* we mean finite, undirected graph without loops and multiedges. Let G be a graph. We denote $V(G)$ and $E(G)$ the vertex and edge set of G, respectively; with $n = |V(G)|$ and $m = |E(G)|$. Let $\alpha(G)$ and $\omega(G)$ denote the maximum size of an independent set of G and the maximum size of a clique in G, respectively. Let $\mathcal{H}$ be a set of graphs. We say that G is $\mathcal{H}$-free if for each $H \in \mathcal{H}$, H is not an induced subgraph of G. The graph $K_{1,3}$ is called *claw*.

Let $x, y \in \mathbb{R}$ be real numbers. We call the set $\{z \in \mathbb{R} : x \leq z \leq y\}$ *closed interval* $[x, y]$, the set $\{z \in \mathbb{R} : x < z < y\}$ *open interval* (x, y), the set $\{z \in \mathbb{R} : x < z \leq y\}$ *open-closed interval* $(x, y]$, and the set $\{z \in \mathbb{R} : x \leq z < y\}$ *closed-open interval* $[x, y)$. By *semi-closed interval* we mean interval which is open-closed or closed-open. We denote the set of all open, closed, open-closed, and closed-open intervals by $\mathcal{I}^{--}$, $\mathcal{I}^{++}$, $\mathcal{I}^{-+}$, and $\mathcal{I}^{+-}$, respectively. Similarly, we denote the set of all open, closed, open-closed, and closed-open intervals of *unit length* by $\mathcal{U}^{--}$, $\mathcal{U}^{++}$, $\mathcal{U}^{-+}$, and $\mathcal{U}^{+-}$ respectively. Formally,

$$\begin{aligned}
\mathcal{I}^{++} &:= \{[x, y] : x, y \in \mathbb{R},\ x < y\} \text{ and } \mathcal{U}^{++} := \{[x, x+1] : x \in \mathbb{R}\},\\
\mathcal{I}^{--} &:= \{(x, y) : x, y \in \mathbb{R},\ x < y\} \text{ and } \mathcal{U}^{--} := \{(x, x+1) : x \in \mathbb{R}\},\\
\mathcal{I}^{+-} &:= \{[x, y) : x, y \in \mathbb{R},\ x < y\} \text{ and } \mathcal{U}^{+-} := \{[x, x+1) : x \in \mathbb{R}\},\\
\mathcal{I}^{-+} &:= \{(x, y] : x, y \in \mathbb{R},\ x < y\} \text{ and } \mathcal{U}^{++} := \{(x, x+1] : x \in \mathbb{R}\}.
\end{aligned}$$

We further denote the set of all (unit) open-closed intervals by

$$\mathcal{I}^{\pm} := \mathcal{I}^{++} \cup \mathcal{I}^{--} \text{ and } \mathcal{U}^{\pm} := \mathcal{U}^{++} \cup \mathcal{U}^{--},$$

and the sets of all (unit) intervals by

$$\mathcal{I} := \mathcal{I}^{++} \cup \mathcal{I}^{--} \cup \mathcal{I}^{+-} \cup \mathcal{I}^{-+} \text{ and } \mathcal{U} := \mathcal{U}^{++} \cup \mathcal{U}^{--} \cup \mathcal{U}^{+-} \cup \mathcal{U}^{-+}.$$

The *type of an interval* I is a pair (r, s) where $I \in \mathcal{I}^{r,s}$, $r, s \in \{+, -\}$. Let I be an interval, we define the left and right end of I as $\ell(I) := \inf(I)$ and $r(I) := \sup(I)$, respectively. Let $I, J \in \mathcal{U}$ be unit intervals, I, J are almost twins if $\ell(I) = \ell(J)$.

Let $\mathcal{R}$ be a family of segments of the real line (real intervals). By family we mean a multiset $\{S_1, \ldots, S_n\}$ which allows the possibility that $S_i = S_j$ even though $i \neq j$. We say that a graph G has an $\mathcal{R}$-*intersection* representation if there exist a function $I : V(G) \to \mathcal{R}$ such that for every two distinct vertices u and v of G, we have an edge $uv \in E(G)$ if and only if $I(u) \cap I(v) \neq \emptyset$; we denote $I_u := I(u)$. A graph which has an $\mathcal{R}$-intersection representation is called $\mathcal{R}$-*graph*, in general *interval graph* and its representation is called, in general, *interval representation*.

Let $G = (V, E)$ be a graph and $\mathcal{I}$ an interval representation of G. Let $v \in V$ be represented by an interval $I_v \in \mathcal{I}^{r,s}$, where $r, s \in \{+, -\}$, in $\mathcal{I}$. The *type of a vertex* $v \in V$ in $\mathcal{I}$, denoted by $\text{type}_{\mathcal{I}}(v)$, is the pair (r, s). We use $\text{type}(v)$ if it is clear which interval representation we have in mind.

We follow the standard approach where the maximum over the empty set is $-\infty$.

The notion of $\tilde{\mathcal{O}}$ denotes the standard "big 0" notion which ignores polylogarithmic factors, i.e, $\mathcal{O}(f(n) \log^k n) = \tilde{\mathcal{O}}(f(n))$, where k is a constant.

2. Mixed unit interval graphs

We define formally classes of mixed unit interval graphs. We summarize results about their recognition and finding their intersection representation.

2.1 Subclasses of mixed unit interval graphs

Definition 1. *A graph G is a* mixed unit interval graph *if it has a $\mathcal{U}$-intersection representation. We call such representation a* mixed unit representation.

Roberts [27] proved in 1969 that a graph is a unit interval graph if and only if it is a proper interval graph if and only if it is a $K_{1,3}$-free interval graph. In 2019, Dourado et al. [10] showed that the first equivalence is true similarly for mixed unit interval graphs and mixed proper interval graphs.

Definition 2 ([10])**.** *A graph G is a* mixed proper interval graph *if it has an $\mathcal{I}$-intersection representation $I \colon V(G) \to \mathcal{I}$ such that*

- *there are no two distinct vertices u and v of G with $I(u), I(v) \in \mathcal{I}^{++}$ such that $I(u) \subseteq I(v)$ and $I(v) \nsubseteq I(u)$, and*
- *for every vertex u of G such that $I(u) \notin \mathcal{I}^{++}$, there is a vertex v of G with $I(v) \in \mathcal{I}^{++}$, $\ell(I(u)) = \ell(I(v))$, and $r(I(u)) = r(I(v))$.*

In other words, no closed interval properly contains another closed interval, and for every non-closed interval, there is a closed interval with the same ends. We call such representation a mixed proper interval representation.

Since a graph G is a mixed proper interval graph if and only if it is a mixed unit interval graph, from now, we will use only the term mixed unit interval graphs.

In the recent few years, researchers wanted to characterized classes of graphs with intersection representation that allows more than one type of intervals. Shuchat et al.[31] observed that a graph has an $\mathcal{I}$-intersection representation if and only if it is an interval graph. Therefore, if there are no restrictions on lengths of intervals, it does not matter which types of intervals are used, and we still end up with the class of interval graphs. When lengths of intervals are restricted to be the same, the situation is different. In particular, the claw $K_{1,3}$ can be represented using one open interval and three closed intervals. Therefore, in recent few years, researchers characterized all the subclasses of mixed unit interval graphs.

At the beginning, it was shown that the same class of graphs is obtained if we require all the unit intervals to be of the same type (either all closed, all open, or all semi-closed). In particular, Frankl and Machara [13] proved that the classes of $\mathcal{U}^{++}$-graphs and $\mathcal{U}^{--}$-graphs are the same. This result was restated in [26] by Rautenbach and Szwarcfiter. Dourado et al. [10] extended this result and showed that the classes of $\mathcal{U}^{++}$-graphs, $\mathcal{U}^{+-}$-graphs, $\mathcal{U}^{-+}$-graphs, $\mathcal{U}^{--}$-graphs, and $\mathcal{U}^{+-,-+}$-graphs are the same.

In 2018, Kratochvíl and Talon [32] proved that the classes of $\mathcal{U}^{++}$-graphs, $\mathcal{U}^{++,+-}$-graphs, $\mathcal{U}^{++,-+}$-graphs, $\mathcal{U}^{--,+-}$-graphs, $\mathcal{U}^{--,-+}$-graphs, $\mathcal{U}^{++,+-,-+}$-graphs, and $\mathcal{U}^{--,+-,-+}$-graphs are the same. This resulted in the following complete hierarchy of mixed unit interval graphs.

Theorem 1 (Kratochvíl, Talon [32])**.** *There are the following different subclasses of the class of mixed unit interval graphs:*

- *unit interval graphs:*
  $\mathcal{U}^{++}$*-graphs,* $\mathcal{U}^{--}$*-graphs,* $\mathcal{U}^{+-}$*-graphs,* $\mathcal{U}^{-+}$*-graphs,* $\mathcal{U}^{+-,-+}$*-graphs,* $\mathcal{U}^{++,+-}$*-graphs,* $\mathcal{U}^{++,-+}$*-graphs,* $\mathcal{U}^{--,+-}$*-graphs,* $\mathcal{U}^{--,-+}$*-graphs,* $\mathcal{U}^{++,+-,-+}$*-graphs, and* $\mathcal{U}^{--,+-,-+}$*-graphs,*

- *unit interval graphs of closed and open intervals:*
  $\mathcal{U}^{\pm}$*-graphs,*

- *semi-mixed unit interval graphs (sometimes called almost-mixed):*
  $\mathcal{U}^{\pm,+-}$*-graphs, and* $\mathcal{U}^{\pm,-+}$*-graphs,*

- *mixed unit interval graphs*
  $\mathcal{U}$*-graphs.*

The following closure holds:

$$\emptyset \subsetneq \textit{unit interval} \subsetneq \textit{unit open and closed interval} \subsetneq \textit{semi-mixed unit interval} \subsetneq \\ \subsetneq \textit{mixed unit interval} \subsetneq \textit{interval graphs.}$$

2.2 Recognition and $\mathcal{U}$-representation of mixed unit graphs

All the classes of mixed unit interval graphs can be characterized using forbidden induced subgraphs, sometimes by infinitely many. Rautenbach and Szwarcfiter [26] gave a characterization of $\mathcal{U}^{\pm}$-graphs using five forbidden induced subgraphs. Joos [21] gave a characterization of twin-free mixed unit interval graphs by an infinite class of forbidden induced subgraphs. Shuchat, Shull, Trenk and West [31] proved independently also this characterisation, moreover, they complemented it by a quadratic-time algorithm which produces a mixed proper interval representation. Finally, Kratochvíl and Talon [32] characterized the remaining classes.

Le and Rautenbach [25] characterized graphs that have a mixed unit interval representations in which all intervals have integer endpoints, and provided a quadratic-time algorithm that decides whether a given interval graph admits such a representation.

We refer the reader to the original papers for more details and concrete forbidden subgraphs.

Moreover, there are nice structural results of the subclasses of mixed unit interval graphs. For example, it is shown in [26] that for $\mathcal{U}^{\pm}$-intersection representations, open intervals are only really needed to represent claws, in particular, for any $\mathcal{U}^{\pm}$-graph there exist a $\mathcal{U}^{\pm}$-representation such that for every open interval, there is a closed interval with the same endpoints. More structural results can be found in [32].

Theorem 2 ([32])**.** *The classes of semi-mixed and mixed unit interval graphs can be recognized in time* $\mathcal{O}(n^2)$*. Moreover, there exists an algorithm which takes a graph* $G \in \mathcal{U}^{\pm,+-}$ *on input, and outputs a corresponding* $\mathcal{U}^{\pm,+-}$*-representation of* G *in time* $\mathcal{O}(n^2)$*.*

Claim 3 ([32]). *It is possible to modify the algorithm for semi-mixed unit interval graphs such that given a mixed unit interval graph G, it outputs a mixed unit interval representation of G in time $\mathcal{O}(n^2)$.*

3. Bubble model for mixed unit interval graphs

In this section, we present a $\mathcal{U}$-bubble model, a new representation of mixed unit interval graphs which is inspired by the notion of bubble model for proper interval graphs created by Heggernes, Meister, and Papadopoulos [20] in 2015.

3.1 Definition of bubble model

First, we present the bubble model for proper interval graphs as it was introduced by Heggernes et al.

Definition 3 (Heggernes et al. [20], reformulated)**.** *If A is a finite non-empty set, then a* 2-dimensional $\mathcal{U}$-bubble structure *for A is a partition $\mathcal{B} = \langle B_{i,j} \rangle_{1 \le j \le k, 1 \le i \le r_j}$, where $A = \bigcup_{i,j} B_{i,j}$, $\emptyset \subseteq B_{i,j} \subseteq A$ for every i, j with $1 \le j \le k$ and $1 \le i \le r_j$, and $B_{1,1} \dots B_{k,r_k}$ are pairwise disjoint. The* graph given by $\mathcal{B}$*, denoted as $G(\mathcal{B})$, is defined as follows:*

1. *$G(\mathcal{B})$ has a vertex for every element in A, and*
2. *uv is an edge of $G(\mathcal{B})$ if and only if there are indices i, i', j, j' such that $u \in B_{i,j}$, $v \in B_{i',j'}$ and one of the two conditions holds: either $j = j'$ or $(i - i')(j - j') < 0$.*

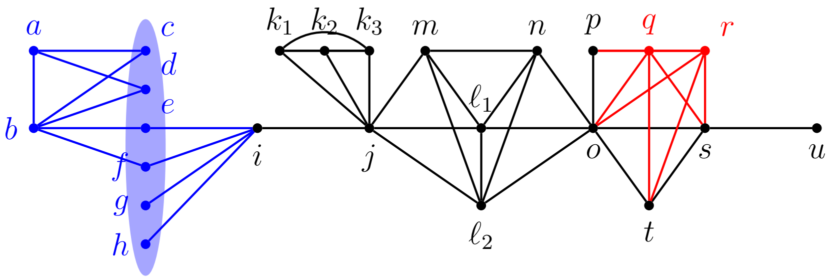
A bubble model *for a graph $G = (V, E)$ is a 2-dimensional bubbles structure $\mathcal{B}$ for V such that $G = G(\mathcal{B})$.*

Theorem 4 (Heggernes et al. [20])**.** *A graph is a proper interval graph if and only if it has a bubble model.*

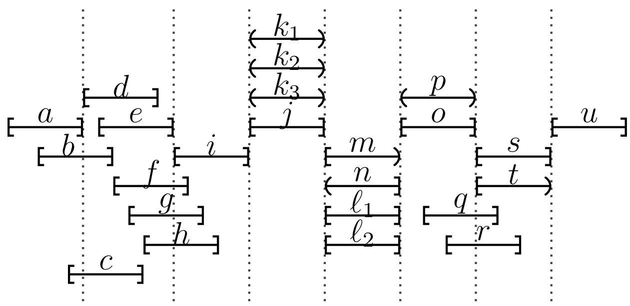
We define a similar matrix-type structure for mixed unit graphs where each bubble is split into four parts and edges are allowed also in the same row under specific conditions.

Definition 4. *Let A be a finite non-empty set and $\mathcal{B} = \langle B_{i,j} \rangle_{1 \le j \le k, 1 \le i \le r_j}$ be a 2-dimensional bubble structure for A such that $B_{i,j} = B_{i,j}^{++} \cup B_{i,j}^{+-} \cup B_{i,j}^{-+} \cup B_{i,j}^{--}$, $B_{i,j}^{r,s}$ are pairwise disjoint, and $\emptyset \subseteq B_{i,j}^{r,s} \subseteq B_{i,j}$ for every $r, s \in \{+, -\}$ and i, j with $1 \le j \le k$ and $1 \le i \le r_j$. We call the partition $\mathcal{B}$ a* 2-dimensional $\mathcal{U}$-bubble structure *for A.*

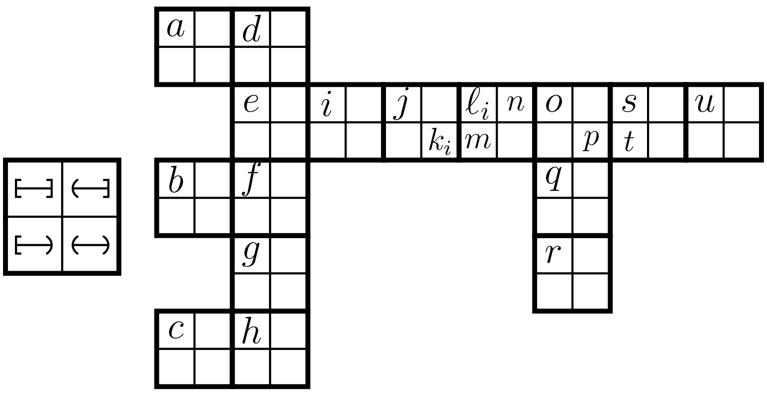
We call each set $B_{i,j}$ a *bubble*, and each set $B_{i,j}^{r,s}, r, s \in \{+, -\}$ a *quadrant* of the bubble $B_{i,j}$. The *type* of a quadrant $B_{i,j}^{r,s}$, $r, s \in \{+, -\}$ is the pair (r, s). We denote by $*$ both $+$ and $-$, for example $B_{i,j}^{*+} = B_{i,j}^{-+} \cup B_{i,j}^{++}$. Bubbles with the same i-index form a *row* of $\mathcal{B}$, and with the same j-index a *column* of $\mathcal{B}$, we say vertices from bubbles $B_{i,1} \cup \ldots \cup B_{i,k}$ *appear in row* i, and we denote i as their *row-index*. We define an analogical notion for columns. We denote the index of the first row with a non-empty bubble as $\operatorname{top}(j) := \min \{i \mid B_{i,j} \in \mathcal{B} \text{ and } B_{ij} \neq \emptyset\}$. Thus, $B_{\operatorname{top}(j),j}$ is the first non-empty bubble in the column j. Let B be a bubble, then $\operatorname{row}(B)$ and $\operatorname{col}(B)$ is the row-index and column-index of B, respectively. Let $u \in B_{i,j}$, $v \in B_{i',j'}$; we say that u is *under* than v and v is *above* u if $i > i'$.



(a) Graph G; the blue ellipse denotes clique $cdefgh$; colors are used only for clarity.



(b) A mixed unit interval representation of G.



(c) A $\mathcal{U}$-bubble model of G.

Figure 3.1: A mixed unit interval graph G.

Definition 5. *Let $\mathcal{B} = \langle B_{i,j} \rangle_{1 \le j \le k, 1 \le i \le r_j}$ be a 2-dimensional $\mathcal{U}$-bubble structure for A. The graph* given by $\mathcal{B}$, *denoted as $G(\mathcal{B})$, is defined as follows:*

1. *$V(G(\mathcal{B})) = A$,*
2. *uv is an edge of $G(\mathcal{B})$ if and only if there are indices i, i', j, j' such that $u \in B_{i,j}$, $v \in B_{i',j'}$, or $v \in B_{i,j}$, $u \in B_{i',j'}$, and one of the three conditions holds:*
   (a) $j = j'$, or
   (b) $j = j' - 1$ and $i > i'$, or
   (c) $j = j' - 1$ and $i = i'$ and $u \in B^{*+}_{i,j}, v \in B^{+*}_{i',j'}$.

The definition says that the edges are only between vertices from the same or consecutive columns and if $u \in B_{i,j}$ and $v \in B_{i',j+1}$, there is an edge between u and v if and only if u is lower than v $(i > i')$, or they are in the same row and $u \in B^{*+}_{i,j}, v \in B^{+*}_{i',j+1}$

Observation 1. *Vertices from the same column form a clique, as well as vertices from the same bubble. Moreover, vertices from the same bubble are almost-twins and their neighborhoods can differ only in the same row, anywhere else they behave as twins. Vertices from the same bubble quadrant are true twins.*

Definition 6. *Let $G = (V, E)$ be a graph. A $\mathcal{U}$*-bubble model *for a graph G is a 2-dimensional $\mathcal{U}$-bubble structure $\mathcal{B} = \langle B_{i,j}\rangle_{1\le j\le k, 1\le i\le r_j}$ for V such that*

(i) G is isomorphic to $G(\mathcal{B})$, and

(ii) each column and each row contains a non-empty bubble, and

(iii) no column ends with an empty bubble, and

(iv) $\mathrm{top}(1) = 1$, and for every $j \in \{1, \ldots, k-1\} : \mathrm{top}(j) \ge \mathrm{top}(j+1)$.

We define the *size* of a $\mathcal{U}$-bubble model $\mathcal{B} = \langle B_{i,j}\rangle_{1\le j\le k, 1\le i\le r_j}$ as the number of columns multiplied by the number of rows, i.e., $k \cdot \max\{r_j \mid 1 \le j \le k\}$.

See Figure 3.1 with an example of a mixed unit interval graph, given by a mixed unit interval representation, and by a $\mathcal{U}$-bubble model.

Theorem 5. *A graph is a mixed unit interval graph if and only if it has a $\mathcal{U}$-bubble model. Moreover, a $\mathcal{U}$-bubble model can be constructed in time $\mathcal{O}(n^2)$.*

Before we prove the theorem, we describe an algorithm which creates a $\mathcal{U}$-bubble model for a given mixed unit graph G.

3.2 Construction of $\mathcal{U}$-bubble model

First, we construct a mixed unit interval representation $\mathcal{I}$ of a graph G using the quadratic-time algorithm from [32]; then each vertex of G is represented by a corresponding interval. Given a mixed unit interval representation $\mathcal{I}$ of a graph G, our algorithm outputs a $\mathcal{U}$-bubble model for G. The algorithm processes vertices of G in the order given by $\mathcal{I}$ sorted by the left ends of the intervals starting from left to right. The vertices with the same left end are processed in an arbitrary order.

The algorithm works as follows. Start by putting the first vertex into the bubble in the first column and row. Then, put all the adjacent vertices into bubbles in the same column filling rows from top to bottom as long as the corresponding intervals intersect the first interval in more than one point. Put two vertices in the same bubble if and only if their corresponding intervals have the same left end. Otherwise, create a new bubble. For a bubble B denote by $\ell(B)$ and $r(B)$ the left and right ends of intervals corresponding to vertices in B, i.e., $\ell(B) := \ell(I_v)$ and $r(B) := r(I_v)$, for $v \in B$. Recall that the ends are same for all such vertices in one bubble. Always, put the vertex into that quadrant of the bubble which has the same type as the vertex. The row-indices of the created bubbles (in the first column) are set to be, in that time, natural numbers from 1 to the number of the bubbles in the first column.

Continue as follows. As soon as an interval I_v which has at most one point in common with the first interval from the current column, is reached, put v into the next column to a new bubble B as the first vertex in the next column, and continue analogously until you process all vertices. Assign the row-index of a new bubble B by the following rules, using the bubbles from the previous column. If there exist a bubble B' from the previous column such that $r(B') = \ell(B)$ (the corresponding interval of $v \in B$ intersects with intervals of vertices in B' in exactly one point), assign the row-index of B' to the row-index of B. Otherwise, let b be the row-index of such the highest bubble B' in the previous column that $r(B') > \ell(B)$. If no such bubble exists, set b to be the row-index of the last bubble in the previous column +1. Let a be the minimum of the row-index of such the lowest bubble B'' that $r(B'') < \ell(B)$ in the previous column, and the row-index of the last bubble in the actual column +1 (we are filling columns from top to bottom – no new vertex is assigned to the same column above an already assigned vertex).

Assign the row-index $(a + b)/2$ to the bubble B. In that point, row-indices are rational numbers instead of natural numbers. Note that a new vertex was assigned above or to the same row than all its neighbours and under than or in the same row as all its non-neighbors with respect to the previous column.

When all vertices are processed, it remains to adjust row-indices to be natural numbers. Exchange values of row-indices for their positions (starting by 1) in the natural order consisting of used values.

In the end, we can consider the bubble model as it has empty bubbles. For algorithmic purposes it is better to work without empty bubbles and store the bubble model into an appropriate structure to ensure that going over the vertices takes linear time (in number of vertices). We further note that the $\mathcal{U}$-bubble model can be modified to be more compact. However, it is not needed to obtain the results included in this thesis.

Observation 2. *Let I_u and I_v be intervals from $\mathcal{I}$ such that $I_u \cap I_v \neq \emptyset$. The algorithm puts vertices u and v into the same or consecutive columns in $\mathcal{B}$, i.e., if B_{ij} and $B_{i'j'}$ are the bubbles assigned to the vertices u and v, respectively, then $|j' - j| \leq 1$.*

Proof. Without loss of generality, assume $\ell(I_u) \leq \ell(I_v)$. If $\ell(I_u) = \ell(I_v)$ then the algorithm put the vertices u and v into the same bubble. It remains to prove the situation where $\ell(I_u) < \ell(I_v)$. For contradiction, suppose that $j' - j \geq 2$. The algorithm process I_u before I_v. It follows from the algorithm that

$$\begin{aligned} &\ell(I_u) < r(B_{\mathrm{top}(j),j}) \leq \ell(B_{\mathrm{top}(j+1),j+1}), \text{ and} \\ &r(B_{\mathrm{top}(j+1),j+1}) < \ell(I_v). \end{aligned}$$

Summarized:

$$r(I_u) < r(B_{\mathrm{top}(j),j}) + 1 \leq r(B_{\mathrm{top}(j+1),j+1}) < \ell(I_v).$$

Which leads to the contradiction with a non-empty intersection of I_u and I_v. $\square$

3.3 Proof of Theorem 5

Proof of Theorem 5. First, we prove the reverse implication, i.e., given a $\mathcal{U}$-bubble model for a graph G, we construct a mixed unit interval representation of G. Let $\mathcal{B} = \langle B_{i,j} \rangle_{1 \le j \le k, 1 \le i \le r_j}$ be a $\mathcal{U}$-bubble model of G. Let

$$\varepsilon := \frac{1}{\max\{r_j \mid 1 \le j \le k\}}.$$

We create a mixed unit interval representation $\mathcal{I}$ of G as follows. Let $v \in B_{i,j}^{r,s}$, where $r, s \in \{+, -\}$. The corresponding interval I_v of v has the properties:

$$I_v \in \mathcal{I}^{r,s} \text{ and } \ell(I_v) := j + (i-1)\varepsilon.$$

Note that all vertices from the same bubble are represented by intervals with the same positions and the type of an interval corresponds with the type of the bubble quadrant. Since ε was chosen such that $\varepsilon(i-1) < 1$ for any row i in $\mathcal{B}$, the graph given by the constructed mixed unit interval representation is isomorphic to the graph given by $\mathcal{B}$.

Now, we prove the second implication—every mixed unit interval graph has a $\mathcal{U}$-bubble model. We use the described constructive algorithm and prove its correctness and that it can be deterministically constructed. Let $\mathcal{I}$ be the used mixed unit interval representation of G. Obviously, there is a vertex in $G(\mathcal{B})$ for each interval $I_u \in \mathcal{I}$, i.e., for each vertex $u \in V(G)$. Furthermore, the properties (ii), (iii), and (iv) of Definition 6 are clearly satisfied. For the correctness of the algorithm, it remains to prove that there is an edge uv in $G(\mathcal{B})$ if and only if uv is an edge in G, which is equivalent to $I_u \cap I_v \neq \emptyset$.

First, suppose $I_u \cap I_v \neq \emptyset$, for $u, v \in V(G)$. We prove that there is an edge uv in $G(\mathcal{B})$ as follows. If $\ell(I_u) = \ell(I_v)$, the algorithm puts u and v into the same bubble. Therefore, u and v are adjacent in $G(\mathcal{B})$ by the definition. Without loss of generality $\ell(I_u) < \ell(I_v)$. The algorithm either put both u and v into the same column, or in different columns. In the first case, there is an edge between u and v in $G(\mathcal{B})$ by the definition. Let us focus on the second case where u and v are in different columns. By Observation 2, they are in two consecutive columns. Denote by $B_{i,j}$, $B_{i',j+1}$ such bubbles that $u \in B_{i,j}$, $v \in B_{i',j+1}$. It follows from the algorithm that $i \ge i'$. In the case of $i > i'$, u and v are adjacent in $G(\mathcal{B})$ by the definition of $G(\mathcal{B})$. In the case of $i' = i$, there is an edge between u and v if and only if $u \in B_{i,j}^{*+}$ and $v \in B_{i,j+1}^{+*}$ which is if and only if $|I_u \cap I_v| = 1$ since the algorithm puts vertices into bubble quadrants according their types.

Second, assume, there is an edge uv in $G(\mathcal{B})$ and $u \in B_{i,j}$, $v \in B_{i',j'}$. We prove that there is an edge uv in G as follows. We distinguish several situations according to i, i', j, j'. Let u, v be from the same column, $j = j'$. The algorithm put those two vertices in the column j since their intervals intersect intervals in the first bubble, i.e., it was true that $\ell(B_{\mathrm{top}(j),j}) \le \ell(I_u), \ell(I_v) < \ell(B_{\mathrm{top}(j),j}) + 1$. Therefore, $I_u \cap I_v \neq \emptyset$. Without loss of generality we assume $j < j'$. Since there are no edges between non-consecutive columns in $G(\mathcal{B})$, $j+1 = j'$. Let $i = i'$, then there is an edge uv if and only if $I_u \cap I_v \neq \emptyset$ by the definition and the fact that the algorithm puts vertices into bubble quadrants of the same type. It remains to consider the case where $j+1 = j'$, $i > i'$. Note that the algorithm put v between such the highest bubble B' in the column j that satisfies $r(B') > \ell(I_v)$, and such the lowest bubble B'' that satisfies $r(B'') < \ell(I_v)$. Since $r(B') > r(B'')$, the bubble B'' was processed earlier than B' and, thus, it is above the bubble B'. Therefore, all bubbles B under $B_{i',j+1}$ (especially B_{ij}) satisfy that $\ell(I_v) < r(B)$. This completed the proof for adjacency.

It remains to prove that $\mathcal{U}$-bubble model can be deterministically constructed according to the algorithm. The only problematic part is the height assignment, i.e., that any bubble can be placed to the right height. We prove that during the whole construction, any time we want to add a new vertex u to $\mathcal{B}$, there is the invariant $k \geq l$, where l is the row-index of the bubble with the biggest row-index in the last created column, i.e.,

$$l = \max\{i \mid B_{i,j} \text{ is already placed in column } j\}$$

and the column j is the last column of $\mathcal{B}$, and k is the row-index of the bubble with the lowest row-index in the previous column such that its vertices are adjacent with u in G,

$$k = \min\{i \mid r(B_{i,j-1}) \geq \ell(I_u))\}.$$

If $j = 1$, we can place all bubbles. Formally, we define k to be the infinity to satisfy the invariant. If u is the first vertex to be in the column $(j+1)$, than there is no problem with assigning the height to its bubble, as well. Formally, we define l to be 0. In any other case, if u is to be placed into an existing bubble (the last in the last column), i.e., it is an almost-twin, then the invariant is satisfied due to the previous step. It remains to consider the situation where u should be placed into the column j. We know:

$$\ell(I_u) > \ell(B_{l,j}) \text{ (from the linear order on } \mathcal{I}\text{) and}$$
$$r(B_{k,j-1}) \geq \ell(I_u) \text{ (from the definition of } k\text{).}$$

Which leads to $r(B_{k,j-1}) > \ell(B_{l,j})$, therefore $k > l$ due to the previous step.

It remains to prove that $\mathcal{U}$-bubble model can be constructed in polynomial time. The mixed unit interval representation can be found in quadratic time in number of vertices according to Theorem 2 and Claim 3. It is clear that the algorithm itself does not take more than quadratic time since each vertex needs only to determine the values a and b (for which linear time is enough), and also quadratic time is enough to assign integer values to the row-indices. This concludes the proof of Theorem 5. □

3.4 Properties of $\mathcal{U}$-bubble model

In this section, we give basic properties of a $\mathcal{U}$-bubble model which are used later in the text. It is readily seen that a $\mathcal{U}$-bubble model of graph $G = (V, E)$ has at most n rows and n columns where n is the number of vertices of G since each column and each row contains at least one vertex. Consequently, the size of a $\mathcal{U}$-bubble model is at most n^2.

Two basic characterisations of a graph are the size of a maximum clique and the size of a maximum independent set in the graph. The problem of finding those numbers is NP-complete in general but it is polynomial-time solvable in interval graphs. We show a relation between those two numbers and the size of a

$\mathcal{U}$-bubble model for the graph. We start with the size of a maximum independent set.

Lemma 6. *Let G be a mixed unit interval graph, and let $\mathcal{B}$ be a $\mathcal{U}$-bubble model for G. The number of columns of $\mathcal{B}$ is at least $\alpha(G)$ and at most $2\alpha(G)$.*

Proof. Let I be a maximum independent set of G, and let k be the number of columns of $\mathcal{B}$. We have that $\alpha(G) \geq \lceil k/2 \rceil$ from the property that two non-consecutive columns from $\mathcal{B}$ are not adjacent in $G(\mathcal{B})$.

Since each column forms a clique, only one vertex from each column can be in I. Therefore, $\alpha(G) \leq k$. □

In the bubble model for unit interval graphs, $\alpha(G)$ is equal to the number of columns [20]. However, the gap in Lemma 6 cannot be narrowed in general—consider an even number k and the following unit interval graphs: path on k-vertices (P_k) and a clique on k vertices (K_k). There exists a unit interval representation of P_k using only closed intervals which leads to a $\mathcal{U}$-bubble model of P_k containing one row and k columns, where $\alpha(P_k) = \lceil k/2 \rceil$. A $\mathcal{U}$-bubble model of K_k contains k rows and one column, where $\alpha(K_k) = 1 =$ number of columns.

Another important and useful property of graphs is the size of a maximum clique. We show that a maximum clique of a mixed unit interval graph can be found in two consecutive columns of a $\mathcal{U}$-bubble model of the graph, see Figure 3.2.

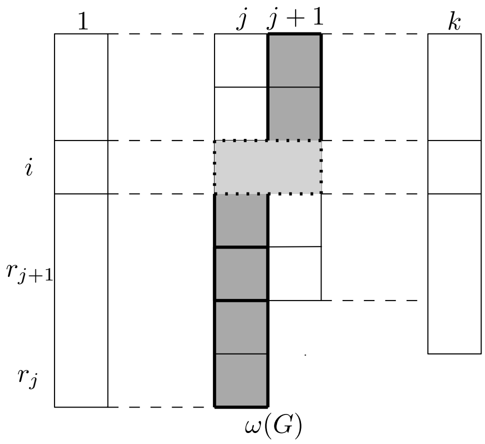


Figure 3.2: A maximum clique of G in a $\mathcal{U}$-bubble model. Dark grey color represents the bubbles that are fully contained in the clique. Light grey color highlights two bubbles where only parts of them are contained in the clique, concretely the one of the sets $B_{i,j}, B_{i,j+1}$, and $B^{*+}_{i,j} \cup B^{+*}_{i,j+1}$ with the maximum size.

Lemma 7. *Let G be a mixed unit interval graph, and let $\mathcal{B}$ be a $\mathcal{U}$-bubble model for G. Then the size of a maximum clique is*

$$\omega(G) = \max_{\substack{j \in \{1,\ldots,k-1\} \\ i \in \{1,\ldots,r_{j+1}\}}} \left(\sum_{i'=i+1}^{r_j} |B_{i',j}| + \sum_{i'=1}^{i-1} |B_{i',j+1}| + a_i \right),$$

$$a_i = \begin{cases} \max \left\{ |B_{i,j}|, |B_{i,j+1}|, |B^{*+}_{i,j}| + |B^{+*}_{i,j+1}| \right\} & i \leq r_j, \\ |B_{i,j+1}| & \textit{otherwise.} \end{cases}$$

Proof. Let K be a maximum clique of G. Notice, K does not contain two vertices from nonconsecutive columns, by the Observation 2. Furthermore, vertices u and v from two consecutive columns C_j and C_{j+1}, respectively, can be in K only if u is under v or they are in the same row in quadrants of types $\{*+\}$ and $\{+*\}$, respectively.

On the other hand, vertices from one column of $\mathcal{B}$ create a clique in $G(\mathcal{B})$. Moreover, if we split any two consecutive columns C_j and C_{j+1} in row i (for any index $i \in \{1, \ldots, \min\{r_j, r_{j+1}\}\}$), the second part of C_j with the first part of C_{j+1} form a clique. This is true even together with bubble quadrants $B^{*+}_{i,j} \cup B^{+*}_{i,j+1}$. $\square$

4. Clique-width of mixed unit interval graphs

The clique-width is one of the parameters which are used to measure the complexity of a graph. Many NP-hard problems, those which are expressible in Monadic Second Order Logic using second-order quantifiers on vertices (MSO_1), can be solved efficiently in graphs of bounded clique-with [8]. For instance, 3-coloring.

Definition of the clique-width is quite technical but it follows the idea that a graph of the clique-width at most k can be iteratively constructed such that in any time, there are at most k types of vertices, and vertices of the same type behave indistinguishable from the perspective of the newly added vertices.

Definition 7 (Courcelle 2000)**.** *The clique-width of a graph G, denoted by $cwd(G)$, is the smallest integer number of different labels that is needed to construct the graph G using the following operations:*

0. *creation of a vertex with label i,*
1. *disjoint union (denoted by $\oplus$),*
2. *relabeling: renaming all labels i to j (denoted by $\rho_{i\to j}$),*
3. *edge insertion: connecting all vertices with label i to all vertices with label j, $j \in \{1, \ldots, k\}$, $i \neq j$; already existing edges are not doubled (denoted by $\eta_{i,j}$).*

Such a construction of a graph can be represented by an algebraic term composed of the operations $\oplus$, $\rho_{i\to j}$, and $\eta_{i,j}$, called *cwd-expression*. We call *k-expression* a cwd-expression in which at most k different labels occur. Using this, we can say that clique-width of a graph G is the smallest integer k such that the graph G can be defined by a k-expression.

Example. The diamond graph G on the four vertices u, v, w, x (the complete graph K_4 without an edge vw) is defined by the following cwd-expression:

$$\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(1(u) \oplus 2(v) \oplus 2(w))) \oplus 2(x)).$$

Therefore, $cwd(G) \leq 2$.

Fellows, Rosamond, Rotics, and Szeider [12] proved in 2009 that the deciding whether the clique-width of a graph G is at most k is NP-complete. Therefore, researchers put effort into the computing an upper-bound of the clique-width.

Courcelle and Olariu [9] showed in 2000 that bounded treewidth implies bounded clique-width (but not vice versa). They showed that for any graph G with the treewidth k, the clique-width of G is at most $4 \cdot 2^{k-1} + 1$.

Golumbic and Rotics [17] proved that unit interval graphs have unbounded clique-width, consequently, (mixed unit) interval graphs have unbounding clique-width as well. Therefore, a computing upper-bounds are of particular interest. Fellows et al. [11] showed that the clique-width of a graph is bounded by its pathwidth + 2, therefore, the clique-width of interval graphs as well as of unit interval graphs is upper-bounded by the size of their maximum clique + 1 [22, 11]. Courcelle [9] observed that clique-width can be computed componentwise.

Lemma 8 (Courcelle 2000,[9]). *Any graph G satisfies that*

$$cwd(G) = max\{cwd(G') \mid G' \text{ is a connected component of } G\}.$$

We provide an upper-bound of the clique-width of a graph G depending on the number of columns in a $\mathcal{U}$-bubble model of G. We express it also in the size of a maximum independent set.

Lemma 9. *Let G be a mixed unit interval graph and $\mathcal{B}$ be a $\mathcal{U}$-bubble model of G. Then $cwd(G) \leq k+3$, where k is the number of columns of $\mathcal{B}$. Moreover, a $(k+3)$-expression defining the graph G can be constructed in $\mathcal{O}(n)$ time from $\mathcal{B}$.*

Proof. The proof is inspired by the proof for unit interval graphs [20].

We find a $(k+3)$-expression defining G and, therefore, prove that $cwd(G) \leq k+3$. We use $k+3$ labels where label i will be assigned to i-th column of $\mathcal{B}$ and the rest three labels, denoted by l_1, l_2, l_3, are used for maintaining the last two added vertices.

We define a linear order on vertices of G according to $\mathcal{B}$ as follows:

(i) We take the vertices from top to bottom, left to right. Formally, let $x \in B_{i,j}$, $y \in B_{l,m}$, we define $x \prec y$ if $i < l$ or ($i = l$ and $j < m$);

(ii) we define the following order on bubble quadrants:

$$x \prec y \prec z \prec w \text{ for } x \in B_{i,j}^{--}, y \in B_{i,j}^{+-}, z \in B_{i,j}^{-+}, w \in B_{i,j}^{++};$$

(iii) we define an arbitrary linear order on vertices in the same quadrant of the same bubble.

The idea of the proof is that every column has its own label and we need three more labels for maintaining the last added vertices. We will add vertices to G in the described order which ensures that a new vertex is complete to all vertices from the following column and anti-complete to all vertices from the previous column except those from the same row. Remind, according to the definition of $\mathcal{U}$-bubble model, there is an edge between vertices $x \in B_{i,j}$ and $y \in B_{i,j+1}$ if and only if $x \in B_{i,j}^{*,+}$ and $y \in B_{i,j+1}^{+*}$. Therefore, vertices from the last constructed bubble in the previous column must have two distinct labels according to the types of the vertices. However, once we add all vertices from the actual bubble, we do not need to distinguish between vertices from the previous column, anymore. Therefore, we rename their labels to the label of their column.

Formally. Let x be the first (smallest) vertex of G according to the defined linear order. We know that x is from the first column by Definition 6 (iv). If x is of type $(-,+)$ or $(+,+)$, we label it by l_1, otherwise by 1, so the expression for $G[\{x\}]$ is $1(x)$ if x is of type $(+,-)$ or $(-,-)$, and $l_1(x)$ otherwise.

Let y be the first non-processed vertex from G, i.e., a label is assigned to all preceding vertices. Let $l_2, l_3 \in \{k+1, k+2, k+3\}$ are currently unused labels or l_2 is used in the actual bubble $B_{i,j}$ and l_3 is unused, and l_1 may be used (in the previous column). Note that at most one label from $\{k+1, k+2, k+3\}$ is used in the previous column any time. We split the proof according to the type of y, the bubble quadrant where y belongs.

(a) $y \in B_{i,j}^{--}$. We use label l_2 for y. Then, we make y (the only one vertex with label l_2) complete to vertices with labels $j+1$ (if $j < k$) and j. Relabel l_2 to j.

(b) $y \in B_{i,j}^{+-}$. We use label l_2 for y. Then, we make y (the only one vertex with label l_2) complete to vertices with labels $j+1$ (if $j < k$), j, l_1. Relabel l_2 to j.

(c) $y \in B_{i,j}^{-+}$. We use label l_2 for y. Then, we make all vertices with label l_2 complete to vertices with labels $j+1$ (if $j < k$), j, l_2. (Do not relabel vertices with label l_2).

(d) $y \in B_{i,j}^{++}$. We use label l_3 for y. Then, we make y (the only one vertex with label l_3) complete to vertices with labels $j+1$ (if $j < k$), j, l_1, l_2. Relabel l_3 to l_2.

If all vertices from $B_{i,j}$ were used, we rename all vertices with the label l_1 to $j-1$ if $j > 1$. If $j = k$, we relabel l_2 to k.

For the correctness, observe that the previous column has always at most two labels, and in a), b), and d) the temporary label for y is unique (no other vertices are labeled by it in that time). The rest follows from the definition of adjacency in $\mathcal{U}$-bubble model. Since we constructed G using at most $k+3$ labels, $cwd(G) \leq k+3$.

The described algorithm processes each vertex once and each vertex has at most three labels in total. Moreover, the algorithm needs a constant work for each vertex—for instance, a cwd-expression for the option a) is:

$$\rho_{l_2 \to j}(\eta_{j,l_2}(\eta_{j+1,l_2}(l_2(y) \oplus G'))),$$

where G' is the already constructed graph before adding the vertex y. Therefore, the $(k+3)$-expression defining G is constructed in linear time given a $\mathcal{U}$-bubble model in an appropriate structure. $\square$

Theorem 10. *Let G be a mixed unit interval graph. Then $cwd(G) \leq 2\alpha(G)+3$. Moreover, a $(2\alpha(G)+3)$-expression defining the graph G can be constructed in $\mathcal{O}(n)$ time provided a $\mathcal{U}$-bubble model of G is given.*

Proof. We apply Lemma 9 and Lemma 6 together to obtain the statements. $\square$

Next, we provide a different bound for clique-width which is obtained by a small extension of the proof for unit interval graphs using the bubble model by Heggernes, Meister, and Papadopoulos [20]. We include the full proof for the completeness.

We need more notation. Let G be a mixed unit interval graph and let $\mathcal{B} = \langle B_{i,j} \rangle_{1 \leq j \leq k, 1 \leq i \leq r_j}$ be a $\mathcal{U}$-bubble model for G. We say that vertices from the same column j of $\mathcal{B}$ create a *group* if they have the same neighbours in the following column $j+1$ of $\mathcal{B}$. Let $v \in B_{i,j}$, the *group number of vertex* v in $\mathcal{B}$, denoted by $g_{\mathcal{B}}(v)$, is defined as the maximum number of groups in $N(v) \cap \left(\bigcup_{i'=i+1}^{r_{j-1}} B_{i',j-1} \cup \bigcup_{i'=1}^{i-1} B_{i',j} \cup A \right)$ over the sets $A = B_{i,j-1}^{*+} \cup B_{i,j}^{+*}$ or $A = B_{i,j}$. Then, the *group number of* G in $\mathcal{B}$ is defined as

$$\varphi_{\mathcal{B}}(G) := \max_{v \in V(G)} g_{\mathcal{B}}(v).$$

Lemma 11. *Let G is a mixed unit interval graph and $\mathcal{B}$ a $\mathcal{U}$-bubble model for G. The following inequality holds*

$$\varphi_{\mathcal{B}}(G) \leq \omega(G) - 1.$$

Proof. Let $v \in B_{i,j}$. Observe that $\bigcup_{i'=i+1}^{r_{j-1}} B_{i',j-1} \cup \bigcup_{i'=1}^{i-1} B_{i',j} \cup A \cup \{v\}$ is a clique (for both the possibilities of A), see Lemma 7. Moreover, v is not included in the counting the group number of v, and no vertex can be in more than one group. Therefore, $g_{\mathcal{B}}(v) \leq \omega(G) - 1$ for any vertex v which leads to the desired inequality. □

Theorem 12. *Let G be a mixed unit interval graph and $\mathcal{B}$ a $\mathcal{U}$-bubble model for G. Then $cwd(G) \leq \varphi_{\mathcal{B}}(G) + 2$. Moreover, a $(\varphi_{\mathcal{B}}(G) + 2)$-expression defining the graph G can be constructed in $\mathcal{O}(n + m)$ time provided a $\mathcal{U}$-bubble model of G is given.*

Proof. Our aim is to find a $(\varphi_{\mathcal{B}}(G) + 2)$-expression defining G. We add vertices in the order from left to right, top to bottom of $\mathcal{B}$ processing vertices of type $(+, *)$ at first, i. e., in the following linear order:

(i) $x \prec y$ for $x \in B_{i,j}$, $y \in B_{l,m}$, where $j < m$ or ($j = m$ and $i < l$);

(ii) $x \prec y \prec z \prec w$ for $x \in B_{i,j}^{++}, y \in B_{i,j}^{+-}, z \in B_{i,j}^{-+}, w \in B_{i,j}^{--}$;

(iii) an arbitrary linear order on the vertices in the same quadrant of the same bubble.

Now, we follow the original proof. Shortly, we add each vertex v in a proper way. We assume that a label is assigned for each previous vertex and all the vertices which belong to the same group have the same label. At first, we change to 1 the label of all the previous vertices which are nonadjacent to v. We know that at most $g_{\mathcal{B}}(v)$ distinct labels are used in the remaining groups, say labels $\{2, \ldots, g_{\mathcal{B}}(v) + 1\}$. This is true since all the groups are adjacent to v and because of the linear order.

Note that it is important to add first all the vertices of type $(+, *)$ from a bubble. Otherwise, $g_{\mathcal{B}}(v) + 1$ remaining groups could be there; in the situation that v is of type $(+, *)$, a potentially one distinct label is needed for $B_{i,j-1}^{*+}$, and another for $B_{i,j}^{*-}$. One the other hand, if all the vertices of type $(+, *)$ preceed vertices of type $(-, *)$ in one bubble, this situation does not happen—a potential label of $B_{i,j-1}^{*+}$ would be realeased. Therefore, it is enough to take into acount only the parts $A = B_{i,j-1}^{*+} \cup B_{i,j}^{+*}$, and $A = B_{i,j}$, and not the bigger one $A = B_{i,j-1}^{*+} \cup B_{i,j}$, in the definition of $g_{\mathcal{B}}(v)$.

We use a free label, say $g_{\mathcal{B}}(v) + 2$, for v and join all the vertices with this label with vertices with labels $2, \ldots, g_{\mathcal{B}}(v) + 1$. Next, change the label of v to a label of its group if v belongs to an already existing group. We continue with the next vertex. During the processing of each vertex we need no more than its group number + 2 distinct labels. Therefore, $cwd(G) \leq \varphi_{\mathcal{B}}(G) + 2$.

It remains to determine the running time for the construction of the expression defining G. Assume a $\mathcal{U}$-bubble model is given in a way that going over all vertices takes linear time in number of vertices. First, we count the time for the creation of groups. For each vertex v we compare its neighbors from the next column with the neighbors of the previous vertex in this column. Therefore, the splitting vertices into groups and determining the group number of G take $\mathcal{O}(m+n)$ time. In a constant time we determine a free label for each vertex. Then, we need to check the labels of groups in the neighborhood of each vertex v and create a $\mathcal{O}(g_{\mathcal{B}}(v))$ long cwd-expression, yealding $\mathcal{O}(m+n)$ time in total. Furthermore, each vertex is at most once relabeled to 1 since once it is relabeled to 1, its label remains 1 for the rest of the algorithm. Therefore, the relabeling of vertices which are nonadjacent to a newly added vertex take $\mathcal{O}(n)$ time in total. To sum up, the algorithm outputs the construction in $\mathcal{O}(n+m)$ time. $\square$

Corollary 1. *Let G be a mixed unit interval graph. Then*

$$cwd(G) \leq \min\{2\alpha(G)+3, \varphi_{\mathcal{B}}(G)+2\} \leq \omega(G)+1,$$

where $\mathcal{B}$ is a $\mathcal{U}$-bubble model for G. Moreover, the corresponding expression can be constructed in $\mathcal{O}(n+m)$ time providing $\mathcal{B}$ is given, otherwise in $\mathcal{O}(n^2)$ time.

Note that $\varphi_{\mathcal{B}}(G) \leq 2r_{max}$, where r_{max} is the length of the longest column in $\mathcal{B}$, i.e., $r_{max} = \max\{r_j \mid 1 \leq j \leq k\}$. Recall that Lemma 9 gives an upperbound on the clique-width depending on the number of columns of a $\mathcal{U}$-bubble model. It is useful to observe that the clique-width is upper-bounded by both numbers, the number of rows or columns. Especially, if one of them is bounded for a particular graph, then the clique-width is bounded, as well.

5. Maximum cardinality cut

This section is denoted to time complexity of the MaxCut problem on (mixed) unit interval graphs.

5.1 Notation

A *cut* of a graph $G(V, E)$ is a partition of $V(G)$ into two subsets $S, \overline{S}$, where $\overline{S} = V(G) \setminus S$. Since $\overline{S}$ is the complement of S we say for the bravity that a set S is a cut and similarly we use terms *cut vertex* and *non-cut* vertex for a vertex $v \in S$ and $v \in \overline{S}$, respectively. The *cut-set* of cut S is the set of edges of G with exactly one endpoint in S, we denote it $E(S, \overline{S})$. Then, the value $|E(S, \overline{S})|$ is the *cut size* of S. A *maximum (cardinality) cut* on G is a cut with the maximum size among all cuts on G. We denote the size of a maximum cut of G by $mcs(G)$. Finally, the MaxCut problem is the problem of finding a maximum cut.

5.2 Time complexity is still unknown on unit interval graphs

As it was mentioned in the introduction, there is a paper *A polynomial-time algorithm for the maximum cardinality cut problem in proper interval graphs* by Boyaci, Ekim, and Shalom from 2017 [4], claiming that the MaxCut problem is polynomial-time solvable in unit interval graphs and giving a dynamic programming algorithm based on the bubble model representation. We realised that the algorithm is incorrect and this section is denoted to it.

We start with a counterexample to the original algorithm in [4].

Example. Let $\mathcal{B} = \langle B_{i,j} \rangle_{1 \le j \le 2, 1 \le i \le 2}$, where $B_{1,1} = \{v_1\}$, $B_{2,1} = \{v_2\}$, $B_{1,2} = \{v_3, v_4, v_5\}$, $B_{2,2} = \{v_6\}$, be a bubble model for a graph G, see also Figure 5.1. In other words, this bubble model corresponds to a unit interval graph on vertices v_2, v_3, v_4, v_5, v_6 where there is an edge $v_1 v_2$, and vertices v_2, v_3, v_4, v_5, v_6 create a complete graph without an edge $v_2 v_6$.

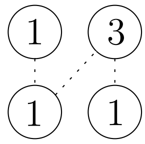


Figure 5.1: A counterexample to the original algorithm, a bubble model $\mathcal{B}$ where the numbers denote the number of vertices in each bubble, and dashed lines indicate the edges between bubbles.

Then, according to the paper [4], the size of a maximum cut in G is eight. However, the size of a maximum cut in G is only seven. To be more concrete, the algorithm fills the following values of dynamic table: $F_{0,1}(0,0) = 4$, $F_{2,1}(1,1) = 8$ for $s_{2,1} = 1, s_{2,2} = 1$, and finally, $F_{0,0}(0,0) = 8$ which is the output of the algorithm.

Suppose, for contradiction, that the size of a maximum cut is eight. There are ten edges in total in G, thus, at least one vertex of the triangle v_3, v_4, v_5 must be a cut-vertex and one not. Then, those two vertices have three common neighbors. Therefore, the size of a maximum cut is at most seven which is possible; for example v_1, v_4, v_5 are cut-vertices.

The brief idea of the algorithm is to process the columns from the biggest to the lowest column from the top bubble to bottom one. Once we know the number of cut-vertices in the actual processed bubble B (in the column j) and the number of cut-vertices which are above B in the columns j and $j+1$, we can count the exact number of edges. For each bubble and each such number of cut-vertices in the columns j and $j+1$ (above the bubble), we remember only the best values of MaxCut[1].

The algorithm and its full idea from [4] is not correct since we lose the consistency there—to obtain a maximum cut, we do not remember anything about the distribution of cut vertices within bubbles, that was used in the previously processed column. Therefore, there is no guarantee that the final outputted cut of the computed size exists. To be more specific, one of two problems is in the moving from the column j to the column $j-1$ since we forget there too much. The second problem is that for each bubble $B_{i,j}$ and for each possible numbers x, x' we count the size $F_{i,j}(x, x')$ of a specific cut and we choose some values $s_{i,j}$, $s_{i,j+1}$ (possibly different; they represents the number of cut-vertices in the bubbles $B_{i,j}, B_{i,j+1}$) which maximize the values of $F_{i,j}(x, x')$. In few steps later, when we are processing the bubble $B_{i,j-1}$, again, for each possible values y and y' we choose some values $s'_{i,j-1}$ and $s'_{i,j}$ such that they maximize the size of $F_{i,j-1}(y, y')$. However, we need to be consistent with the selection in the previous column, i.e., to guarantee that $s_{i',j} = s_{i,j}$ for any particular values y, $y' = x$, and x'.

A straightforward correction of the algorithm would lead to remembering too much for a polynomial-time algorithm. However, we can be inspired by it to obtain a subexponential-time algorithm. I tried to correct the algorithm or extend the idea leading to the polynomiality. In the end, despite lots of effort, I was not successful and it seemed to me that the algorithm is hardly repairable and the problem is more difficult than I expected.

We note here, that there is another paper by the same authors [5] where a very similar polynomial algorithm is used for MaxCut of co-bipartite chain graphs with twins. Those graphs can be viewed as graphs given by bubble models with two columns; but having two columns is a crucial property for the algorithm.

To conclude, the time complexity of the MaxCut problem in unit interval graphs is still open.

5.3 Subexponential algorithm in mixed unit interval graphs

Here, we present a subexponential-time algorithm for the MaxCut problem in mixed unit interval graphs. Our aim is to have an algorithm running in $2^{\tilde{\mathcal{O}}(\sqrt{n})}$ time. A similar idea for unit interval graphs was suggested by Karczmarz, Nadara,

[1] We refer the reader to the paper [4] for the notation and the description of the algorithm.

Zych-Pawlewicz, and Rzazewski during discussions at Parameterized Algorithms Retreat of University of Warsaw 2019 [7].

Let G be a mixed unit interval graph. We start with a $\mathcal{U}$-bubble model $\mathcal{B} = \langle B_{i,j} \rangle_{1 \le j \le k, 1 \le i \le r_j}$ for G and we distinguish columns of $\mathcal{B}$ according to their number of vertices.

At first, let us write some notation. We denote by b_{ij} the number of vertices in bubble $B_{i,j}$ and by c_j the number of vertices in column j, i.e., $b_{ij} = |B_{i,j}|$ and $c_j = \sum_{i=1}^{r_j} b_{i,j}$. We call a column j with $c_j > \sqrt{n}$ a *heavy column*, otherwise a *light column*. We call consecutive heavy columns and the two their bordering light columns a *heavy part* of $\mathcal{B}$ (if $\mathcal{B}$ starts or ends with a heavy column, for brevity, we add an empty column at the beginning or the end of $\mathcal{B}$), and we call their light columns *borders*. Heavy part might contain no heavy columns in a case that two light columns are consecutive.

Note that we can guess all possible cuts in one light column without exceeding the aimed time, and that the most of those light column guesses are independent of each other—once we know the cut in the previous column, it does not matter what is the cut in columns before. Furthermore, there are at most $\sqrt{n}$ consecutive heavy columns which allows us to process them together. More formally, we show that we can determine a maximum cut independently for each heavy part, given a fixed cut on its borders.

Lemma 13. *Let G be a mixed unit interval graph and $\mathcal{B}$ be a $\mathcal{U}$-bubble model for G partitioned into heavy parts $\hat{\mathcal{B}}_1, \cdots, \hat{\mathcal{B}}_p$ in this order. If $S = S_0 \cup \cdots \cup S_p$ is a (fixed) cut of light columns $C_0, \ldots, C_p$ in $G(\mathcal{B})$ such that S_j is a cut of C_j, $j \in \{0, \ldots, p\}$, then the size of a maximum cut of G is*

$$mcs(G) = \sum_{j=1}^{p} mcs(G(\hat{\mathcal{B}}_j)) - (\sum_{j=1}^{p-1} |S_j| \cdot |C_j \setminus S_j|).$$

Proof. It is readily seen that once we have a fixed cut in an entire column C of a bubble model, a maximum cut of columns which are to the left of C (including C) is independent on a maximum cut of those which are to the right of C (including C). Therefore, we can sum the sizes of maximum cuts in heavy parts which are separated by fixed cuts. However, the cut size of middle light columns are counted twice since they are contained in two heavy parts. Therefore, we subtract them. □

Note that, if $\hat{\mathcal{B}}$ is a heavy part with no heavy columns, we can straightforwardly count the number of cut edges of $G(\hat{\mathcal{B}})$, $mcs(G(\hat{\mathcal{B}}))$, assuming a fixed cut on borders is given. Therefore, we are focusing on a situation where at least one heavy column is present in a heavy part. We use dynamic programming for the determining a maximum cut on each such heavy part.

First, we present a brief idea of the dynamic programming approach. We store all possible $(l+1)$-tuples $(s_1, s_2, \ldots, s_l, a)$, where l is the number of heavy columns and $0 \le s_j \le \sum_{i'=1}^{r_j} b_{i',j}$ characterizes the number of cut vertices in the j-th heavy column, the number a is auxiliary and characterizes the number of cut vertices of types $(*, +)$ in the last processed bubble. Then, we define recursive function f which is connected with the size of a specific cut.

We start formally with a new notation. Let $\hat{\mathcal{B}}$ be a heavy part with $l \ge 1$ heavy columns (numbered by $1, \ldots, l$) and borders C_0 and C_{l+1}. Let $B_1, \ldots, B_m$

be bubbles in $\hat{\mathcal{B}} \setminus (C_0 \cup C_{l+1})$ numbered in the top-bottom, left-right order. Let S_0 and S_{l+1} be (fixed) cuts in C_0 and C_{l+1}.

We define functions $n^{\downarrow}, n^{\leftarrow}, n^{\uparrow}$, $n^{\rightarrow}$ which output the number of cut vertices in borders in a specific positions depending on the given row and column, they outputs 0 if the column of the given column number is not next to the borders.

- The number of (fixed) cut vertices in C_0 under the row r or 0 if the previous column is not C_0,

$$n^{\downarrow}(r,c) := \begin{cases} |S_0 \cap \bigcup_{k=r+1}^{r_0} B_{k,0}| & c = 1 \\ 0 & c \neq 1 \end{cases}$$

- the number of (fixed) cut vertices of type $(*,+)$ in the left border C_0 in the row r,

$$n^{\leftarrow}(r,c) := \begin{cases} |S_0 \cap B_{r,0}^{*,+}| & c = 1 \\ 0 & c \neq 1 \end{cases}$$

- the number of (fixed) cut vertices in the right border C_{l+1} above the row r,

$$n^{\uparrow}(r,c) := \begin{cases} |S_{l+1} \cap \bigcup_{k=1}^{r-1} B_{k,l+1}| & c = l \\ 0 & c \neq l \end{cases}$$

- the number of (fixed) cut vertices of type $(+,*)$ in the right border C_{l+1} in the row r,

$$n^{\rightarrow}(r,c) := \begin{cases} |S_{l+1} \cap \ B_{r,l+1}^{+,*}| & c = l \\ 0 & c \neq l \end{cases}$$

We denote the number of vertices in B_i by $b_i := |B_i|$, analogously $b_i^{xy} := |B_i^{xy}|$, $x, y \in \{+,-\}$.

We further denote the set of counts corresponding to all possible choices of cut vertices in the bubble B_i by β_i, i.e.,

$$\beta_i := \Big\{(n_1, n_2, n_3, n_4) \mid n_1 \in \{0, \ldots, b_i^{++}\}, n_2 \in \{0, \ldots, b_i^{+-}\}, n_3 \in \{0, \ldots, b_i^{-+}\}, \\ n_4 \in \{0, \ldots, b_i^{--}\}, n_1 + n_2 + n_3 + n_4 \leq s_{\mathrm{col}(B_i)}\Big\}.$$

In addition, we denote the set of $(l+1)$-tuples characterizing all possible counts of cut-vertices in the l heavy columns and an auxiliary number characterizing the count of possible edges from the last processed bubble, by

$$T = \Big\{(s_1, \ldots, s_l, a) \mid a \in \mathbb{N}, 0 \leq a \leq \max_{i \in \{1,\ldots,m\}} (b_i^{-+} + b_i^{++}), \\ a \in \mathbb{N}, a \in \{0, \ldots, \max_{i \in \{1,\ldots,m\}} (b_i^{-+} + b_i^{++})\}, \\ \forall j \in \{0, 1, \ldots, l\} : s_j \in \mathbb{N}, 0 \leq s_j \leq \sum_{i=1}^{r_j} b_{i,j}\Big\}.$$

Let $e(s_1, s_2)$ denote the number of cut-edges between two sets S_1, S_2 which are complete to each other and S_k, $k \in \{1, 2\}$, contains s_k cut vertices and $\overline{s_k}$ non-cut vertices, i.e., $e(s_1, s_2) = s_1 \cdot \overline{s_2} + \overline{s_1} \cdot s_2$. We remark that it is important to know the numbers of non-cut vertices ($\overline{s_1}$ and $\overline{s_2}$), however, we will not write them explicitly for the easier formulas. It will be seen that they can be, for instance, stored in parallel with the numbers of cut vertices (or counted in each step again).

Finally, we define a recursive function f by the following recurrence relation:

$$\begin{aligned}
&\forall (s_1, \dots, s_l, a) \in T:\\
&\quad \text{if } s_1 \le b_1, s_2 = \dots = s_l = 0:\\
&\qquad f_1((s_1, \dots, s_l, a)) = \max_{\substack{(b^{++}, b^{-+}, b^{+-}, b^{--}) \in \beta_1:\\ b^{++}+b^{-+}+b^{+-}+b^{--}=s_1,\\ b^{++}+b^{-+}=a}} \Bigg(e\left(s_1, n^{\downarrow}(r,1)\right) + s_1 \cdot (b_1 - s_1)\\
&\qquad\qquad + e\left(n^{\leftarrow}(r,1), (b^{++}+b^{+-})\right)\\
&\qquad\qquad + e\left(n^{\rightarrow}(r,1), b^{++}+b^{-+}\right)\Bigg),\\
&\quad \text{otherwise:}\\
&\qquad f_1((s_1, \dots, s_l, a)) = -\infty.
\end{aligned}$$

$$\begin{aligned}
&\forall i \in \{1, \dots, m-1\}, \forall (s_1, \dots, s_l, a) \in T:\\
&\quad f_i((s_1, \dots, s_l, a)) = \max_{\substack{(b^{++},\, b^{+-},\, b^{-+},\, b^{--}) \in \beta_i,\\ z \in \mathbb{N}:\\ b^{++}+b^{-+}=a,\\ (s_1, \dots, s_c - b, \dots, s_l, z) \in T,\\ z \le |B^{*+}_{i-1}|,\ z \le s_c - b}} \Bigg(f_{i-1}((s_1, \dots, s_c - b, \dots, s_l, z))\\
&\qquad\qquad + b \cdot (b_i - b)\\
&\qquad\qquad + e(b, s_c + 1) + e(b, s_c - b)\\
&\qquad\qquad + e\left(n^{\downarrow}(r,c), b\right)\\
&\qquad\qquad + e\left(n^{\leftarrow}(r,c), (b^{++}+b^{+-})\right)\\
&\qquad\qquad + e\left((b^{++}+b^{-+}), n^{\rightarrow}(r,c)\right) + A\Bigg)\\
&\quad \text{where } A = \begin{cases} e(z\,, b^{++}+b^{+-}) & i > 1,\ c = \mathrm{col}(B_{i-1}) + 1\\ 0 & \text{otherwise} \end{cases}\\
&\qquad\qquad b = b^{++}+b^{+-}+b^{-+}+b^{--},\ c = \mathrm{col}(B_i), r = \mathrm{row}(B_i), \text{ and } s_{l+1} = n^{\uparrow}(r,c).
\end{aligned}$$

Theorem 14. *Let $\hat{\mathcal{B}}$ be a heavy part with $l \ge 1$ heavy columns (numbered by $1, \dots, l$) and borders C_0 and C_{l+1}. Let $B_1, \dots, B_m$ be bubbles in $\hat{\mathcal{B}} \setminus (C_0 \cup C_{l+1})$ numbered in the top-bottom, left-right order. Let S_0 and S_{l+1} be (fixed) cuts in C_0 and C_{l+1}. Then, the size of a maximum cut in $G(\hat{\mathcal{B}})$ is*

$$mcs(G(\hat{\mathcal{B}})) = \max_{s \in T} f_m(s).$$

Proof. At first, we prove the following auxiliary lemma. We denote by G_i the induced subgraph of $G(\hat{\mathcal{B}})$ with $V(G_i) = B_1 \cup \dots \cup B_i \cup C_0 \cup C_{l+1}$ where C_0 and C_{l+1} are borders of $\hat{\mathcal{B}}$ (as it is shown in Figure 5.2).

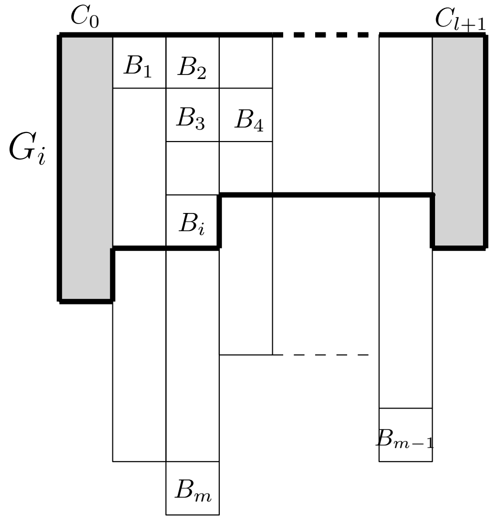


Figure 5.2: A heavy part with light columns C_0 and C_{l+1} and the highlighted subgraph G_i.

Lemma 15. *For each $s = (s_1, \ldots, s_l, a) \in T$ and for every $i \in \{1, \ldots, m\}$, the value $f_i(s)$ is equal to the maximum size of such a cut S in G_i which satisfies that for every $j \in \{1, \ldots, l\}$ the number of cut vertices in the column j in G_i is equal to s_j, and z is equal to the number of cut vertices from $B_i^{++} \cup B_i^{-+}$, or f_i is equal to $-\infty$ if there is no such cut.*

Proof. We prove Lemma 15 by induction on number of steps (bubbles). Since B_1 is in the first heavy column, Lemma 15 is true for $i = 1$ by Definition 6 (iv).

In the inductive step, suppose that for every $s = (v_1, v_2, \cdots, v_l, z) \in T$, $f_{i-1}(s)$ is equal to the size of a maximum cut S_{i-1} in G_{i-1} such that the number of cut vertices in each column j, for every $j \in \{1, 2, \ldots, l\}$, in G_{i-1} is equal to v_j, and the number of cut vertices from B_{i-1}^{*+} is equal to z. Or $f_{i-1}(s)$ is equal to $-\infty$ if the such cut does not exist.

Observe that the edges of G_i can be partitioned into following disjoint sets:

- E_1 edges of the graph G_{i-1},
- E_2 edges with both endpoints in B_i,
- E_3 edges with one endpoint in B_i and the second endpoint in the same column above B_i,
- E_4 edges with one endpoint in B_i and the second endpoint in the next column above B_i,
- E_5 edges with one endpoint in B_i and the second endpoint in the previous column in the same row as B_i,
- E_6 edges with one endpoint in B_i and the second endpoint in the column C_0 bellow B_i,
- E_7 edges with one endpoint in B_i and the second endpoint in the column C_{l+1} in the same row as B_i.

Note that E_6 is non-empty only if B_i is in the column 1, similarly E_7 is non-empty only if B_i is in the column l.

Let $s = (s_1, \ldots, s_l, a) \in T$ is fixed. At first assume, S is a maximum cut in G_i such that it contains s_j vertices from the column j for each $j \in \{1, 2, \cdots, l\}$ and a vertices from B_i^{*+}, we say S *satisfies the conditions* s. We discuss the case, where no such cut exists, later. We denote by s^{xy} the number of vertices in $B_i \cap S^{xy}$, $x, y \in \{+, -\}$, and by s' the sum of these values, i.e., $s' = s^{++} + s^{+-} + s^{-+} + s^{--}$. We denote $\text{col}(B_i)$ by j, and $\text{row}(B_i)$ by r. Then,

$$E(S, \overline{S}) = (E(S, \overline{S}) \cap E(G_{i-1})) \cup \{uv \in E_k \mid u \in S, v \notin S, k \in \{2, \ldots, 6\}\}$$

Which leads to

$$\begin{aligned}
|E(S, \overline{S})| = &|E(S, \overline{S}) \cap E(G_{i-1})| \\
&+ s' \cdot (b_i - s') \\
&+ e(s', (s_j - s')) \\
&+ e(s', s_{j+1}) \\
&+ A \\
&+ e(s', n^{\downarrow}(r, j)) + e(s^{++} + s^{+-}, n^{\leftarrow}(r, j)) \\
&+ e(s^{++} + s^{-+}, n^{\rightarrow}(r, j))
\end{aligned}$$

$$\text{where } A = \begin{cases} e(|S \cap B_{i-1}^{*+}|, s^{++} + s^{+-}) & j = \text{col}(B_{i-1} + 1), \\ 0 & \textit{otherwise}. \end{cases}$$

By the induction hypothesis,

$$|E(S, \overline{S}) \cap E(G_{i-1})| \le f_{i-1}(s_1, \cdots, s_j - s', \cdots, s_l, |S \cap B_{i-1}^{*+}|). \tag{5.1}$$

It gives us together with the right part of the equation, the definition of f_i for $b^{xy} = s^{xy}$, $b = s'$ and $z = |S \cap B_{i-1}^{*+}|$. Therefore

$$|E(S, \overline{S})| \le f_i(s).$$

Furthermore, we show that $f_i(s)$ is the size of a cut satisfying the conditions s. Since the value of the function $f_{i-1}((s_1, \cdots, s_j - b, \cdots, s_l, z))$ is for any number $b \in \{0, \ldots, \min(s_j, b_i)\}$ a size of a cut in G_{i-1} which satisfies the conditions $(s_1, \cdots, s_j - b, \cdots, s_l, z)$, or $-\infty$ (if no such cut exists), we can extend that cut into G_i by adding b^{xy} vertices from B_i^{xy} where $b^{++} + b^{-+} = a$ and $b^{++} + b^{+-} + b^{-+} + b^{--} = b_i$. Consequently, $f_i(s)$ is a size of a cut on G_i satisfying that it contains s_i vertices from the column i and a vertices from $|B_{i-1}^{+*}|$. At least one such such exists, by (5.1). Therefore, $|E(S, \overline{S})| \ge f_i(s)$. It leads to the equation $|E(S, \overline{S})| = f_i(s)$, otherwise, S is not a maximum cut.

In a similar way, we can extend every cut on G_{i-1} to G_i. Therefore, if there exist no cut on G_i which satisfies the conditions s, there exists no cut in G_{i-1} which can be extended to the cut on G_i satisfying the conditions s. Consequently, $f_i(s) = -\infty$ by the definition of f since $f_{i-1}(v) = -\infty$ for all $(l+1)$-tuples v which appear in the definition. □

Finally, Theorem 14 is a corollary of Lemma 15.

□

Lemma 16. *Let $\hat{\mathcal{B}}$ be a heavy part with $l \geq 1$ columns, m bubbles, and a fixed cut on the borders. The size of a maximum cut of $\hat{\mathcal{B}}$ can be determined in time:*

$$(c_1 + 1) \cdots (c_l + 1) \cdot (a + 1) \cdot \sum_{i=1}^{m} \left(b_i^{++} \cdot b_i^{+-} \cdot b_i^{-+} \cdot b_i^{--} \right)$$

where c_j is the number of vertices in the column j, i.e., $c_j = \sum_{i'=1}^{r_j} B_{i',j}$, and $a = \max_i |B_i^{*+}|$.

Proof. We analyze the time complexity of the algorithm from Theorem 14. Let T denote all the possible $(l+1)$-tuples. Then $|T| = (c_1 + 1) \cdots (c_l + 1) \cdot (a + 1)$. The time for processing a bubble B_i is $|T| \cdot b_i^{++} \cdot b_i^{+-} \cdot b_i^{-+} \cdot b_i^{--}$. The time complexity of processing $\hat{\mathcal{B}}$ is then

$$|T| \cdot \sum_{i=1}^{m} b_i^{++} \cdot b_i^{+-} \cdot b_i^{-+} \cdot b_i^{--}$$

□

Theorem 17. *Let G be a mixed unit interval graph. The maximum cardinality cut can be found in time $2^{\tilde{0}(\sqrt{n})}$.*

Proof. By Lemma 13, heavy parts can be processed independently on each other, given a cut on their borders. Moreover, it is sufficient for a light column C to remember only the biggest cuts on the left of C (containing C) for each possible cut in C. Therefore, there is no need to quess cuts in all light columns at once. It is sufficient to quess a cut only in two consecutive light columns at once.

Observe that there are at most $2^{\sqrt{n}}$ guesses of cut vertices for a light column and there are at most n light columns. Therefore, the time complexity of determining the size of a maximum cut in G is at most $n \cdot (2^{\sqrt{n}})^2 \cdot P$, where P is the maximum time for processing a heavy part. Now, we want to prove that a time complexity of processing a heavy part $\hat{\mathcal{B}} = \langle B_{i,j} \rangle_{1 \leq j \leq l, 1 \leq i \leq r_j}$ with a given guess of light columns is $2^{\tilde{\mathcal{O}}(\sqrt{n})}$.

By Lemma 16, the time complexity of processing a heavy part with a given guess of light columns is

$$\begin{aligned} P &= (c_1 + 1) \cdots (c_l + 1) \cdot (a + 1) \cdot \sum_{i=1}^{m} \left(b_i^{++} \cdot b_i^{+-} \cdot b_i^{-+} \cdot b_i^{--} \right) \\ &\leq (n+1)^{1+\sqrt{n}} \cdot \sum_{i=1}^{m} b_i^4 \leq (n+1)^{1+\sqrt{n}} \cdot n^4 \in 2^{\tilde{\mathcal{O}}(\sqrt{n})}. \end{aligned}$$

To sum up, we can determine the size of a maximum cut in the time: $n \cdot (2^{\sqrt{n}})^2 \cdot P \in 2^{\tilde{\mathcal{O}}(\sqrt{n})}$. For brevity we analyzed only the size of a maximum cut. However, the maximum cut itself can be determine retroactively in the time $2^{\tilde{\mathcal{O}}(\sqrt{n})}$, as well. □

Lemma 16 has a nice corollary for graphs with a $\mathcal{U}$-bubble models with constant number of columns. According to Lemma 16, we are able to solve the MaxCut problem in those graphs in polynomial time. There were a polynomial time algorithm by Boyaci, Ekim, Shalom[5] solving the MaxCut problem in co-bipartite chain graphs with possible twins which is exactly the class of graphs defined by a classic bubble model with only two columns. We extend this result for $\mathcal{U}$-bubble model and constant number of columns.

Corollary 2. *The size of a maximum cut in the graph class defined by $\mathcal{U}$-bubble models with k columns can be determined in the time $\mathcal{O}(n^{k+5})$. Moreover, for $k = 2$ in time $\mathcal{O}(n^5)$.*

Proof. Let G be a graph on n vertices which is defined by a $\mathcal{U}$-bubble model $\mathcal{B}$ with k columns and m bubbles. The bubble model $\mathcal{B}$ can be seen as a heavy part with no cut-vertices in its borders. By Lemma 16, the size of a maximum cut in $\mathcal{B}$ can be determined in time $T = (c_1+1)\cdots(c_k+1)\cdot(a+1)\cdot\sum_{i=1}^{m}\left(b_i^{++}\cdot b_i^{+-}\cdot b_i^{-+}\cdot b_i^{--}\right)$ where $b_i^{xy}, xy \in \{+,-\}$ is the number of vertices in the bubble quadrant B_i^{xy}, and c_j is the number of vertices in the column j, i.e., $c_j = \sum_{i'=1}^{r_j} B_{i',j}$, and $a = \max_i |B_i^{*+}|$.

By Arithmetic Mean-Geometric Mean Inequality (AM-GM) we obtain

$$\begin{aligned}
T &\leq (a+1)\cdot\left(\frac{1}{k}\cdot\sum_{j=1}^{k}(c_j+1)\right)^k\cdot\sum_{i=1}^{m}\left(\frac{b_i^{++}+b_i^{+-}+b_i^{-+}+b_i^{--}}{4}\right)^4 \\
&= (a+1)\cdot\left(\frac{n+k}{k}\right)^k\cdot\sum_{i=1}^{m}\left(\frac{b_i}{4}\right)^4 \\
&\leq (a+1)\cdot\left(\frac{n+k}{k}\right)^k\cdot\left(\frac{n}{4}\right)^4 \in \mathcal{O}(n^{k+5}).
\end{aligned}$$

It remains to prove the special case where $k = 2$. Notice, it is sufficient to distinguish only between vertices in quadrants of types $(*,+)$ and $(*,-)$ in the first column, and simillarly $(+,*)$ and $(-,*)$ in the second column. Therefore, we obtain $\left(\frac{b_i}{2}\right)^2$ instead of $\left(\frac{b_i}{4}\right)^4$ which leads to the time complexity $\mathcal{O}(n^{k+1+2}) = \mathcal{O}(n^5)$. $\square$

Note that Theorem 14 states the explicit size of a maximum cut.

Conclusion

The main contribution of this work is a new representation of mixed unit interval graphs—the $\mathcal{U}$-bubble model. This structure is particularly useful in the design of algorithms and their analysis. Using the $\mathcal{U}$-bubble model, we presented new upper-bounds for the clique-width of mixed unit interval graphs and designed a subexponential-time algorithm for the MaxCut problem on mixed unit interval graphs. We further realised that the state-of-the-art polynomial-time algorithm for the MaxCut problem on unit interval graphs is incorrect.

There are several directions that can be further studied. Which results for unit interval graphs can be extended to mixed unit interval graphs? On the other hand, a long standing open problem is the time complexity of the MaxCut problem in unit interval graphs, in particular, distinguish if it is NP-hard or polynomial time solvable.

Another interesting direction could be the study of labeling problems – either $L_{2,1}$-labeling or Packing Coloring. Both problems were motivated by assigning frequencies to transmitters. The $L_{2,1}$-labeling problem was firstly introduced by Griggs and Yeh in 1992 [18]. The packing coloring problem is newer, it was introduced by Goddard et al. in 2008 [15]. Although, these are well-known problems, quite surprisingly, their time complexity is open for interval graphs.

The $L_{2,1}$-labeling problem assigns labels $\{0, \dots, k\}$ to vertices such that the labels of neighbouring vertices differ by at least two and the labels of vertices in distance two are different. The time complexity of this problem is still wide open even for unit interval graphs, despite a partial progress about concrete values for the largest used label. Sakai proved that the value of the largest label lies between $2\chi - 2$ and 2χ where χ is the chromatic number [30].

The packing coloring problem asks for an existence of such a mapping $c : V \to \{1, \dots, k\}$ that for all $u \neq v$ with $c(u) = c(v) = i$ the distance between u and v is at least i. This problem is wide open on interval graphs. Recently, there was a small progress on unit interval graphs leading to an FPT algorithm (time $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f and a parameter k). It is shown in [24] that the packing coloring problem is in FPT parameterized by the size of a maximum clique. We note that the algorithm can be straightforwardly extended to mixed unit interval graphs. However, a polynomial time algorithm or alternatively NP-hardness for (unit) interval graphs is of a much bigger interest.

Bibliography

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

[2] Hans L. Bodlaender, Ton Kloks, and Rolf Niedermeier. SIMPLE MAX-CUT for unit interval graphs and graphs with few p_4s. *Electronic Notes in Discrete Mathematics*, 3:19–26, 1999. doi:10.1016/S1571-0653(05)80014-9.

[3] Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using pq-tree algorithms. *Journal of Computer and System Sciences*, 13(3):335–379, 1976. doi:10.1016/S0022-0000(76)80045-1.

[4] Arman Boyaci, Tinaz Ekim, and Mordechai Shalom. A polynomial-time algorithm for the maximum cardinality cut problem in proper interval graphs. *Information Processing Letters*, 121:29–33, 2017. doi:10.1016/j.ipl.2017.01.007.

[5] Arman Boyaci, Tinaz Ekim, and Mordechai Shalom. The maximum cardinality cut problem in co-bipartite chain graphs. *Journal of Combinatorial Optimization*, 35(1):250–265, 2018. doi:10.1007/s10878-015-9963-x.

[6] Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph Classes: A Survey*, volume 3. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[7] Personal communication with Adam Karczmarz, Wojciech Nadara, Anna Zych-Pawlewicz, and Pawel Rzazewski. *Parameterized Algorithms Retreat of University of Warsaw 2019.*

[8] Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

[9] Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, 101(1-3):77–114, 2000. doi:10.1016/S0166-218X(99)00184-5.

[10] Mitre C. Dourado, Van B. Le, Fábio Protti, Dieter Rautenbach, and Jayme L. Szwarcfiter. Mixed unit interval graphs. *Discrete Mathematics*, 312(22):3357–3363, 2012. doi:10.1016/j.disc.2012.07.037.

[11] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-width minimization is NP-hard. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing, Seattle, WA, USA, May 21-23, 2006*, pages 354–362, 2006. doi:10.1145/1132516.1132568.

[12] Michael R. Fellows, Frances A. Rosamond, Udi Rotics, and Stefan Szeider. Clique-width is NP-complete. *SIAM Journal on Discrete Mathematics*, 23(2):909–939, 2009. doi:10.1137/070687256.

[13] Peter Frankl and Hiroshi Maehara. Open-interval graphs versus closed-interval graphs. *Discrete Mathematics*, 63(1):97–100, 1987. doi:10.1016/0012-365X(87)90156-7.

[14] Frédéric Gardi. The roberts characterization of proper and unit interval graphs. *Discrete Mathematics*, 307(22):2906–2908, 2007. doi:10.1016/j.disc.2006.04.043.

[15] Wayne Goddard, Sandra M. Hedetniemi, Stephen T. Hedetniemi, John M. Harris, and Douglas F. Rall. Braodcast chromatic numbers of graphs. *Ars Comb.*, 86, 2008.

[16] Martin Charles Golumbic. Algorithmic graph theory and perfect graphs. *Computer Science and Applied Mathematics*, XX:284, 1980.

[17] Martin Charles Golumbic and Udi Rotics. On the clique-width of some perfect graph classes. *International Journal of Foundations of Computer Science*, 11(3):423–443, 2000. doi:10.1142/S0129054100000260.

[18] Jerrold R. Griggs and Roger K. Yeh. Labelling graphs with a condition at distance 2. *SIAM Journal on Discrete Mathematics*, 5(4):586–595, 1992. doi:10.1137/0405048.

[19] Gyorgy Hajós. Über eine art von graphen. *Internationale Mathematische Nachrichten*, 11:65, 1957.

[20] Pinar Heggernes, Daniel Meister, and Charis Papadopoulos. A new representation of proper interval graphs with an application to clique-width. *Electronic Notes in Discrete Mathematics*, 32:27–34, 2009. doi:10.1016/j.endm.2009.02.005.

[21] Felix Joos. A characterization of mixed unit interval graphs. *Journal of Graph Theory*, 79(4):267–281, 2015. doi:10.1002/jgt.21831.

[22] Haim Kaplan and Ron Shamir. Pathwidth, bandwidth, and completion problems to proper interval graphs with small cliques. *SIAM Journal on Computing*, 25(3):540–561, 1996. doi:10.1137/S0097539793258143.

[23] J. Mark Keil. Finding hamiltonian circuits in interval graphs. *Information Processing Letters*, 20(4):201–206, 1985. doi:10.1016/0020-0190(85)90050-X.

[24] Minki Kim, Bernard Lidický, Tomáš Masařík, and Florian Pfender. Notes on complexity of packing coloring. *Information Processing Letters*, 137:6–10, 2018. doi:10.1016/j.ipl.2018.04.012.

[25] Van Bang Le and Dieter Rautenbach. Integral mixed unit interval graphs. *Discrete Applied Mathematics*, 161(7-8):1028–1036, 2013. doi:10.1016/j.dam.2012.09.013.

[26] Dieter Rautenbach and Jayme Luiz Szwarcfiter. Unit interval graphs of open and closed intervals. *Journal of Graph Theory*, 72(4):418–429, 2013. doi:10.1002/jgt.21650.

[27] F. S. Roberts. Indifference graphs. *Proof Techniques in Graph Theory*, pages 139–146, 1969. URL: https://ci.nii.ac.jp/naid/10025491782/en/.

[28] F. S. Roberts. Some applications of graph theory. Draft, 2000.

[29] F.S. Roberts, Society for Industrial, and Applied Mathematics. *Graph Theory and Its Applications to Problems of Society*. CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1978. URL: https://books.google.cz/books?id=EYAwztXnzf8C.

[30] Denise Sakai. Labeling chordal graphs: Distance two condition. *SIAM Journal of Discrete Mathematics*, 7(1):133–140, 1994. doi:10.1137/S0895480191223178.

[31] Alan Shuchat, Randy Shull, Ann N. Trenk, and Lee C. West. Unit mixed interval graphs. *Congressus Numerantium*, 221:189–223, 2014. URL: https://repository.wellesley.edu/scholarship/98/.

[32] Alexandre Talon and Jan Kratochvíl. Completion of the mixed unit interval graphs hierarchy. *Journal of Graph Theory*, 87(3):317–332, 2018. doi:10.1002/jgt.22159.

List of Figures