Generalizing Roberts' Characterization of Unit Interval Graphs

Virginia Ardévol Martínez ✉ 🄳
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016 Paris, France

Romeo Rizzi ✉
Department of Computer Science, University of Verona, Italy

Abdallah Saffidine ✉
University of New South Wales, Sydney, Australia

Florian Sikora ✉ 🄳
Université Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016 Paris, France

Stéphane Vialette ✉ 🄳
LIGM, CNRS, Univ Gustave Eiffel, F77454 Marne-la-Vallée, France

───── **Abstract** ─────

For any natural number d, a graph G is a (disjoint) d-interval graph if it is the intersection graph of (disjoint) d-intervals, the union of d (disjoint) intervals on the real line. Two important subclasses of d-interval graphs are unit and balanced d-interval graphs (where every interval has unit length or all the intervals associated to a same vertex have the same length, respectively). A celebrated result by Roberts gives a simple characterization of unit interval graphs being exactly claw-free interval graphs. Here, we study the generalization of this characterization for d-interval graphs. In particular, we prove that for any $d \geqslant 2$, if G is a $K_{1,2d+1}$-free interval graph, then G is a unit d-interval graph. However, somehow surprisingly, under the same assumptions, G is not always a *disjoint* unit d-interval graph. This implies that the class of disjoint unit d-interval graphs is strictly included in the class of unit d-interval graphs. Finally, we study the relationships between the classes obtained under disjoint and non-disjoint d-intervals in the balanced case and show that the classes of disjoint balanced 2-intervals and balanced 2-intervals coincide, but this is no longer true for $d > 2$.

49th International Symposium on Mathematical Foundations of Computer Science (MFCS 2024).
Editors: Rastislav Královič and Antonín Kučera; Article No. 12; pp. 12:1–12:15
Leibniz International Proceedings in Informatics
LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Interval graphs are the intersection graphs of intervals on the real line: every vertex represents an interval and there is an edge between two vertices if and only if their corresponding intervals intersect. The class of interval graphs is one of the most important classes of intersection graphs, mostly due to their numerous applications in scheduling or allocation problems and in bioinformatics, see for examples these monographs [11, 23, 25].

Already in the late 70s, situations arising naturally in scheduling and allocation motivated the generalization of interval graphs to *multiple interval graphs*, where every vertex is associated to the union of d intervals on the real line (called a d-interval), for some natural number d, instead of to a single interval. This allowed a more robust modeling of problems such as multi-task scheduling or allocation of multiple associated linear resources [15, 22, 29], and led to several interesting problems [10, 12, 17, 18, 19]. The applications of 2-interval graphs to bioinformatics also increased the interest on this class of graphs [20, 30].

These concrete applications of multiple interval graphs, specifically 2-interval graphs, suggested a focus on different restrictions, such as unit 2-interval graphs [2], or balanced 2-interval graphs [7]. For both interval and multiple interval graphs, we say that they are *unit* if all the intervals in the representation, i.e. the set of intervals associated to the graph, have unit length. For multiple interval graphs, we also define the subclass of *balanced d-interval graphs*, where all intervals forming the same d-interval have equal length, but intervals of different d-intervals can have different lengths. Finally, for both interval and multiple interval graphs, we say that they are *proper* if there exists an interval representation where no interval properly contains another one. The class of unit 2-interval graphs is known to be properly contained in the class of balanced 2-interval graphs [13].

Let us remark that in the literature, d-intervals have been defined both as the union of d *disjoint* intervals [2, 5, 31], as the union of d *not necessarily disjoint* intervals [29], and simply as the union of d intervals, without specifying whether they are disjoint or not [9, 27]. This ambiguity is not relevant in the general case, since both definitions lead to the same class of graphs. However, in this paper we focus on subclasses of multiple interval graphs, namely unit and balanced, for which this equivalence is not known to be true. Therefore, we will distinguish between the two possible definitions of d-intervals. The first definition is denoted as *disjoint d-intervals* while the second is simply denoted as *d-intervals* (further details are discussed in Section 2).

From an algorithmic perspective, another reason why interval graphs have been widely studied is because many problems that are NP-hard become solvable in polynomial time when restricted to this class of graphs. This is not the case for d-interval graphs [2, 5, 12]. The problem of recognizing d-interval graphs is no exception: it is NP-complete for every natural number $d \geqslant 2$ [31], even for unit 2-interval graphs [1] and balanced 2-interval graphs [13]. In sharp contrast, the recognition of interval graphs (both in the unit and unrestricted case) can be done in polynomial time [4, 16], and there exist multiple characterizations of them, including a characterization in terms of forbidden induced subgraphs [21, 24]. In particular, in 1969, Roberts proved that the class of proper interval graphs and the class of unit interval graphs coincide [24], and showed that unit interval graphs are exactly $K_{1,3}$-free interval graphs (i.e., interval graphs that do not contain the star with three leaves as an induced subgraph). To do so, he used the Scott-Suppes characterization of semiorders (see [3, 14] for short constructive proofs of this result). This is a remarkable result as it gives a simple characterization of unit interval graphs. It also implies that if $G = (V, E)$ is a unit interval graph, then there exists a semiorder $S(V, P)$ on the vertices of V such that $(u, w) \in P$ if and

only if $(u, w) \notin E$, which justifies the original name of "indifference graphs" for unit interval graphs (as they can represent indifference relations by joining two elements by an edge if neither is preferred over the other one).

It is straight-forward to check that being $K_{1,3}$-free is a necessary condition for being a unit interval graph, as an interval of unit length cannot intersect three pairwise disjoint intervals of length one. The reader can observe that this necessary condition extends naturally to multiple interval graphs: a unit 2-interval graph cannot contain a $K_{1,5}$ as an induced subgraph; and more generally, a unit d-interval graph cannot contain a $K_{1,2d+1}$ as an induced subgraph. Thus the following natural question arises: can we generalize Roberts characterization of unit interval graphs to multiple interval graphs? Perhaps the most straight-forward generalization would be to characterize unit d-interval graphs as $K_{1,2d+1}$-free d-interval graphs, but this has already been proven false in [28]: there exists a graph which is 2-interval and $K_{1,5}$-free, but not unit 2-interval. But not all hope of generalizing Roberts characterization must be lost yet! What if we add some additional constraints?

Already in 2016, Durán et al. decided to focus on d-interval graphs which are also *interval* [8]. In a presentation at VII LAWCG, they claimed that if G is an interval graph, then G is a disjoint unit d-interval graph if and only if it is $K_{1,2d+1}$-free [1]. In this paper, we show that the aforementioned statement is actually false, and that, perhaps surprisingly, Roberts characterization can only be generalized depending on the chosen definition of d-interval graphs! (See Figure 1 for a summary of the main results).

We also study the subclasses obtained under the two definitions of d-intervals in the balanced case, expanding the knowledge of the relationships between the different subclasses of 2-interval graphs.
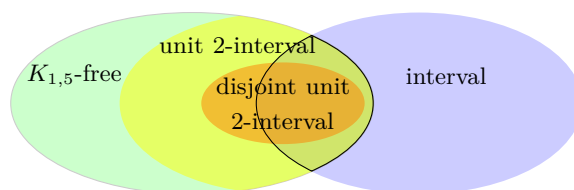
Figure 1 $K_{1,5}$-free interval graphs are not contained in the class of disjoint unit 2-interval graphs. The class of unit 2-interval graphs is a superclass of disjoint unit 2-interval graphs, and spans the whole intersection of $K_{1,5}$-free and interval graphs.

The structure of the paper is as follows: Section 2 briefly introduces the necessary definitions and discusses the definition of d-interval. In Section 3, we prove that if G is an interval graph, then it is unit d-interval if and only if it is $K_{1,2d+1}$-free. We then show that this result cannot be generalized for *disjoint* multiple intervals in Section 4, which implies that the class of disjoint unit d-interval graphs is actually properly contained in the class of unit d-interval graphs. Finally, we study the balanced case in Section 5, and show that the definition of d-interval also matters, as the classes of disjoint balanced 2-intervals and balanced 2-intervals coincide, but this is no longer true for $d > 2$. We conclude with some open questions in Section 6. Due to space constraints, some proofs, marked with a ($\star$), are deferred to the full version of this paper.

---

[1] Note that they refer to disjoint unit d-intervals simply as unit d-intervals, but they are explicitly defined beforehand as the union of d disjoint intervals.

2   Preliminaries

In the following, $G = (V, E)$ will denote a simple undirected graph on the set of vertices V and with edges E, and an interval will be a set of real numbers of the form $[a, b] := \{x \in \mathbb{R} \mid a \leqslant x \leqslant b\}$.

A graph G is an *interval graph* if there exists a bijection from the vertices of G to a multiset of intervals, $f : V \rightarrow \mathcal{I}$, such that there exists an edge between two vertices if and only if their corresponding intervals intersect. The multiset $\mathcal{I}$ is called an *interval representation* of G.

For any natural number $d > 0$, a (disjoint) *d-interval* is the union of d (disjoint) intervals on the real line.

For any natural number $d > 0$, a graph G is a (disjoint) *d-interval graph* if there exists a bijection from the vertices of G to a multiset of (disjoint) d-intervals, $f : V \rightarrow \mathcal{I}$, such that there exists an edge between two vertices if and only if their corresponding d-intervals intersect. The multiset $\mathcal{I}$ of d-intervals is called a *d-interval representation* of G, and the family of all intervals that compose the d-intervals in $\mathcal{I}$ is called the *underlying family of intervals* of $\mathcal{I}$.

A (disjoint) d-interval graph is *unit* if there exists a (disjoint) d-interval representation where all the intervals of the underlying family have unit length, and it is *proper* if there exists a representation where no interval of the underlying family is properly contained in another one. A (disjoint) d-interval graph is *balanced* if there exists a (disjoint) d-interval representation where the d intervals of a same d-interval have the same length, but intervals of different d-intervals can differ in length.

The graph $K_{1,t}$ is the star with t leaves (also referred to as *t-claw* in the following). For any $t \geqslant 3$, if the set of vertices $\{v_0, v_1, \ldots, v_t\}$ induces a $K_{1,t}$ with center v_0, we will denote it by $[v_0; v_1, \ldots, v_t]$. We say that a graph is $K_{1,t}$-free if it does not contain any induced $K_{1,t}$'s. Furthermore, we say that an induced t-claw $K_{1,t}$ is *maximal* if it is not contained in an induced $K_{1,m}$ with $m > t$.

Discussion on the definition of d-intervals.   As mentioned in the introduction, d-interval graphs have been defined in the literature both as the union of d disjoint intervals and as the union of d not necessarily disjoint intervals. This might be related to the fact that when there are no length restrictions on the intervals, both definitions lead to the same class of graphs (as one can simply stretch the intervals associated to a same vertex that intersect to make them disjoint without changing any of the other intersections).

▶ **Observation 1.** (⋆) *The classes of disjoint d-interval and d-interval graphs are equivalent.*

However, if there are length restrictions, the previous observation does not hold. For unit intervals, one cannot replace two intersecting intervals $[a, b]$ and $[c, d]$, with $a < c < b < d$, by $[a, d]$, as the resulting interval would not be of unit length, and stretching it to make it unit might disrupt the rest of the intersections. Thus, in this case, it cannot be inferred that both definitions of multiple intervals lead to the same class of graphs. In fact, our results prove that they do not. Therefore, we study the generalization of Roberts characterization separately for both definitions of d-intervals.

3   Unit d-interval graphs

In this section, we generalize Roberts characterization of unit interval graphs for d-interval graphs. Recall that by d-interval graphs we refer to intersection graphs of d-intervals where the d intervals are not necessarily disjoint, or in other words, to the most general definition.

▶ **Theorem 2.** *Let G be an interval graph. Then, for any natural number $d \geqslant 2$, G is a unit d-interval graph if and only if G does not contain a copy of a $K_{1,2d+1}$ as an induced subgraph. Furthermore, given a $K_{1,2d+1}$-free interval graph, a unit d-interval representation can be constructed in $\mathcal{O}(n + m)$ time, where n and m are the number of vertices and edges of the graph, respectively.*

We present a polynomial-time algorithm that, given an arbitrary interval representation $\mathcal{I}$ of a $K_{1,2d+1}$-free graph, returns a d-interval representation $\mathcal{I}'$ of the graph where no interval of the underlying family of $\mathcal{I}'$ intersects three or more pairwise disjoint intervals. This ensures that the underlying family of intervals returned corresponds to an interval representation of a $K_{1,3}$-free graph, so we can use the algorithm described in [3] to turn it into a proper representation (and then to a unit one in linear time [14]). Note that if an interval representation of the graph is not given, we can always compute it in linear time [6].

Before presenting the algorithm formally, let us give the idea behind it. The algorithm constructs a family $\mathcal{I}'$ of d-intervals in the following way: for every interval $I \in \mathcal{I}$ that intersects m (and no more than m) pairwise disjoint intervals, we create a t-interval $I_1 \cup \ldots \cup I_t$, where $t = \lceil \frac{m}{2} \rceil$. Note that for every interval I that intersect only two disjoint intervals, we have $t = 1$, and the interval I_1 added to $\mathcal{I}'$ will be exactly I. We will refer to such intervals as *original intervals*, as they are equal to the ones in $\mathcal{I}$. After creating the t-intervals described above, to obtain a d-interval representation of the graph, it suffices to add $d - t$ "dummy" intervals for each vertex that is represented by $t < d$ intervals (where by "dummy" intervals we mean that they do not intersect any other interval from the representation). Each d-interval $I_1 \cup \ldots \cup I_d$ introduced will preserve the same intersections as the interval $I \in \mathcal{I}$, and each I_i will possess three key properties: it intersects at most two disjoint original intervals, it contains an original interval, and each of its endpoints coincides with an endpoint of an original interval. These properties ensure that the representation $\mathcal{I}'$ can be made unit.

Algorithm

Let the family of intervals $\mathcal{I}$ be an interval representation of G. For every interval $I \in \mathcal{I}$, let $l(I)$ and $r(I)$ stand for its left and right endpoint, respectively. Furthermore, define a partial order as follows: given two intervals $I, J \in \mathcal{I}$, let $I \prec J$ if and only if $r(I) < l(J)$ (i.e. interval J is fully to the right of interval I). Two intervals are incomparable if they intersect.

Step 1 Initialize a set of intervals $\mathcal{C}$ with all the intervals of $\mathcal{I}$, set $\mathcal{I}' := \emptyset$, and go to **Step 2**.

Step 2 Pick an interval I of $\mathcal{C}$, remove it from the set and define its neighborhood $\mathcal{N}(I) = \{J \in \mathcal{I} : J \cap I \neq \emptyset\}$. Let m be the maximum number of pairwise disjoint intervals that I intersects. If $m \leqslant 2$, go to **Step 3**; if $m = 3$, go to **Step 4**; and if $m > 3$, go to **Step 5**.

Step 3 If $m \leqslant 2$, add the interval $I_1 = I$ to the family $\mathcal{I}'$ and call I_1 an original interval. Then go to **Step 6**.

Step 4 If $m = 3$, define four auxiliary intervals:

$$A_1 = \underset{J \in \mathcal{N}(I)}{\arg\min}\{r(J)\} \qquad\qquad A_2 = \underset{\{J \in \mathcal{N}(I)\,:\, A_1 \prec J\}}{\arg\min}\{r(J)\}$$

$$A_4 = \underset{J \in \mathcal{N}(I)}{\arg\max}\{l(J)\} \qquad\qquad A_3 = \underset{\{J \in \mathcal{N}(I)\,:\, J \prec A_4\}}{\arg\max}\{l(J)\}$$

Then add to $\mathcal{I}'$ the 2-interval $I_1 \cup I_2$, with $I_1 = [l(I), r(A_2)]$ and $I_2 = [l(A_3), r(I)]$. Note that A_2 and A_3 necessarily intersect, as otherwise we would have $m \geqslant 4$, so $I_1 \cup I_2$ is not a disjoint 2-interval. After adding it to $\mathcal{I}'$, go to **Step 6**.
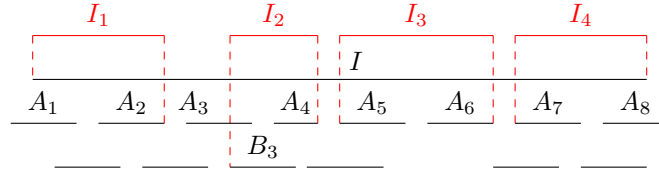
Figure 2 Interval I intersects 8 disjoint intervals. In red, the 4-interval returned by the algorithm. Note that if $l(I_2)$ were defined as $l(A_3)$ instead of $l(B_3)$, it would create a forbidden $K_{1,3}$.

Step 5 If $m > 3$, define two families of auxiliary intervals. The first family $\mathcal{A} := \{A_i \,|\, i \in \{1, \ldots, m\}\}$ forms a maximum set of pairwise disjoint intervals intersecting I, and it will ensure that all the intersections are preserved. It is defined as follows:

$$A_1 = \arg\min_{J \in \mathcal{N}(I)}\{r(J)\} \qquad A_i = \arg\min_{\{J \in \mathcal{N}(I)\,:\, A_{i-1} \prec J\}} \{r(J)\} \,, \, \forall \, i \in \{2, \ldots, m-2\}$$

$$A_m = \arg\max_{J \in \mathcal{N}(I)}\{l(J)\} \quad A_{m-1} = \arg\max_{\{J \in \mathcal{N}(I)\,:\, J \prec A_m\}} \{l(J)\}$$

The second family $\mathcal{B} := \{B_i \,|\, i \in \{1, \ldots, m\}\}$ is a tool to ensure that each new interval I_i intersects only two disjoint intervals in $\mathcal{I}'$. Note that restricting each I_i to intersect only two disjoint intervals from the family $\mathcal{A}$ is not enough: for example, in Figure 2, if I_2 were defined as $[l(A_3), r(A_4)]$, then it would intersect three pairwise disjoint intervals in $\mathcal{I}'$ (as all the intervals except I are original intervals in this example), whereas if the left endpoint of I_2 were chosen as $r(A_{2i-1})$, then an original interval that was not the center of a claw in $\mathcal{I}$ might become the center of a new claw in $\mathcal{I}'$. Thus, for every $i \in \{1, \ldots, m\}$, B_i is defined as follows:

$$B_i = \arg\max_{J \in \mathcal{N}(A_i) \cup A_i} \{l(J)\}$$

In other words, B_i is the interval in the closed neighborhood of A_i starting the latest. Note that if there does not exist any interval intersecting A_i which starts after A_i, then $B_i = A_i$ since we are considering the closed neighborhood. Now, add to $\mathcal{I}'$ the t-interval $I_1 \cup \ldots \cup I_t$, defined as follows. We distinguish two slightly different cases:

a. If m is even, i.e., $m = 2t$ for some $t > 1$, define $I_1 = [l(I), r(A_2)]$, $I_i = [l(B_{2i-1}), r(A_{2i})]$ for every $i \in \{2, \ldots, t-1\}$, and $I_t = [l(A_{2t-1}), r(I)]$.

b. If m is odd, i.e., $m = 2t-1$ for $t > 2$, define I_{t-1} and I_t differently, as $I_{t-1} = A_{2t-3}$ and $I_t = [l(A_{2t-2}), r(I)]$, and the rest of the intervals as before.

Notice that by definition, the intervals $I_1, \ldots, I_t$ are actually pairwise disjoint, so if $m > 3$, the t-interval added to $\mathcal{I}'$ is a disjoint d-interval. After adding the t-interval, go to **Step 6**.

Step 6 If $\mathcal{C} = \emptyset$, return $\mathcal{I}'$, else go to **Step 2**.

Figure 2 illustrates the algorithm on a concrete interval which intersects eight pairwise disjoint intervals. Before proceeding to the proof of correctness of the algorithm, we highlight the properties of the intervals constructed that will be useful to prove the next three claims. In the following, we say that an interval I of $\mathcal{I}$ has been *transformed* into a t-interval $I_1 \cup \cdots \cup I_t$ by the algorithm after it has been processed.

▶ **Observation 3.** *Let $I \in \mathcal{I}$ be an interval transformed into $I_1 \cup \cdots \cup I_t$ by the algorithm, for some $1 < t \leqslant d$. Then, for every $i \in \{1, \ldots, t\}$:*

1. *The left (resp., right) endpoint of every interval I_i coincides with the left (resp., right) endpoint of an original interval.*
2. *There is an original interval contained in I_i.*

Now, to prove the correctness of the algorithm, we need to show that for every interval $I \in \mathcal{I}$, the t-interval $I_i \cup \ldots \cup I_t \in \mathcal{I}'$ preserves the same intersections as I, and that no interval in the underlying family of $\mathcal{I}'$ intersects three pairwise disjoint intervals. In the next claim, we prove that intersections are preserved:

▷ **Claim 4.** Let I be an interval transformed into $I_1 \cup \cdots \cup I_t$ by the algorithm, for some $1 \leqslant t \leqslant d$. Then, the t-interval $I_1 \cup \ldots \cup I_t$ preserves the intersections of I.

Proof. It is clear that no new intersections are created as $I_1 \cup \ldots \cup I_t \subseteq I$. To see that no intersection is lost, suppose that there exists an interval L that intersects I in the original representation $\mathcal{I}$, and after the algorithm finishes, L is transformed into a t_0-interval $L_1 \cup \ldots \cup L_{t_0}$ (for some $1 \leqslant t_0 \leqslant d$, where if $t_0 = 1$, the interval remains as in the original representation) such that the t-interval $I_1 \cup \ldots \cup I_t$ does not intersect the t_0-interval $L_1 \cup \ldots \cup L_{t_0}$ in $\mathcal{I}'$. Since $l(I_1) = l(I)$ and $r(I_t) = r(I)$ (and the same holds for L), this means that there exists an L_j (with $1 \leqslant j \leqslant t_0$) such that $I_i \prec L_j \prec I_{i+1}$ for some $1 \leqslant i \leqslant t - 1$.

For $1 \leqslant t \leqslant 2$, this cannot occur because $I \subseteq I_1 \cup \ldots \cup I_t$. For $t > 3$, since the set of intervals A_k used to defined the t-interval associated to I forms a maximal set of pairwise disjoint intervals intersecting I, we cannot have that $A_k \prec L_j \prec A_{k+1}$ for any $1 \leqslant k \leqslant 2t - 1$. Indeed, this would contradict maximality, as L_j is either an original interval or it contains an original interval (by Observation 3). Thus, the only possible option is that there exists an i such that $A_{2i} \prec L_j \prec B_{2i+1}$ (where B_{2i+1} is different from A_{2i+1}). Then, since B_{2i+1} intersects A_{2i+1} and $L_j \prec B_{2i+1}$, we have that $r(L_j) < r(A_{2i+1})$. But this contradicts the choice of A_{2i+1}, which should have been L_j or the original interval contained in L_j, as $A_{2i} \prec L_j$.   ◁

The next two claims are dedicated to proving that no interval in the underlying family of $\mathcal{I}'$ intersects three or more pairwise disjoint intervals. We distinguish the cases when the center of the claw is an original interval and when it is not.

▷ **Claim 5.** Let $I \in \mathcal{I}$ be an original interval (i.e., transformed to I_1 by the algorithm). Then, I_1 intersects at most two disjoint intervals in the underlying family of $\mathcal{I}'$.

Proof. Suppose, towards a contradiction, that there exists an original interval I_1 that intersects three pairwise disjoint intervals L_1, L_2 and L_3 in the underlying family of $\mathcal{I}'$, with $L_1 \prec L_2 \prec L_3$. By Observation 3, there exists an original interval L_1' with the same right endpoint as L_1, an original interval L_2' contained in L_2, and an original interval L_3' with the same left endpoint as L_3. Note that if any of the L_i are original, then $L_i' = L_i$. But then, $L_1' \prec L_2' \prec L_3'$ are three pairwise disjoint original intervals that intersect I_1, which contradicts the fact that it is an original interval. Indeed, this implies that the interval I intersects three pairwise disjoint intervals in $\mathcal{I}$, and so the algorithm would have transformed it into a t-interval with t strictly greater than 1.   ◁

▷ **Claim 6.** Let $I \in \mathcal{I}$ be an interval transformed into the t-interval $I_1 \cup \cdots \cup I_t$ by the algorithm, for some $1 < t \leqslant d$. For every $1 \leqslant i \leqslant t$, I_i intersects at most two disjoint intervals of the underlying family of $\mathcal{I}'$.

Proof. We proceed by contradiction. Suppose that there exists an interval I_i, with $1 \leqslant i \leqslant t$ that intersects three pairwise disjoint intervals L_1, L_2, L_3, with $L_1 \prec L_2 \prec L_3$. By Observation 3, there exists an original interval L_1' with the same right endpoint as L_1, an original interval L_2' contained in L_2, and an original interval L_3' with the same left endpoint as L_3.

Assume first that $t = 2$. Then, if $i = 1$, this contradicts the choice of the interval A_2 (resp. A_3 if $i = 2$), which should have been L_2'.

Let us now study the general case for $t > 2$. Suppose first that $1 < t < d$ and I_i is defined as $[B_{2i-1}, A_{2i}]$ with $B_{2i-1} \neq A_{2i-1}$. We distinguish two cases:

1. $r(L_1) > r(A_{2i-1})$. Then, since we are assuming that L_1 and L_2 are disjoint, $l(L_2) > r(A_{2i-1})$. Furthermore, as L_3 also intersects I_i, we need $r(L_2) < r(A_{2i})$. But this contradicts the choice of A_{2i}, which should have been L_2'.

2. $r(L_1) < r(A_{2i-1})$. If $l(L_2) > r(A_{2i-1})$, we are in the same case as before. Thus, L_2 and A_{2i-1} must intersect. However, we have $l(L_2) > l(B_{2i-1})$ (since otherwise I_i would not be able to intersect L_1 on its left extreme). This contradicts the choice of B_{2i-1} if L_2' intersects A_{2i-1}, or the choice of A_{2i} otherwise.

On the other hand, if $B_{2i-1} = A_{2i-1}$, then by construction, since we take the two disjoint intervals that finish first, we cannot have three pairwise disjoint intervals intersecting I_i. This is also the case for I_1 and I_t (although in the latter case, we take the two disjoint intervals starting last). Finally, for odd claws, it is also clear that I_{t-1} intersects at most two disjoint intervals, as it is equal to an original interval. ◁

Combining Claims 4, 5 and 6, plus the fact that we can trivially transform a t-interval with $t < d$ into a d-interval, we obtain that the algorithm returns a d-interval representation of the input graph where no interval of the underlying family intersects more than two disjoint intervals, which as explained before can be converted into a unit representation. The last part of Theorem 2 follows because an efficient implementation of the algorithm described above requires $\mathcal{O}(1 + deg(v))$ operations for each vertex v (where $deg(v)$ denotes the degree of vertex v), as it suffices to iterate over the neighborhood of a given interval to transform it into the corresponding d-interval. Finally, the obtained representation can be converted to a unit representation in linear time, which yields the stated runtime $\mathcal{O}(n + m)$. This concludes the proof of Theorem 2.

We have proven that the algorithm constructs a unit d-interval representation, but it is not a disjoint one. Indeed, as mentioned before, in the case of maximal $K_{1,3}$'s, the constructed intervals I_1 and I_2 intersect each other. However, in the case of maximal $K_{1,m}$'s with $m > 3$, the t intervals of the t-interval created are actually pairwise disjoint. Thus, we obtain as a direct corollary that if G is a $K_{1,2d+1}$-free interval graph not containing any maximal $K_{1,3}$'s, then G is a disjoint unit d-interval graph. In fact, with a more careful analysis, we can infer an even stronger corollary, which instead of requiring the absence of maximal 3-claws altogether, only forbids a subset of them. We refer to these forbidden claws, which are exactly those maximal 3-claws contained in an induced E graph, as E-claws. Recall that an E graph (or $star_{1,2,2}$) is a graph on six vertices which has as edge set a path v_1, v_2, v_3, v_4, v_5 and an additional edge (v_3, v_6).

▶ **Corollary 7.** *Let G be a $K_{1,2d+1}$-free graph that does not contain any E-claws. Then, G is a disjoint unit d-interval graph.*

Proof. To prove the theorem, we modify **Step 4** of the previous algorithm so that it produces a disjoint 2-interval.

Step 4' Let I be an interval and let $m = 3$ be the maximum number of pairwise disjoint intervals that it intersects. By assumption, the vertex associated to I is a center of a maximal claw which is not an E-claw. We define

$$A_1 = \underset{J \in \mathcal{N}(I)}{\arg\min}\{r(J)\}$$

$$A_2 = \underset{\{J \in \mathcal{N}(I)\,:\,A_1 \prec J\}}{\arg\min}\{r(J)\}$$

$$A_4 = \underset{J \in \mathcal{N}(I)}{\arg\max}\{l(J)\}$$

$$A_3 = \underset{\{J \in \mathcal{N}(I)\,:\,J \prec A_4\}}{\arg\max}\{l(J)\}$$

Note that A_2 and A_3 necessarily intersect (or are the same interval). Now, since the vertex associated to I is a center of a claw that is not an E-claw, this means that at least one of A_1 or A_4 does not intersect an interval which is disjoint from I. Thus, we can modify the representation so that A_1 (resp. A_4) is properly contained in I without loosing any intersections, by simply stretching them. Then, if A_1 is properly contained in I, we define $I_1 = A_1$ and $I_2 = [l(A_3), r(I)]$. On the other hand, if A_4 is properly contained in I instead, we define $I_1 = [l(I), r(A_2)]$ and $I_2 = A_4$. If both of them are properly contained in I, we can define I_1 and I_2 either way.

Notice that the 2-intervals introduced in this step have the same properties as in Observation 3, so the proof of correctness of the previous algorithm can be directly adapted for this extension.   ◀

4   Disjoint unit d-interval graphs

In this section, we prove that Theorem 2 cannot be generalized for disjoint unit d-interval graphs. Note that by Corollary 7, if we have a graph which does not contain any E-claws, then the generalization still holds for disjoint unit d-interval graphs, but this is not the case in general. Indeed, suppose there is an interval I that intersects exactly three pairwise disjoint intervals, A_1, A_2 and A_3, and both A_1 and A_3 intersect each an interval disjoint from I. Then, the algorithm presented in the previous section would return a 2-interval $I_1 \cup I_2$, where I_1 and I_2 are not disjoint. If we try to extend the algorithm in the most natural way, that is, stretching these two intervals until they are disjoint, we would still not succeed. This is because, since I_1 and I_2 cannot intersect, then either $r(I_1)$ will be to the left of the right endpoint returned by the algorithm, or $l(I_2)$ will be to the right of the endpoint returned by the algorithm. But then, one of I_1 or I_2 might not properly contain a complete interval from the original representation, which can cause I_1 or I_2 – the interval which does not properly contain a complete original interval – to be contained in an interval that intersects three pairwise disjoint intervals (see Figure 3).

In the following, we show that there is no way to extend the algorithm to make it work in the general case for disjoint unit d-interval graphs. In particular, we prove the following theorem.

▶ **Theorem 8.** *There exists a $K_{1,5}$-free interval graph that is not a disjoint unit 2-interval graph.*
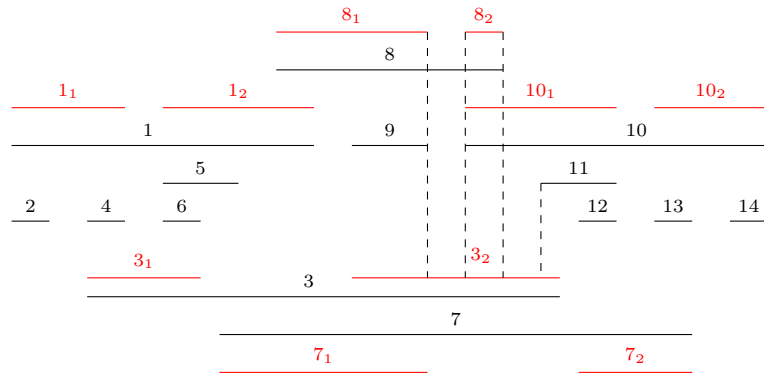
■ **Figure 3** Interval representation of a $K_{1,5}$-free graph that cannot be turned into a disjoint unit 2-interval representation just by "cutting" intervals that intersect more than three pairwise disjoint intervals. In the figure, the intervals in red are all obtained using a natural extension of the algorithm. We can see that in this way, 3_2 intersects three disjoint intervals: $8_1, 8_2, 11$. The reader can check that no other way of stretching the intervals works if 8_1 and 8_2 are required to be disjoint.

To prove Theorem 8, we offer the graph G in Figure 4 as a certificate. The reader can check that G has no induced $K_{1,5}$, and an interval representation of G is provided in Figure 5. The proof that G is not a disjoint unit 2-interval graph is the challenging part. Indeed, checking whether a graph is disjoint unit 2-interval is a computationally expensive task, and even with the aid of computer search, a naive ILP implementation already takes too much time to consider an exhaustive search. Needless to say, checking manually by brute force leads to a very long branching process. The proof presented here is based on a careful analysis of the graph, and the technique employed (which uses the characterization of unit 2-interval graphs in [[1], Lemma 5]) may be applied to establish that other graphs are not disjoint unit 2-interval graphs. We also verify the proof computationally, using an encoding in answer set programming based on the semiorder characterization of unit interval graphs, which proves to be way more efficient than an ILP encoding. Our code and experimental setting can be found on our git repository [2]. Furthermore, there exist five other $K_{1,5}$-free interval graphs on the same number of vertices, and with a very similar structure, that are not disjoint unit 2-interval (see the full version of this paper). The proof that they are not disjoint unit 2-interval is omitted, but it is analogous to the one presented here. These six graphs are the only such graphs on 14 vertices, and there does not exist a graph satisfying the conditions of Theorem 8 with fewer vertices. These assertions were verified by computer search over all interval graphs of a given size without induced $K_{1,5}$'s [32].

Theorem 8 follows directly from the next lemma.

▶ **Lemma 9.** (⋆) *The graph G in Figure 4 is not a disjoint unit 2-interval graph.*

We conclude this section showing that Theorem 8 can actually be generalized for disjoint unit d-interval graphs for any $d > 2$.

▶ **Corollary 10.** (⋆) *There exists a $K_{1,2d+1}$-free interval graph that is not a disjoint unit d-interval graph.*

---

[2] https://github.com/AbdallahS/unit-graphs
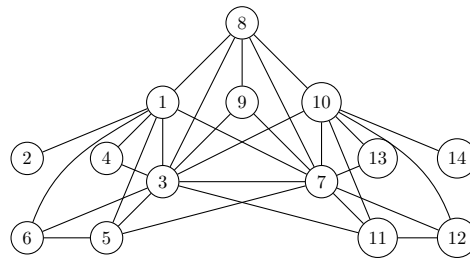
Figure 4 One of the 6 graphs with 14 vertices (the one with the fewest edges) which is an interval graph (see Figure 5) and $K_{1,5}$-free, but not disjoint unit 2-interval.
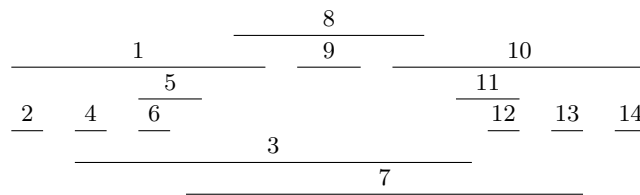
Figure 5 An interval representation of the graph in Figure 4.

5 Inclusions between the different subclasses of d-interval graphs

In this section, we analyze the relationships between different subclasses of multiple interval graphs. We have already seen that 2-interval graphs and disjoint 2-interval graphs are equivalent. Furthermore, the results from the previous two sections imply that the class of disjoint unit 2-interval graphs is properly contained in the class of unit 2-interval graphs. In the following, we summarize the containment relationships between unit 2-interval graphs, disjoint unit 2-interval graphs, balanced 2-interval graphs and disjoint balanced 2-interval graphs (see Figure 6 for a graphical illustration).
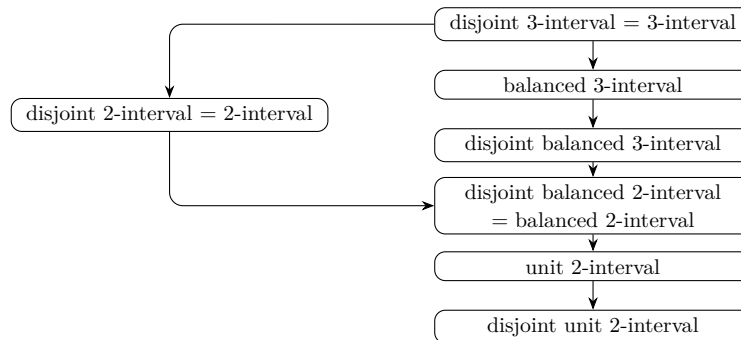
Figure 6 Landscape of graph subclasses of 3-interval graphs. An arrow from a graph class $\mathcal{C}$ to a $\mathcal{C}'$ indicates that $\mathcal{C}' \subset \mathcal{C}$. The relationships between the class of 2-interval graphs and the classes of balanced 3-interval graphs and disjoint balanced 3-interval graphs are not known.

▶ **Theorem 11.** (⋆)
1. *The classes of* 2-interval *and* disjoint 2-interval *graphs are* equivalent.
2. *The classes of* balanced 2-interval *and* disjoint balanced 2-interval *graphs are* equivalent.
3. *The class of* unit 2-interval graphs *is* properly contained *in the class of* disjoint balanced 2-interval *graphs.*
4. *The class of* disjoint unit 2-interval *graphs is* properly contained *in the class of* unit 2-interval *graphs.*

We finish by showing that the previous theorem cannot be completely generalized for the subclasses of d-interval graphs, as the class of balanced d-interval graphs is not equivalent to the class of disjoint balanced d-interval graphs for $d > 2$. We first construct a graph that is balanced 3-interval but not disjoint balanced 3-interval and then show how to generalize this construction for every $d > 3$.

▶ **Theorem 12.** *The class of disjoint balanced 3-interval graphs is properly contained in the class of balanced 3-interval graphs.*

Proof. We construct a graph G which is balanced 3-interval but not disjoint balanced 3-interval. The high-level idea of the construction is that for a particular vertex, one of its intervals is forced to a given length, while the other two are forced to be placed somewhere where there is not enough space for both of them, and thus they cannot be disjoint (note that the difference with the case $d = 2$ is that now, if we stretch two of the intervals so that they do not intersect, we also have to modify the length of the third interval, and as we show here, this is not always possible). To enforce these constraints, we use the complete bipartite graph $K_{11,4}$ as a gadget and exploit the fact that any 3-interval representation of this gadget must be continuous (i.e., the union of the intervals in its underlying family is an interval) [31, Lemma 2] (see also [13, Fig. 3] for the idea of its representation).

We construct G as follows: we connect in a chain five $K_{11,4}$'s, to which we add six vertices v_1, v_2, v_3, v_4, v_5, v_6 (Figure 7 shows how to link v_1, v_2, v_3, v_4 to the chain, while vertices v_5 and v_6 mimic the behavior of v_3 and v_4 with a different set of neighbors, namely, v_5 is connected to the corresponding vertices of the first two $K_{11,4}$'s, and v_6 is connected to the corresponding vertices of the second and the third $K_{11,4}$'s). More precisely, let C_i, with $i \in \{1, \ldots, 5\}$, be the five $K_{11,4}$'s forming the chain, enumerated from left to right. Moreover, for every C_i, let f_i^j with $j \in \{1, \ldots, 11\}$ be the eleven vertices of one side of the bipartition, and t_i^k, with $k \in \{1, 2, 3, 4\}$, the four vertices of the other side of the bipartition. We assume that the chain is connected such that f_i^{11} is linked to f_{i+1}^1. Then, v_1 is connected to all the vertices of C_2 and C_4, and to f_3^{11} and f_5^1, plus another independent vertex. Similarly, v_2 is connected to all the vertices of C_2 and C_4, to v_1 and to f_1^{11} and f_3^1. On the other hand, v_3 is connected to f_3^1, t_3^4, t_4^1 and f_4^j for $j \in \{1, \ldots, 9\}$; while v_4 is connected to f_5^1, t_4^4, t_5^1 and f_4^j for $j \in \{3, \ldots, 11\}$. Finally, v_5 is connected to f_1^{11}, t_1^3, t_2^1 and f_2^j for $j \in \{1, \ldots, 7\}$, as well as f_4^8 and f_4^9, whereas v_6 is connected to f_3^1, t_2^4, t_3^1 and f_2^j for $j \in \{5, \ldots, 11\}$, as well as f_4^8 and f_4^9. The vertices v_1 and v_2 are both connected to v_3, v_4, v_5 and v_6.

Now, as any 3-interval representation of a $K_{11,4}$ is continuous, any realization of G groups the five $K_{11,4}$'s in a block [31]. For $j \in \{1, 2, 3\}$, let I_j be the intervals associated to v_1, J_j the intervals associated to v_2, and K_j the intervals associated to v_3. First, it is clear that we need three different intervals to cover the neighbors of v_1 (and these three intervals must be disjoint). Instead, the neighbors of v_2 could be covered only with two intervals. However, we will see that the two segments of the real line that need to be covered cannot have the same length (assuming that the 3-interval associated to v_1 is balanced). We will show that we need two intersecting intervals to cover the first segment.

Suppose that only two intervals are needed to represent the adjacencies of v_2, and let J be the interval displaying the edges between C_2 and v_2, and J_3 the interval displaying the edges between C_4 and v_2. Similarly, let I_1 be the interval associated to v_1 used to represent the edges with C_2, let I_2 the interval used to represent the edges with C_4, and I_3 the interval displaying the edge with the isolated vertex. One can easily see that J_3 is properly contained in I_2 (since I_2 must also intersect an interval associated to f_3^{11} on its left and an interval associated to f_5^1 on its right), while I_1 is properly contained in J (by an analogous argument). Thus, $len(J_3) < len(I_2) = len(I_1) < len(J)$. In order for the representation to be balanced,
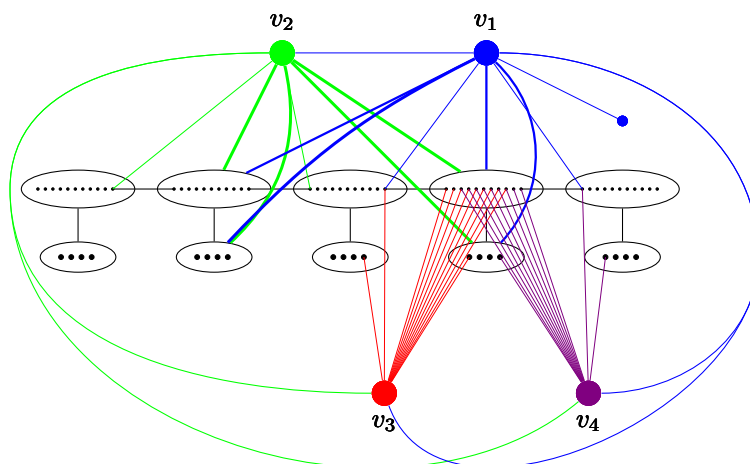
Figure 7 G is balanced 3-interval but not disjoint balanced 3-interval. $K_{11,4}$ graphs are drawn abstractly and are chained. A thick edge stopping at the border of the ellipse means that the vertex is connected to every vertex in the corresponding part of the $K_{11,4}$. Vertices v_5 and v_6 are omitted for readability purposes.

the segment of the real line covered by J needs to be covered by two different intervals, say J_1 and J_2. To prove that G is balanced 3-interval but not disjoint balanced 3-interval, we need to bound $len(J) - len(J_3)$. In particular, we need $len(J) - len(J_3) < len(J_3)$. Vertices v_3 and v_4 will allow us to find constants a and a' to bound $len(I_2) - len(J_3) \leqslant a + a'$, while vertices v_5 and v_6 will serve to find constants b and b' to bound $len(J) - len(I_1) \leqslant b + b'$. By showing that we can force the constants such that $a + a' + b + b' < len(J_3)$, we have the result. This will follow since we will have eight pairwise disjoint intervals properly contained in J_3: two of length a, two of length a', two of length b and two of length b'.

Indeed, let a and a' be the lengths of the intervals associated to v_3 and to v_4, respectively. The next claim implies that there are two disjoint intervals associated to v_3 properly contained in J_3, and another disjoint interval that properly contains the segment between $l(I_2)$ and $l(J_3)$, and so $l(J_3) - l(I_2) < a$.

$\triangleright$ **Claim 13.** ($\star$) Let G be a graph formed by the union of a $K_{11,4}$ and a vertex v which is adjacent to nine vertices in S_{11}, where S_{11} denotes the side of the bipartition with eleven vertices. Then, vertex v must be represented by three pairwise disjoint intervals, two of which are each properly contained in an interval representing a vertex of S_{11}.

Similarly, the segment between $r(I_2)$ and $r(J_3)$ is also contained in an interval associated to v_4, which has the same properties as v_3 and does not intersect any interval associated to v_3. This proves that there are two intervals of length a and two intervals of length a' (all pairwise disjoint) contained in J_3. Doing the same to bound $l(I_1) - l(J)$ and $r(J) - l(I_1)$, we get the result. Thus, to represent v_2, we need two intervals associated to v_2 to intersect. If we do not allow intersection, the length of these two intervals will be smaller than the length of the third interval associated to v_2, contradicting the fact that they are balanced. $\blacktriangleleft$

$\triangleright$ **Corollary 14.** ($\star$) *The class of disjoint balanced d-interval graphs is properly contained in the class of balanced d-interval graphs for every natural number $d \geqslant 3$.*

6   Concluding remarks

We have shown that the natural generalization of Roberts characterization for unit interval graphs remains valid for the most general definition of d-interval graphs that are interval graphs. However, quite surprisingly, if we require the d intervals to be disjoint, then the result does not hold anymore. It remains as an open question whether disjoint d-interval graphs that are also interval can be characterized in some other way, or simply if they can be recognized in polynomial time. Finally, we have obtained a relatively complete landscape of the containment relationships between different subclasses of 2-interval graphs, that cannot be fully generalized for $d > 2$. In particular, for $d > 2$, it is still unknown whether the class of unit d-interval graphs is contained in the class of disjoint balanced d-interval graphs.

References

1   Virginia Ardévol Martínez, Romeo Rizzi, Florian Sikora, and Stéphane Vialette. Recognizing unit multiple intervals is hard. In Satoru Iwata and Naonori Kakimura, editors, *34th International Symposium on Algorithms and Computation, ISAAC 2023*, volume 283 of *LIPIcs*, pages 8:1–8:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPICS.ISAAC.2023.8`.

2   Reuven Bar-Yehuda, Magnús M Halldórsson, Joseph Naor, Hadas Shachnai, and Irina Shapira. Scheduling split intervals. *SIAM J. Comput.*, 36(1):1–15, 2006.

3   Kenneth P. Bogart and Douglas B. West. A short proof that 'proper = unit'. *Discret. Math.*, 201(1-3):21–23, 1999. `doi:10.1016/S0012-365X(98)00310-0`.

4   Kellogg S. Booth and George S. Lueker. Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-Tree algorithms. *J. Comput. Syst. Sci.*, 13(3):335–379, 1976. `doi:10.1016/S0022-0000(76)80045-1`.

5   Ayelet Butman, Danny Hermelin, Moshe Lewenstein, and Dror Rawitz. Optimization problems in multiple-interval graphs. *ACM Trans. Algorithms*, 6(2):1–18, 2010.

6   Derek G. Corneil, Stephan Olariu, and Lorna Stewart. The LBFS structure and recognition of interval graphs. *SIAM J. Discret. Math.*, 23(4):1905–1953, 2009. `doi:10.1137/S0895480100373455`.

7   Maxime Crochemore, Danny Hermelin, Gad M. Landau, Dror Rawitz, and Stéphane Vialette. Approximating the 2-interval pattern problem. *Theor. Comput. Sci.*, 395(2-3):283–297, 2008. `doi:10.1016/j.tcs.2008.01.007`.

8   Guillermo Durán, Florencia Fernández Slezak, Luciano N. Grippo, Fabiano de S. Oliveira, and Jayme Luiz Szwarcfiter. On unit d–interval graphs. VII Latin American Workshop on Cliques in Graphs, 2016. URL: `https://www.mate.unlp.edu.ar/~liliana/lawclique_2016/ffslezak.pdf,https://www.mate.unlp.edu.ar/~liliana/lawclique_2016/prolist.pdf`.

9   Paul Erdös and Douglas B West. A note on the interval number of a graph. *Discret. Math.*, 55(2):129–133, 1985.

10 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009. `doi:10.1016/j.tcs.2008.09.065`.

11 Peter C. Fishburn. *Interval Orders and Interval Graphs: A Study of Partially Ordered Sets*. Wiley, 1985.

12 Mathew C. Francis, Daniel Gonçalves, and Pascal Ochem. The maximum clique problem in multiple interval graphs. *Algorithmica*, 71(4):812–836, 2015. `doi:10.1007/s00453-013-9828-6`.

13 Philippe Gambette and Stéphane Vialette. On restrictions of balanced 2-interval graphs. In Andreas Brandstädt, Dieter Kratsch, and Haiko Müller, editors, *Graph-Theoretic Concepts in Computer Science, 33rd International Workshop, WG 2007, Dornburg, Germany, June 21-23, 2007. Revised Papers*, volume 4769 of *LNCS*, pages 55–65. Springer, 2007. `doi:10.1007/978-3-540-74839-7_6`.

14 Frédéric Gardi. The Roberts characterization of proper and unit interval graphs. *Discret. Math.*, 307(22):2906–2908, 2007. `doi:10.1016/j.disc.2006.04.043`.

15 Jerrold R. Griggs and Douglas B. West. Extremal values of the interval number of a graph. *SIAM J. Algebraic Discret. Methods*, 1(1):1–7, 1980. `doi:10.1137/0601001`.

16 Michel Habib, Ross M. McConnell, Christophe Paul, and Laurent Viennot. Lex-BFS and partition refinement, with applications to transitive orientation, interval graph recognition and consecutive ones testing. *Theor. Comput. Sci.*, 234(1-2):59–84, 2000. `doi:10.1016/S0304-3975(97)00241-7`.

17 Minghui Jiang. On the parameterized complexity of some optimization problems related to multiple-interval graphs. *Theor. Comput. Sci.*, 411(49):4253–4262, 2010. `doi:10.1016/j.tcs.2010.09.001`.

18 Minghui Jiang. Recognizing d-interval graphs and d-track interval graphs. *Algorithmica*, 66(3):541–563, 2013. `doi:10.1007/s00453-012-9651-5`.

19 Minghui Jiang and Yong Zhang. Parameterized complexity in multiple-interval graphs: Domination, partition, separation, irredundancy. *Theor. Comput. Sci.*, 461:27–44, 2012. `doi:10.1016/j.tcs.2012.01.025`.

20 Deborah Joseph, Joao Meidanis, and Prasoon Tiwari. Determining DNA sequence similarity using maximum independent set algorithms for interval graphs. In *Scandinavian Workshop on Algorithm Theory*, pages 326–337. Springer, 1992.

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

22 Robert McGuigan. Presentation at NSF-CBMS Conference at Colby College, 1977.

23 Terry A McKee and Fred R McMorris. *Topics in intersection graph theory*. SIAM, 1999.

24 Fred S. Roberts. Indifference graphs. In F. Harary, editor, *Proof Techniques in Graph Theory*, pages 139–146. Academic Press, NY, 1969.

25 Fred S. Roberts. *Graph theory and its applications to problems of society*. SIAM, 1978.

26 Abdallah Saffidine. Unit 2-interval graph checker. Software, swhId: `swh:1:dir:3d7b6495d1f70618a537cd23c94530c23c030215` (visited on 2024-08-06). URL: `https://github.com/AbdallahS/unit-graphs`.

27 Edward R Scheinerman and Douglas B West. The interval number of a planar graph: Three intervals suffice. *J. Comb. Theory, Ser. B*, 35(3):224–239, 1983.

28 Alexandre Simon. Algorithmic study of 2-interval graphs. Master's thesis, Delft University of Technology, 2021.

29 William T. Trotter and Frank Harary. On double and multiple interval graphs. *J. Graph Theory*, 3(3):205–211, 1979. `doi:10.1002/jgt.3190030302`.

30 Stéphane Vialette. On the computational complexity of 2-interval pattern matching problems. *Theor. Comput. Sci.*, 312(2-3):223–249, 2004. `doi:10.1016/j.tcs.2003.08.010`.

31 Douglas B. West and David B. Shmoys. Recognizing graphs with fixed interval number is NP-complete. *Discret. Appl. Math.*, 8(3):295–305, 1984. `doi:10.1016/0166-218X(84)90127-6`.

32 Kazuaki Yamazaki, Toshiki Saitoh, Masashi Kiyomi, and Ryuhei Uehara. Enumeration of nonisomorphic interval graphs and nonisomorphic permutation graphs. *Theor. Comput. Sci.*, 806:310–322, 2020. `doi:10.1016/J.TCS.2019.04.017`.