$\mathcal{U}$-Bubble Model for Mixed Unit Interval Graphs and Its Applications: The MaxCut Problem Revisited

Jan Kratochvíl
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
honza@kam.mff.cuni.cz

Tomáš Masařík
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
masarik@kam.mff.cuni.cz

Jana Novotná
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Poland
Faculty of Mathematics and Physics, Charles University, Prague, Czech Republic
janca@kam.mff.cuni.cz

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 and demonstrate the advantages of this generalized model by providing a subexponential-time algorithm for solving the MaxCut problem on mixed unit interval graphs. In addition, we derive a polynomial-time algorithm for certain subclasses of mixed unit interval graphs. We point out a substantial mistake in the proof of the polynomiality of the MaxCut problem on unit interval graphs by Boyaci, Ekim, and Shalom (2017). Hence, the time complexity of this problem on unit interval graphs remains open. We further provide a better algorithmic upper-bound on the clique-width of mixed unit interval graphs.

2012 ACM Subject Classification Theory of computation → Graph algorithms analysis

Keywords and phrases Interval Graphs, Mixed Unit Interval Graphs, MaxCut Problem, Clique Width, Subexponential Algorithm, Bubble Model

Digital Object Identifier 10.4230/LIPIcs.MFCS.2020.57

Related Version A full version [22] of this paper is available at https://arxiv.org/abs/2002.08311.

Funding Tomáš Masařík and Jana Novotná received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme Grant Agreement 714704, and from Charles University student grant SVV-2017-260452.
Jan Kratochvíl was supported by grant GAČR 19-17314J of the Czech National Science Foundation.

Acknowledgements The authors would like to thank Vít Jelínek for helpful comments. This paper is based on the master thesis of Jana Novotná.

45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020).
Editors: Javier Esparza and Daniel Kráľ; Article No. 57; pp. 57:1–57:14
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

A graph G is an *intersection graph* if there exists a family of nonempty sets $\mathcal{F} = \{S_1, \ldots, S_n\}$ 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 $S_i \cap S_j \neq \emptyset$. We say that G has an *$\mathcal{F}$-intersection representation*. 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. Such a representation is being referred to as *interval representation*. They have been a well known and widely studied class of graphs from both the theoretical and the algorithmic points of view. 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 [20], Graph isomorphism [4] or Colorability [14] are solvable even in linear time. Surprisingly, the complexity of some well-studied problems is still unknown despite extensive research, e.g. the MaxCut problem, the $L_{2,1}$-labeling problem, or the packing coloring problem.

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 [25] 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 [13]. The mentioned results do not specifically care about what types of intervals (open, closed, semi-closed) are used in the interval representation. However, as far as there are no restrictions on lengths of intervals, it does not matter which types of intervals are used [27]. The same applies if there is only one type of interval 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. In particular, the claw $K_{1,3}$ can be represented using one open interval and three closed intervals.

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. 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. Let $\mathcal{U}$ be the set of all types of unit intervals. Although there are 16 different combinations of types of unit intervals, it was shown in [9, 24, 27, 18, 28] in the years 2012–2018 that they form only four different classes of mixed unit interval graphs. In particular, the following closure holds: $\emptyset \subsetneq$ unit interval $\subsetneq$ unit open and closed interval $\subsetneq$ semi-mixed unit interval $\subsetneq$ mixed unit interval $\subsetneq$ interval graphs, where *unit open and closed interval* graphs have $(\mathcal{U}^{++} \cup \mathcal{U}^{--})$-representation, *semi-mixed unit interval* graphs have $(\mathcal{U}^{++} \cup \mathcal{U}^{--} \cup \mathcal{U}^{-+})$-representation, and *mixed unit interval* graphs have $\mathcal{U}$-representation. Hence, mixed unit interval graphs allow all types of intervals of unit length.

▶ **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 interval representation.

There are lots of characterizations of interval and unit interval graphs. Among many of the characterizations, we single out a matrix-like representation called *bubble model* [16].

A similar notion was independently discovered by Lozin [23] under the name canonical partition. In the bubble model, vertices of a unit interval graph G are placed into a "matrix" where each matrix entry 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, edges are only between consecutive colums where they form nested neighborhood (two vertices u and v from consecutive colums are adjacent if and only if v occurs in a higher row than 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. We introduce a similar representation of mixed unit interval graphs, called *$\mathcal{U}$-bubble model*, and we extend some results from unit interval graphs to mixed unit interval graphs using this representation. 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.

▶ **Theorem 1.** *A graph is a mixed unit interval graph if and only if it has a $\mathcal{U}$-bubble model. Moreover, given a mixed unit interval representation of graph G on n vertices, a $\mathcal{U}$-bubble model can be constructed in $\mathcal{O}(n)$ time.*

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. 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 [3], the second one by Boyaci, Ekim, and Shalom which has been published in 2017 [5]. The result of the first paper was disproved by authors themselves a few years later [2]. In the second paper, the authors used a bubble model for proving the polynomiality. However, we realized that this algorithm is also incorrect. Moreover, it seems to us to be hardly repairable. We provide further discussion and also a concrete example, in Subsection 3.1.

Using the $\mathcal{U}$-bubble model, we obtain at least a subexponential-time algorithm for MaxCut in mixed unit interval graphs. We are not aware of any subexponential algoritms on interval graphs. In general graphs, there has been an extensive research dedicated to aproximation of MaxCut in subexponential time, see e.g. [1] or [17]. 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 [6] 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.

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

▶ **Corollary 3.** *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 $\mathcal{O}(n^5)$.*

Many NP-hard problems can be solved efficiently on graphs with bounded clique-width [8]. In general, it is NP-complete even to decide if the graph has clique-width at most k for a given number k, see [11]. Unit interval graphs are known to have unbounded clique-width [15]. It follows from results by Fellows et al. [10], and Kaplan and Shamir [19] that the clique-width of (mixed) unit interval graphs is upper-bounded by ω (the maximum size of their clique) $+1$. Heggernes et al. [16] improved this result for unit interval graphs using the bubble model.

There, the clique-width is upper-bounded by 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 $\omega + 1$. The upper-bound can be also expressed in the number of rows or colums of $\mathcal{U}$-bubble model. See Section 4 for further details. As a consequence, we obtain an analogous result to Corollary 3 for rows using the following result. Fomin, Golovach, Lokshtanov, and Saurabh [12] showed that the MaxCut problem can be solved in time $\mathcal{O}(n^{2t+\mathcal{O}(1)})$ where t is clique-width of the input graph. By combination of their result and our upper-bounds on clique-width (Theorem 8 in Section 4) we derive not only polynomial-time algorithm when the number of columns is bounded (with worse running time) but also a polynomial-time algorithm when the number of rows is bounded, formulated as Corollary 4.

▶ **Corollary 4.** *The size of a maximum cut in the graph class defined by $\mathcal{U}$-bubble models with k rows can be determined in the time $\mathcal{O}(n^{4k+\mathcal{O}(1)})$.*

Preliminaries and Notation. By a *graph* we mean a finite, undirected graph without loops and multiedges. Let G be a graph. We denote by $V(G)$ 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. By a *family* we mean a multiset $\{S_1, \dots, S_n\}$ which allows the possibility that $S_i = S_j$ even though $i \neq j$.

Let $x, y \in \mathbb{R}$ be real numbers. We call the set $\{z \in \mathbb{R} : x \le z \le y\}$ *closed interval* $[x, y]$, the set $\{z \in \mathbb{R} : x < z < y\}$ *open interval* (x, y), the set $\{z \in \mathbb{R} : x < z \le y\}$ *open-closed interval* $(x, y]$, and the set $\{z \in \mathbb{R} : x \le z < y\}$ *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 of unit length by $\mathcal{U}^{--}$, $\mathcal{U}^{++}$, $\mathcal{U}^{-+}$, and $\mathcal{U}^{+-}$, respectively. Formally, $\mathcal{U}^{++} := \{[x, x+1] : x \in \mathbb{R}\}$, $\mathcal{U}^{--} := \{(x, x+1) : x \in \mathbb{R}\}$, $\mathcal{U}^{+-} := \{[x, x+1) : x \in \mathbb{R}\}$, and $\mathcal{U}^{-+} := \{(x, x+1] : x \in \mathbb{R}\}$. We further denote the set of all unit intervals by $\mathcal{U} := \mathcal{U}^{++} \cup \mathcal{U}^{--} \cup \mathcal{U}^{+-} \cup \mathcal{U}^{-+}$. From now on, we will be speaking only about unit intervals. 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 $\ell(I) = \ell(J)$. The *type of an interval* I is a pair (r, s) where $I \in \mathcal{U}^{r,s}$, $r, s \in \{+, -\}$.

Let $G = (V, E)$ be a graph and $\mathcal{I}$ an interval representation of G. Let $v \in V$ be represented by an interval $I_v \in \mathcal{U}^{r,s}$, 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 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 [16] in 2009.

▶ **Definition 2** (Heggernes et al. [16], reformulated). *If A is a finite non-empty set, then a* 2-dimensional 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 $u \in B_{i,j}$, $v \in B_{i',j'}$, $|j - j'| \le 1$, 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 bubble structure $\mathcal{B}$ for V such that $G = G(\mathcal{B})$.*

▶ **Theorem 5** (Heggernes et al. [16]). *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 interval graphs where each set $B_{i,j}$ is split into four parts and edges are allowed also in the same row under specific conditions.

▶ **Definition 3.** *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 analogous notion for columns. We denote the index of the first row with a non-empty bubble as $\operatorname{top}(j) := \min \{ i \mid B_{i,j} \in \mathcal{B} \text{ and } B_{ij} \ne \emptyset \}$. 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 $i > i'$.

▶ **Definition 4.** *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 $u \in B_{i,j}^{*+}, v \in B_{i',j'}^{+*}$.*

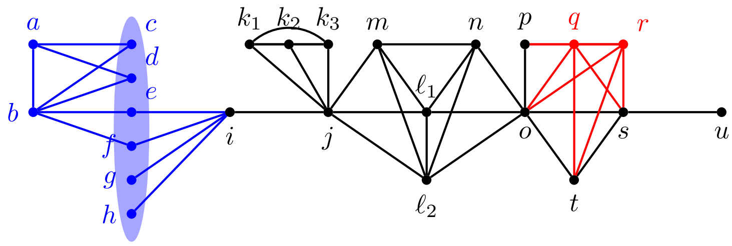
The definition says that the edges are only between vertices from the same or consecutive columns and if $u \in B_{i,j}$ and $v \in B_{i',j+1}$, there is an edge between u and v if and only if u is lower than v ($i > i'$), or they are in the same row and $u \in B_{i,j}^{*+}, v \in B_{i',j+1}^{+*}$. Vertices from the same column form a clique, as well as vertices from the same bubble. 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 5.** *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*

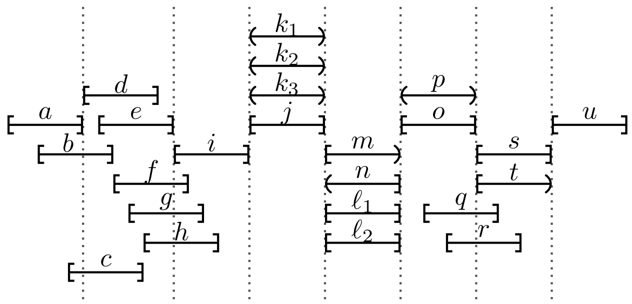
(i) *G is isomorphic to $G(\mathcal{B})$, and*
(ii) *each column and each row contains a non-empty bubble, and*
(iii) *no column ends with an empty bubble, and*
(iv) $\mathrm{top}(1) = 1$*, and for every* $j \in \{1, \dots, k-1\} : \mathrm{top}(j) \le \mathrm{top}(j+1)$.

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

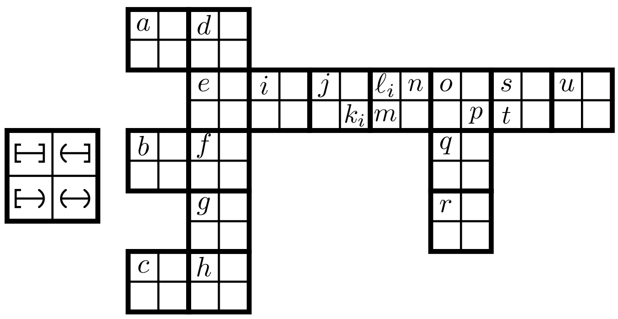
See Figure 1 with an example of a mixed unit interval graph, given by a mixed unit interval representation, and by a $\mathcal{U}$-bubble model.



(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 on the right, types of bubble quadrants on the left.

Figure 1 Three different representations of a mixed unit interval graph G.

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 by [28]; then each vertex of G is represented by a corresponding interval in $\mathcal{I}$. Having a mixed unit interval representation of the graph, our algorithm outputs a $\mathcal{U}$-bubble model for the graph in $\mathcal{O}(n)$ time.

Given a mixed unit interval representation $\mathcal{I}$, we put all intervals (vertices) that are almost-twins in $\mathcal{I}$ into a single bubble, to the particular quadrant which corresponds by its type to the type of the interval. From now on, we speak about bubbles only, we denote the set of all such bubbles by $\mathcal{B}$. We are going to determine their place (row and column) to create a 2-dimensional $\mathcal{U}$-bubble structure for $\mathcal{B}$. We show that the $\mathcal{U}$-bubble structure is a $\mathcal{U}$-bubble model for our graph. Based on the order σ by endpoints of intervals in the representation $\mathcal{I}$ from left to right, we obtain the same order on bubbles in $\mathcal{B}$. The idea of the algorithm is to process the bubbles in the order σ, and assign to each bubble its column immediately after processing it. During the processing, the algorithm maintains an auxiliary path in order to assign rows at the end. Thus, rows are assigned to each bubble after all bubbles are processed.

For bubbles $A, B \in \mathcal{B}$, $A <_\sigma B$ denotes that A is smaller than B in order σ. We denote the order of bubbles by subscripts, i.e., $B_1 <_\sigma B_2 <_\sigma \dots$ are all bubbles in the described order σ. For technical reasons, we create two new bubbles: B_{start}, B_{end} such that $\ell(B_{start}) = r(B_{start}) = -\infty$. We refer to them as *auxiliary bubbles*, in particular, if we speak about bubbles, we exclude auxiliary bubbles. We enhance the representation in a way that each bubble $B \in \mathcal{B}$ has a pointer $prev : \mathcal{B} \to \mathcal{B} \cup \{B_{start}\}$ defined as follows.

$$prev(B) = \begin{cases} B_{start} & \text{if } \ell(B) < r(B_1), \\ A \quad \text{s.t.} \quad \ell(B) = r(A) & \text{if such a bubble } A \text{ exists,} \\ B_j \quad \text{s.t.} \quad j = \max_i\{i \mid \ell(B) > r(B_i)\} & \text{otherwise.} \end{cases}$$

In order to set rows at the end, the algorithm is creating a single oriented path P that has the necessary information about the height of elements in the $\mathcal{U}$-bubble structure being constructed. Some of the arcs of the path can be marked with level indicator (L). For ease of notation, we use $next^P(B_i) = B_j$ to say that B_j is the next element on path P after B_i. Note that we can view P as an order of bubbles; we denote by $A <_P B$, $A, B \in \mathcal{B}$, the information that A occurs earlier than B on P. Also from technical reasons, P starts and ends with B_{start} and B_{end}, respectively. Except P and pointers $prev$ and $next^P$, the algorithm remembers the highest bubble of column i, denoted by C_i^{top}. Also, denote by curr, the index of the currently processed column.

Now, we are able to state the algorithm for assigning columns and rows to bubbles in $\mathcal{B}$ and its properties which will be useful for showing the correctness.

Property 1: Bubbles are processed (and therefore added somewhere to P) one by one respecting the order σ.

Property 2: The order induced by P of already processed vertices never changes, i.e., once $A \leq_P B$ then $A \leq_P B$ for the rest of the algorithm.

Property 3: The arc of P between bubbles A and B has the level indicator (L) if and only if $r(A) = \ell(B)$. Moreover, if the arc from A to B has level indicator, then $\mathrm{col}(A) < \mathrm{col}(B)$.

Property 4: $\mathrm{col}(B_i) \leq \mathrm{col}(B_j)$ whenever $i \leq j$.

Property 5: $prev(B)$ is the closest ancestor of B on P in the previous column, i.e., $prev(B) = \max\{A \mid A \leq_P B, \mathrm{col}(A) = \mathrm{col}(B) - 1\}$.

Property 6: The order induced by P of vertices in the same column is exactly the order of those vertices induced by σ.

Algorithm. Given bubbles $B_1, B_2, \ldots$ in $\mathcal{B}$ ordered by σ, the algorithm creates P by processing bubbles one by one in order σ. The algorithm outputs a row and a column to each bubble. Initially, set $\mathrm{col}(B_1) = 1$, $P = \{B_{start}, B_1, B_{end}\}$, curr =1 and $C_1^{\mathsf{top}} = B_1$. Suppose that $i-1$ bubbles have been already processed, for $i \geq 2$. Split the cases of processing bubble B_i based on the following possibilities:

- **i.** $\ell(B_i) > r(C_{\mathsf{curr}}^{\mathsf{top}})$: First increase curr by one, then set $\mathrm{col}(B_i) = \mathsf{curr}$ and $C_{\mathsf{curr}}^{\mathsf{top}} = B_i$.
- **ii.** $\ell(B_i) = r(C_{\mathsf{curr}}^{\mathsf{top}})$: First increase curr by one, then set $\mathrm{col}(B_i) = \mathsf{curr}$ and $C_{\mathsf{curr}}^{\mathsf{top}} = B_i$. Let Q be $next^P(C_{\mathsf{curr}-1}^{\mathsf{top}})$. Substitute arc in P from $C_{\mathsf{curr}-1}^{\mathsf{top}}$ to Q with two new arcs $C_{\mathsf{curr}-1}^{\mathsf{top}}$ to B_i that has L indicator set and from B_i to Q.
- **iii.** $\ell(B_i) < r(C_{\mathsf{curr}}^{\mathsf{top}})$: Set $\mathrm{col}(B_i) = \mathsf{curr}$.

We continue only with cases i. and iii. and distinguish multiple possibilities:

1. $r(prev(B_i)) = \ell(B_i)$: Let Q be $next^P(prev(B_i))$. Then substitute arc in P from $prev(B_i)$ to Q with two new arcs $prev(B_i)$ to B_i that has L indicator set and from B_i to Q.
2. $r(prev(B_i)) < \ell(B_i)$: And split this case further based on the properties of B_{i-1}.
   - **2a.** $prev(B_{i-1}) = prev(B_i)$: Let Q be $next^P(B_{i-1})$. Substitute arc in P from B_{i-1} to Q with two new arcs B_{i-1} to B_i and from B_i to Q.
   - **2b.** $prev(B_{i-1}) \neq prev(B_i)$: Let Q be $next^P(prev(B_i))$. Then substitute arc in P from $prev(B_i)$ to Q with two new arcs $prev(B_i)$ to B_i and from B_i to Q.

Now, assign rows to bubbles by a single run over P, inductively: Take the first bubble B of P and assign $\mathrm{row}(B) := 1$. Let B be the last bubble on P with already set row index. We are about to determine $\mathrm{row}(next^P(B))$. If arc in P from B to $next^P(B)$ has L indicator, set $\mathrm{row}(next^P(B)) := \mathrm{row}(B)$, otherwise $\mathrm{row}(next^P(B)) := \mathrm{row}(B) + 1$.

Proof of Theorem 1. The forward implication follows from the above algorithm. Its correctness is proved in the full version of the paper. Second, we prove the reverse implication: given a $\mathcal{U}$-bubble model for a graph G, we construct a mixed unit interval representation of G. Let $\mathcal{B} = \langle B_{i,j} \rangle_{1 \leq j \leq k, 1 \leq i \leq r_j}$ be a $\mathcal{U}$-bubble model of G. Let $\varepsilon := 1/(\max\{r_j \mid 1 \leq j \leq k\})$. 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: $I_v \in \mathcal{I}^{r,s}$ and $\ell(I_v) := j + (i-1)\varepsilon$. Note that all vertices from the same bubble are represented by intervals that are almost twins 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 $\mathcal{B}$. ◀

3 Maximum cardinality cut

This section is devoted to the time complexity of the MaxCut problem on (mixed) unit interval graphs. 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 determining the size of the maximum cut.

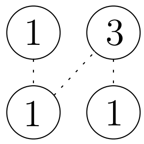


Figure 2 A counterexample to the original algorithm, a bubble model $\mathcal{B}$ where the numbers denote the number of vertices in each bubble, and dashed lines indicate the edges between bubbles.

3.1 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 [5], 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 realized that the algorithm is incorrect; this section is devoted to it.

We start with a counterexample to the original algorithm.

▶ **Example 6.** 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 2. In other words, this bubble model corresponds to a unit interval graph on vertices $v_1, v_2, v_3, v_4, v_5, v_6$ where there is an edge v_1v_2, and vertices v_2, v_3, v_4, v_5, v_6 create a complete graph without an edge v_2v_6.

Then, according to the paper [5], the size of a maximum cut in G is eight. To be more concrete, the algorithm from [5] 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. However, the size of a maximum cut in G is only seven. Suppose, for contradiction, that the size of a maximum cut is eight. As there are ten edges in total in G, 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 in [5] is to process the columns from the biggest to the lowest column from the top bubble to the 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].

We claim that the algorithm and its full idea from [5] is incorrect 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

[1] We refer the reader to the paper [5] for the notation and the description of the algorithm.

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. We attempted to correct the algorithm or extend the idea leading to the polynomiality. However, despite lots of effort, we were not successful and it seemed to us that the presented algorithm is hardly repairable. To conclude, the time complexity of the MaxCut problem on unit interval graphs is still not resolved and it seems to be a challenging open question.

3.2 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. Some of the ideas, for unit interval graphs, originated in discussion with Karczmarz, Nadara, Rzazewski, and Zych-Pawlewicz at Parameterized Algorithms Retreat of University of Warsaw 2019 [7].

Let us start with a notation. Let G be a graph, H be a subgraph of G, and S be a cut of H, we say that a cut X of G *agrees with* S in H if $X = S$ on H. Let G be a mixed unit interval graph. We take 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. 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 their two 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 $\mathcal{B}$, respectively), and we call their light columns *borders*. Heavy part might contain no heavy columns in the 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 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 the cut is in columns before. Furthermore, there are at most $\sqrt{n}$ consecutive heavy columns which allow 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, as stated in the following lemma. The formal proof is in the full version.

▶ **Lemma 7.** *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 that agrees with S in light columns is*

$$mcs(G, S) = \sum_{j=1}^{p} mcs(G(\hat{\mathcal{B}}_j), S_{j-1} \cup S_j) - \left(\sum_{j=1}^{p-1} |S_j| \cdot |C_j \setminus S_j|\right)$$

where $mcs(G(\hat{\mathcal{B}}_j), S_{j-1} \cup S_j)$ denotes the size of a maximum cut of $G(\hat{\mathcal{B}}_j)$ that agrees with $S_{j-1} \cup S_j$ in its borders C_{j-1}, C_j.

Now, our aim is to determine the size of a maximum cut for a heavy part $\hat{\mathcal{B}}$ given a fixed cut on its borders, which is stated in Theorem 6 (bellow). We provide only a sketch and key ideas here.Note that if $\hat{\mathcal{B}}$ is a heavy part with no heavy columns, we can straightforwardly count the number of cut edges of $G(\hat{\mathcal{B}})$, i.e., $mcs(G(\hat{\mathcal{B}}))$, 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 to determine the size of a maximum cut on each such heavy part. First, we present a brief idea of the dynamic programming approach. We take bubbles in $\hat{\mathcal{B}}$ which are not in borders and process them one-by-one in top-bottom, left-right order. When processing a bubble, we consider all the possibilities of numbers of cut-vertices in each its quadrant. We refer to the already processed part after i-th step as G_i, that is G_i is the the induced subgraph of $G(\hat{\mathcal{B}})$ with $V(G_i) = B_1 \cup \cdots \cup B_i \cup C_0 \cup C_{l+1}$ where C_0 and C_{l+1} are borders of $\hat{\mathcal{B}}$ and B_j, $j \in \{1, \ldots, i\}$ are first i bubbles in top-bottom, left-right order in $\hat{\mathcal{B}}$.

We store all possible $(l+1)$-tuples $(s_1, s_2, \ldots, s_l, a)$, where l is the number of heavy columns, s_j characterizes the number of all cut vertices in the j-th heavy column, and number a characterizes the number of cut vertices of types $(*, +)$ in the last processed bubble. Then, we define recursive function f_i which will be related to the maximum size of a cut that has exactly s_j cut vertices in column j (for all j) in the already processed part G_i. More precisely, we want the recursive function f to satisfy the following properties. For each stored tuple $s = (s_1, \ldots, s_l, a)$ and for every $i \in \{1, \ldots, m\}$, where m is the total number of bubbles in $\hat{\mathcal{B}}$, the value $f_i(s)$ is equal to the maximum size of a cut S in G_i that satisfies:

- 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 S agrees with $S_0 \cup S_{l+1}$ in $C_0 \cup C_{l+1}$, and
- a is equal to the number of cut vertices from $B_i^{++} \cup B_i^{-+}$,

or $f_i(s)$ is equal to $-\infty$ if there is no such cut.

Once, f satisfies the desired properties, we easily obtain Theorem 6 which gives us the size of a maximum cut in the heavy part. Due to space limitation the formal definition of the function f is in the full version. Here, we present a key observation for construction of f. Observe, by the properties of $\mathcal{U}$-bubble model, 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 inside $B_i\}$, $E_3 = \{$edges between B_i and the same column above $B_i\}$, $E_4 = \{$edges between B_i and the next column above $B_i\}$, $E_5 = \{$edges between B_i and the bubble in the previous column and the same row as $B_i\}$, $E_6 = \{$edges between B_i and column C_0 bellow $B_i\}$, $E_7 = \{$edges between B_i and the bubble in column C_{l+1} in the same row as $B_i\}$. The idea there is to count the size of a desired cut of G_i using the sizes of possible cuts in G_{i-1}, which are stored in f_{i-1}, and add the size of a cut using edges $E_2 - E_7$, which we can count from the number of cut vertices in currently processed bubble B_i and numbers in the $(l+1)$-tuple we are processing.

▶ **Theorem 6.** *Let $\hat{\mathcal{B}}$ be a heavy part with $l \geq 1$ 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}. Then, the size of a maximum cut in $G(\hat{\mathcal{B}})$ that agrees with $S_0 \cup S_{l+1}$ in light columns is $mcs(G(\hat{\mathcal{B}}), S_0 \cup S_{l+1}) = \max_{s \in T} f_m(s)$.*

Towards proving Theorem 2 and Corollary 3, it remains to prove the time complexity of processing a heavy part, see the full version for complete proofs.

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. 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 indistinguishably from the perspective of the newly added vertices. Formally, 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 four operations: creation of a labeled vertex, disjoint union of labeled graphs, renaming all labels i to j, and connecting all vertices with label i to all vertices with label j, $i \neq j$ (already existing edges are not doubled). Such a construction of a graph can be represented by an algebraic term composed of the operations, called *cwd-expression*.

We present here the main result for the better upper-bounds on clique-width which is inspired by a similar result for unit interval graphs [16]. In general, unit interval graphs (and therefore mixed unit interval graphs) have unbounded clique-width [15] and the known upper-bound (even for interval graphs) is the size of a maximum clique + 1 [19, 10]. The proofs can be found in the full version.

To state the main theorem, we need more notation. Let G be a mixed unit interval graph and 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 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 $A = B^{*+}_{i,j-1} \cup B^{+*}_{i,j}$ and $A = B_{i,j}$. Then the *group number of* G in $\mathcal{B}$ is defined as $\varphi_{\mathcal{B}}(G) := \max_{v \in V(G)} g_{\mathcal{B}}(v)$.

▶ **Theorem 8.** *Let G be a mixed unit interval graph. Then*

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

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.

Observe that $\varphi_{\mathcal{B}}(G) \le 2 \max \{r_j \mid 1 \le j \le k\}$. We also obtain a useful Corollary 7. In particular, if the number of rows or number of columns is bounded, than clique-width is bounded.

▶ **Corollary 7.** *Let G be a mixed unit interval graph. Then $cwd(G) \le \min \{k+3, 2r+2\}$, where k is the number of columns and r is the length of a longest column in a $\mathcal{U}$-bubble model for G.*

Note that by an application of Lemma 4.1 in [23], slightly worse bounds on clique-width in terms of rows and columns can also be derived. In particular, if we take two natural orderings of the bubbles in $\mathcal{U}$-bubble model, one taking rows first and the other taking columns first, we obtain two times larger multiplicative factor than in Corollary 7.

5 Conclusion

A long-term task is to determine the difference between the time complexity of basic problems on unit interval graphs compared to interval graphs. In particular, on a more precise scale of mixed unit interval graphs, determine what is a key property for the change of the complexity. Independently, a long-standing open problem is the time complexity of the MaxCut problem on unit interval graphs, in particular, decide if it is NP-hard or polynomial time solvable. An interesting direction to pursuit the first task could be the study of labeling problems; either $L_{2,1}$-labeling or Packing Coloring. Although, these are well-known problems, quite surprisingly, their time complexity is open for interval graphs. The complexity of $L_{2,1}$-labeling is still wide open even for unit interval graphs, despite partial progress on specific values for the largest used label [26]. 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 parameter k).

It is shown in [21] 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.

References

1 Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. *Journal of the ACM*, 62(5):1–25, November 2015. doi:10.1145/2775105.

2 Hans L. Bodlaender, Celina M. H. de Figueiredo, Marisa Gutierrez, Ton Kloks, and Rolf Niedermeier. Simple max-cut for split-indifference graphs and graphs with few p_4's. In Celso C. Ribeiro and Simone L. Martins, editors, *Experimental and Efficient Algorithms, Third International Workshop, WEA 2004, Angra dos Reis, Brazil, May 25-28, 2004, Proceedings*, volume 3059 of *Lecture Notes in Computer Science*, pages 87–99. Springer, 2004. doi:10.1007/978-3-540-24838-5_7.

3 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.

4 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.

5 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.

6 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.

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 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.

10 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.

11 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.

12 Fedor V. Fomin, Petr A. Golovach, Daniel Lokshtanov, and Saket Saurabh. Almost optimal lower bounds for problems parameterized by clique-width. *SIAM Journal on Computing*, 43(5):1541–1563, January 2014. doi:10.1137/130910932.

13 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.

14 Martin Charles Golumbic. Algorithmic graph theory and perfect graphs. *Computer Science and Applied Mathematics*, XX:284, 1980. doi:10.1016/C2013-0-10739-8.

15 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.

16 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.

17 Samuel B. Hopkins, Tselil Schramm, and Luca Trevisan. Subexponential lps approximate max-cut, 2019. arXiv:1911.10304.

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

19 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.

20 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.

21 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.

22 Jan Kratochvíl, Tomáš Masařík, and Jana Novotná. U-bubble model for mixed unit interval graphs and its applications: The maxcut problem revisited, 2020. arXiv:2002.08311.

23 Vadim V. Lozin. Minimal classes of graphs of unbounded clique-width. *Annals of Combinatorics*, 15(4):707–722, October 2011. doi:10.1007/s00026-011-0117-2.

24 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.

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

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

27 Alan Shuchat, Randy Shull, Ann N. Trenk, and Lee C. West. Unit mixed interval graphs. *Congressus Numerantium*, 221:189–223, 2014.

28 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.