Recognizing Unit Multiple Intervals Is Hard

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

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

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

Multiple interval graphs are a well-known generalization of interval graphs introduced in the 1970s to deal with situations arising naturally in scheduling and allocation. A d-interval is the union of d intervals on the real line, and a graph is a d-interval graph if it is the intersection graph of d-intervals. In particular, it is a unit d-interval graph if it admits a d-interval representation where every interval has unit length.

Whereas it has been known for a long time that recognizing 2-interval graphs and other related classes such as 2-track interval graphs is NP-complete, the complexity of recognizing unit 2-interval graphs remains open. Here, we settle this question by proving that the recognition of unit 2-interval graphs is also NP-complete. Our proof technique uses a completely different approach from the other hardness results of recognizing related classes. Furthermore, we extend the result for unit d-interval graphs for any $d \geqslant 2$, which does not follow directly in graph recognition problems –as an example, it took almost 20 years to close the gap between $d = 2$ and $d > 2$ for the recognition of d-track interval graphs. Our result has several implications, including that recognizing $(x, \dots, x)$ d-interval graphs and depth r unit 2-interval graphs is NP-complete for every $x \geqslant 11$ and every $r \geqslant 4$.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Mathematics of computing → Graph theory

Keywords and phrases Interval graphs, unit multiple interval graphs, recognition, NP-hardness

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2023.8

Related Version *Full Version*: https://arxiv.org/abs/2309.11908

Funding This work was partially supported by the ANR project ANR-21-CE48-0022 ("S-EX-AP-PE-AL").

Acknowledgements Part of this work was conducted when RR was an invited professor at Université Paris-Dauphine.

1 Introduction

Interval graphs are undirected graphs formed from a set of intervals on the real line, with a vertex for each interval and an edge between vertices whose intervals intersect. In particular, they are chordal and perfect graphs. Due to its numerous applications the class of interval graphs is one of the most well-studied classes of graphs [27, 12, 23]. These include DNA mapping [33], resource allocation problems in scheduling theory [1] and ecological niche and food web [6].

The practical applications of interval graphs have led to the study of various generalizations, including multiple interval graphs [22, 29, 16]. A graph is a d-interval graph if each vertex is associated with a d-interval (the union of d disjoint intervals on the real line) instead of a simple interval, and again, there is an edge between two vertices if and only if the corresponding d-intervals overlap at some point of the real line. This generalization enables us to model more complex situation arising naturally in scheduling and allocation problems, such as multi-task scheduling, allocation of multiple associated linear resources, or transmission of continuous-media data [2]. Applications to bioinformatics, namely to model DNA sequence similarity or RNA secondary structure [19, 30], increased the interest in this class of graphs.

Inside the class of multiple interval graphs, different restrictions have been studied. One of the most natural ones is the subclass of unit d-interval graphs, which corresponds to d-interval graphs that have an interval representation where every interval has unit length. Unit multiple intervals can be applied, for example, to model tasks of the same duration in scheduling.

Apart from their concrete applications, another reason why interval graphs have been widely studied in the literature is because many problems that are NP-hard in general graphs become polynomial-time solvable when restricted to interval graphs: colorability, clique, independent set, or Hamiltonian cycle, to name a few. In particular, recognizing interval graphs is also polynomial, and more precisely, it can be done in linear time [4, 7]. Furthermore, there exist multiple characterizations of interval graphs, including a characterization in terms of forbidden induced subgraphs [21]. This is also the case for unit interval graphs [25], which are exactly graphs that do not contain any claw, tent, net, or induced cycle of length at least 4. Unit interval graphs are also characterized as interval graphs that are claw-free [27].

However, for multiple interval graphs, most problems remain hard, even their recognition, and they do not have any simple characterization. In particular, they are neither chordal graphs nor perfect graphs. It is known that Maximum Clique remains NP-complete in multiple interval graphs, even for unit 2-intervals [13], and so do other problems such as Independent Set or Dominating Set [2, 5]. The parameterized complexity of some of these problems in multiple interval graphs has also been studied, see for instance [18, 10]. With respect to the recognition of multiple interval graphs, it was proven to be NP-hard in 1984 [31]. More precisely, West and Shmoys showed that determining whether the interval number of a graph (i.e., the smallest integer d such that the graph has a disjoint d-interval representation) is smaller or equal to d, for any $d \geqslant 2$, is NP-complete. Furthermore, they also proved that for any $r \geqslant 3$ and any $d \geqslant 2$, determining whether a graph has an r-depth d-interval representation (i.e., a d-interval representation with at most r intervals sharing a common point) is NP-complete. On the other hand, the complexity of recognizing depth 2 d-interval graphs is still open, although it is known to be polynomial for depth 2 unit d-interval graphs [18]. The above-mentioned proof of hardness (for unrestricted depth) was then adapted by Gambette and Vialette for balanced 2-intervals [15], which are 2-interval graphs that admit a representation such that every 2-interval is composed of two intervals of the same length, while intervals of different 2-intervals can have different lengths. In the same paper, the authors also initiate the study of the recognition of unit 2-interval graphs and of (x, x) 2-interval graphs (where the two disjoint open intervals have integer endpoints and have length x), but the complexity of both problems remained unsettled. Note that contrary to the previous characterization by Roberts of unit interval graphs, unit 2-interval graphs *cannot* be characterized as $K_{1,5}$-free 2-interval graphs [28].

Another well-studied generalization of interval graphs are d-track interval graphs, where each vertex is associated to the *union* of d disjoint intervals, each in a different parallel line called *track*. Gyárfás and West proved that their recognition is NP-hard for $d = 2$, and conjectured the same for $d \geqslant 3$ [17]. This conjecture was proven way later in [18] by Jiang, who also showed that recognition remains hard for unit d-track interval graphs for any $d \geqslant 2$, but left the recognition of unit d-interval graphs as an open question.

Multiple track interval graphs can be seen as the union of interval graphs. In the same manner, d-boxicity graphs can be seen as the *intersection* of interval graphs. Boxicity is a graph invariant introduced by Roberts [26] and it is the minimum dimension in which a graph can be represented as the intersection graph of boxes. Furthermore, given a graph $G = (V, E)$, it corresponds to the minimum number of interval graphs on the set of vertices V such that the intersection of their edge sets is G. Their recognition is NP-complete [8, 32], even for $d = 2$ [20].

In this paper, we finally settle the complexity of the recognition of unit 2-interval graphs, answering the open question by Jiang [18]. To do so, we prove that it is NP-hard by reducing from Satisfiability instead of Hamiltonian Path, which has been often used for proving the hardness of the recognition of variants of interval graphs. The reductions from Hamiltonian path in triangle-free cubic graphs used previously to prove the hardness of recognizing d-interval graphs, balanced d-interval graphs and d-track interval graphs all use a special vertex which is adjacent to n vertices of a triangle free graph, and therefore, cannot be directly adapted for unit 2-interval graphs. We then extend the hardness result for unit d-interval graphs, for any $d \geqslant 2$. Note that, as pointed out in the concluding remarks of [18], recognition problems are very different from optimization problems, and the boundary of a graph class is not necessarily harder than that of a subclass[1]. Thus, even though one would expect the recognition of unit d-interval graphs to be hard for any d if it's hard for $d = 2$, it is not directly implied.

Our result has several consequences, namely that recognizing $(x, \dots, x)$ d-interval graphs and depth r unit d-interval graphs is NP-complete for every $x \geqslant 11$ and every $r \geqslant 4$. Finally, our reduction implies as well a lower bound under the ETH.

Structure of the paper. The paper is organized as follows. Section 2 briefly introduces the necessary concepts and definitions. In Section 3, we present the results of the paper. First, in Subsection 3.1, we prove that a generalization of the recognition of unit 2-intervals, Colored unit 2-interval recognition, is NP-complete. Then, we use this result in Subsection 3.2 to prove the main theorem of the paper, which states the NP-completeness of Unit 2-interval recognition. Finally, we present several implications of our result in Subsection 3.3, namely the NP-completeness of recognizing unit d-interval graphs for every $d \geqslant 2$, and of recognizing $(x, \dots, x)$ d-interval graphs and depth r unit d-interval graphs for every $x \geqslant 11$ and every $r \geqslant 4$. We conclude with some directions for future work in Section 4.

Due to space constraints, some proofs (marked with $(\star)$) are deferred to the full version of this paper.

2 Definitions

An *interval* is a set of real numbers of the form $[a, b] := \{x \in \mathbb{R} \mid a \leqslant x \leqslant b\}$.[2]

A *d-interval* is the union of d disjoint intervals. A d-interval is *balanced* if all its d intervals have the same length, and *unit* when this common length is 1. A family $\mathcal{F}$ of d-intervals is *balanced* (resp., *unit*) if it comprises only balanced (resp., unit) d-intervals. Notice that,

[1] As an example, the class of $K_{1,5}$-free graphs, which admits a brute-force $\mathcal{O}(n^6)$ time recognition algorithm, contains the class of unit 2-track interval graphs, which is NP-hard to recognize [18].

[2] In the literature, it is not always specified whether the intervals considered for the intersection representation of interval graphs are open or closed. As discussed in [24], the reason for this might be that both definitions lead to the same class of finite graphs [14], even for unit interval graphs. However, note that if we allow the use of both open and closed intervals within one representation, then the class of unit interval graphs obtained is not the same as if we only allowed open or closed intervals within one representation [24].

for $d \geqslant 2$, different d-intervals of a same balanced family may comprise 1-intervals with different lengths. A family $\mathcal{F}$ of d-intervals can be used as a representation of the graph $\Omega(\mathcal{F})$ having the d-intervals of $\mathcal{F}$ as its vertex set, and where two d-intervals are adjacent if and only if their intersection is not empty. A graph G is called a (possibly balanced, unit) *d-interval graph* when it admits a representation $\mathcal{F}$ consisting only of (respectively balanced, unit) d-intervals. Notice that the representing family is not unique (in fact, even only by translating all intervals by a same value, we already obtain an infinite number of them). Multiple interval graphs generalize the standard notion of interval graphs (special case for $d = 1$). In this paper, we will use the term *unit 1-interval* (resp. *unit 1-interval graph*) to denote a classical unit interval (resp. a classical unit interval graph), to avoid confusion with a unit 2-interval (resp. unit 2-interval graphs).

Note that many references do not specify whether the intervals of a d-interval must be disjoint or not, and some even define them as the union of d not necessarily disjoint intervals [29]. However, this might be related to the fact that, when there are no restrictions on the length of the intervals, the two definitions lead to the same class of graphs. This is not true for unit d-intervals, so we study the case where disjointness is required, as in the hardness proof of recognizing multiple interval graphs [31].

A d-interval graph is *proper* when it admits a representing family $\mathcal{F}$ such that no 1-interval is properly contained in another one. The classes of proper and unit 1-interval graphs are equivalent, and they correspond exactly to $K_{1,3}$-free interval graphs. The graph $K_{1,3}$ is the star with 3 leaves, and is also called a *claw*. Equivalently, unit interval graphs are known to be exactly those graphs that do not contain any claw, tent, net, or cycle of length at least 4 as an induced subgraph [25].

A d-interval is a $(x_1, \ldots, x_d)$ d-interval if the d disjoint intervals are open, have integer endpoints, and have lengths $x_1, \ldots, x_d$, respectively.

The *depth* of a family of intervals is the maximum number of intervals that share a common point, and the *representation depth* of a d-interval graph is the minimum depth of any d-interval representation of the graph.

The hierarchy of subclasses of d-interval graphs is as follows [15, 18]: $(x, \ldots, x) \subset (x+1, \ldots, x+1) \subset$ *unit* $\subset$ *balanced* $\subset$ *unrestricted*.

The problem Unit 2-interval recognition is defined as follows.

> Unit 2-interval recognition
> **Input:** A graph $G = (V, E)$
> **Task:** Decide whether G has a unit 2-interval representation.

Furthermore, we define a more general version of the above problem, which will be useful to prove the hardness of Unit 2-interval recognition.

> Colored unit 2-interval recognition
> **Input:** A graph $G = (V, E)$ and a coloring $\gamma : V \to \{\texttt{white}, \texttt{black}\}$.
> **Task:** Decide whether G has a unit 2-interval representation where:
> - each white vertex is represented by a unit 2-interval,
> - each black vertex is represented by a unit 1-interval.
>
> We refer to this representation as a *colored unit 2-interval representation*.

3 Hardness of recognizing unit multiple interval graphs

In this section, we prove the main result of this paper, which is the hardness of recognizing unit 2-interval graphs, used later on to prove the hardness of recognizing unit d-intervals for every $d \geqslant 2$. The result for $d = 2$ is obtained in two steps. We first prove that the more general version Colored unit 2-interval representation is NP-complete, and then reduce this problem to Unit 2-interval recognition, which yields the main result of this paper.

3.1 Hardness of Colored Unit 2-Interval Recognition

Before proceeding to the hardness proof of Colored unit 2-interval recognition, we first introduce the variant of SAT that we will reduce from. In the following, we use the term "j-clause" to refer to a clause that contains exactly j literals.

▶ **Lemma 1** ([11]). ($\star$) *Satisfiability is* NP*-complete even when restricted to CNF-formulae such that:*

1. *Every clause contains either* 3 *literals (*3*-clause) or* 2 *literals (*2*-clause).*
2. *Each variable appears in exactly one* 3*-clause.*
3. *Each* 3*-clause is positive monotone, i.e., is comprised of three positive literals.*
4. *Each variable occurs exactly in three clauses, once negated and twice positive.*

We can now proceed to the proof of hardness of Colored Unit 2-Interval Recognition.

▶ **Theorem 2.** *Colored Unit 2-Interval Recognition is* NP*-complete, even for graphs of degree at most 6.*

The rest of the subsection is dedicated to the proof of Theorem 2. We first describe the construction used for the reduction and then prove its correctness.

Construction. Let Ψ be an instance of the variant of SAT described in Lemma 1, formed by a set of Boolean variables $x_1, \ldots, x_n$ and a set of clauses $C_1, \ldots, C_m$. We construct an equivalent instance (G_Ψ, γ_Ψ) of Colored unit 2-interval recognition as follows.

For every variable x_i, we introduce the variable gadget $\hat{V}_i$ (truth setting component), which is the vertex-colored graph on three black vertices A_i, B_i, C_i and three white vertices x_i^1, x_i^2 and x_i^N, with all edges between a black vertex and a white vertex, plus the edges (x_i^1, x_i^2), (C_i, A_i) and (C_i, B_i). We anticipate that the white vertices of $\hat{V}_i$ will be adjacent also to vertices outside $\hat{V}_i$; in order to underline this distinction, these three vertices are called *public*, and the black vertices are called *private*.

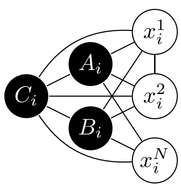

Figure 1 Variable gadget $\hat{V}_i$ corresponding to a variable x_i. Black vertices are displayed with a black background.

Figure 1 illustrates the variable gadget $\hat{V}_i$. Notice that the three white node x_i^1, x_i^2, x_i^N correspond each to precisely one of the occurrences of the represented variable x_i: vertex x_i^N represents the negated occurrence of x_i, vertex x_i^1 represents the positive occurrence in a 3-clause, and vertex x_i^2 represent the positive occurrence in a 2-clause. Therefore, we refer to them as *literal vertices*. Furthermore, note that a vertex of $\hat{V}_i$ is adjacent to A_i if and only if it is adjacent to B_i; and being private, these two nodes will remain false twins also in G. We will exploit this symmetry to simplify the case analysis.

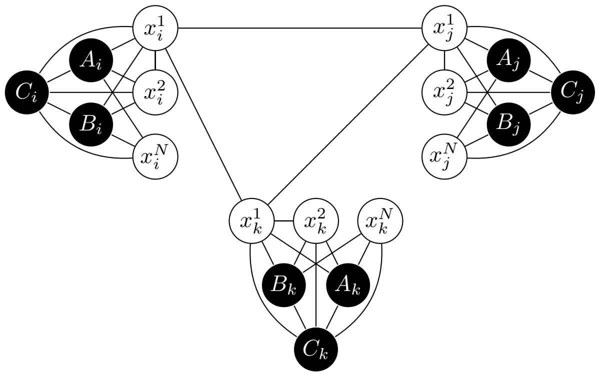

Figure 2 Clause gadget $\hat{C}_\alpha$ associated to a 3-clause $C_\alpha = (x_i \vee x_j \vee x_k)$. Note that in the final graph, each vertex x_i^m, x_j^m, x_k^m, for every $m \in \{1, 2, N\}$, will be incident to exactly 2 edges linking them to vertices outside their variable gadget.

To conclude the construction, we show how to encode each clause C_α, for $\alpha = 1, \ldots, m$. If C_α is a 3-clause, then it is monotone positive, i.e., $C_\alpha = (x_i \vee x_j \vee x_k)$ for some $i, j, k \in \{1, \ldots, n\}$, and all that is needed is to introduce the three edges (x_i^1, x_j^1), (x_j^1, x_k^1), (x_k^1, x_i^1). These three edges comprise the clause gadget (see Figure 2).

If C_α is a 2-clause, say $C_\alpha = (x_i^r \vee x_j^s)$ with $i, j \in \{1, \ldots, n\}$ and $r, s \in \{2, N\}$, then we introduce a public black vertex $L_{i,j}^\alpha$ with a private black neighbor $p_{i,j}^\alpha$ and we add the four edges (x_i^r, x_j^s), $(x_i^r, L_{i,j}^\alpha)$, $(x_j^s, L_{i,j}^\alpha)$ and $(L_{i,j}^\alpha, p_{i,j}^\alpha)$. These four edges together with the two vertices added comprise the clause gadget (see Figure 3).

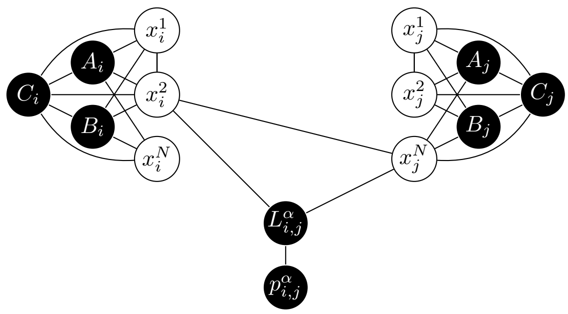

Figure 3 Gadget for a 2-clause $\hat{C}_\alpha$ of the form $C_\alpha = (x_i \vee \overline{x}_j)$.

The description of the reduction is complete. Clearly, G_Ψ has at most $6n + 2m$ vertices and at most $12n + 4m$ edges. We next introduce a few notions to ease the proof that G_Ψ is a colored unit 2-interval graph if and only if Ψ is satisfiable.

▶ **Definition 3.** *Given a colored graph (G, γ), we say that a pair (S, f) formed by a graph S and a function $f : V(S) \mapsto V(G)$ is a split of (G, γ) if f satisfies the following conditions:*

- $|f^{-1}(v)| = 1$ *for every $v \in V(G)$ with $\gamma(v) =$ `black`.*
- $|f^{-1}(v)| = 2$ *for every $v \in V(G)$ with $\gamma(v) =$ `white`.*
- *For every vertex v of G, $f^{-1}(v)$ is an independent set in S.*
- *For every edge (s, t) of S, $(f(s), f(t))$ is an edge of G.*
- *For every edge (u, v) of G, there exist two vertices s and t in $f^{-1}(\{u, v\})$ such that (s, t) is an edge of S.*

▶ **Definition 4.** *We define the family of splits of G that lead to a unit 1-interval graph as* $\mathcal{S}_{\mathcal{U}}(G) := \{(S, f) \mid (S, f)$ *is a split of G and S is a unit 1-interval graph*$\}$.

The next lemma shows how a split (S, f) of a colored graph G can be used to certify that G is a colored unit 2-interval graph. This has the advantage of being a truly combinatorial certificate, whereas the number of interval families representing a same graph is infinite with the power of the continuous as soon as at least one exists. Trotter and Harary [29] have already studied vertex splitting in the context of turning a graph into an interval graph.

▶ **Lemma 5.** ($\star$) *A colored graph (G, γ) is a colored unit 2-interval graph if and only if the family $\mathcal{S}_{\mathcal{U}}(G)$ is not empty.*

We can now proceed to study the shape of the possible splits $(S, f) \in \mathcal{S}_{\mathcal{U}}(G_\Psi)$. Let (S, f) be a split of a graph G. For every vertex $v \in V(G)$, we call each element of the set $f^{-1}(v)$ a representative of v. In particular, if v is a white node, we denote its two representatives in $V(S)$ by $f_1^{-1}(v)$ and $f_2^{-1}(v)$. For simplicity, when we refer to an arbitrary representative of a vertex or to the unique representative of a black vertex, we abuse notation and denote it by its label in $V(G)$. Furthermore, given an edge $(u, v) \in G$, we call the edge $(s, t) \in S$, a representative of (u, v) if $s \in f^{-1}(u)$ and $t \in f^{-1}(v)$. Furthermore, given a split (S, f) of the graph G_Ψ, we denote by $S[\hat{V}_i]$ the subgraph of S induced by the vertices of the variable gadget $\hat{V}_i$ (i.e., vertices A_i, B_i, C_i, $f_1^{-1}(x_i^N), f_1^{-1}(x_i^1)$, $f_1^{-1}(x_i^2)$, $f_2^{-1}(x_i^N), f_2^{-1}(x_i^1)$ and $f_2^{-1}(x_i^2)$). Finally, we say that a representative of a literal vertex is an *isolated vertex* if it is not adjacent to any of the private vertices of its variable gadget (i.e., it is not adjacent to A_i, B_i or C_i).

▷ Claim 6. Let (S, f) be an arbitrary graph in $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Then, none of the black vertices of $S[\hat{V}_i]$ can be adjacent to both representatives of a literal vertex. Furthermore, if a black vertex is adjacent to a representative of x_i^1 and to a representative of x_i^2, these two representatives must be adjacent to each other.

Proof. Suppose that the two representatives of a literal vertex are adjacent to the same black vertex. If the literal vertex is x_i^1 or x_i^2, the black vertex would be a center of a $K_{1,3}$ with these two representatives plus a representative of the vertex x_i^N as leaves. If the literal vertex is x_i^N, the black vertex would be a center of a $K_{1,3}$ with the two representatives of x_i^N and one of x_i^1 or x_i^2 as leaves. Since the graph $K_{1,3}$ is a forbidden induced subgraph for unit 1-interval graphs, this contradicts the fact that S belongs to $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Finally, if a black vertex is adjacent to a representative of x_i^1 and to a representative of x_i^2 which are not adjacent, the black vertex would be a center of a $K_{1,3}$ with these two representatives plus a representative of x_i^N as leaves. ◁

▷ Claim 7. Let (S, f) be an arbitrary split in $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Then, for every variable x_i with $i \in \{1, \ldots, n\}$, the subgraph $S[\hat{V}_i]$ satisfies at least one of the following two conditions, up to symmetry:

1. The vertex $f_1^{-1}(x_i^N)$ is adjacent to A_i and the vertex $f_2^{-1}(x_i^N)$ is adjacent to B_i.
2. The vertices $f_1^{-1}(x_i^1)$ and $f_1^{-1}(x_i^2)$ are adjacent to each other and to A_i, and the vertices $f_2^{-1}(x_i^1)$ and $f_2^{-1}(x_i^2)$ are adjacent to each other and to B_i.

Proof. By the properties of f, for every edge $(u, v) \in G_\Psi$, there exist elements $s, t \in V(S)$ with $f^{-1}(u) = s$ and $f^{-1}(v) = t$ such that (s, t) is an edge in S.

Suppose condition 1 does not hold, i.e., one of the representatives of x_i^N, say $f_1^{-1}(x_i^N)$, is adjacent to both A_i and B_i. We will show that if condition 2 does not hold either, S cannot be a unit 1-interval graph. Assume that one of the representatives of x_i^1 or x_i^2, say $f_1^{-1}(x_i^1)$

(resp. $f_1^{-1}(x_i^2)$), is adjacent to both A_i and B_i. Then, S contains an induced cycle of length four: $\big(f_1^{-1}(x_i^N), B_i, f_1^{-1}(x_i^1), A_i\big)$ (resp. $\big(f_1^{-1}(x_i^N), B_i, f_1^{-1}(x_i^2), A_i\big)$). This is a forbidden induced subgraph for unit 1-interval graphs, so it contradicts the hypothesis. Thus, it follows that, up to symmetry, vertices $f_1^{-1}(x_i^1)$ and $f_1^{-1}(x_i^2)$ need to be adjacent to A_i, and vertices $f_2^{-1}(x_i^1)$ and $f_2^{-1}(x_i^2)$, to B_i. Finally, by Claim 6, $f_1^{-1}(x_i^1)$ and $f_1^{-1}(x_i^2)$ need to be adjacent to each other, so condition 2 must hold.

Conversely, suppose condition 2 does not hold, i.e., at least one of the representatives of x_i^1 or x_i^2, say $f_1^{-1}(x_i^1)$ w.l.o.g., is adjacent to both A_i and B_i. We will see that condition 1 must hold in order for S to be a unit 1-interval graph. Indeed, if a single representative of x_i^N, say $f_1^{-1}(x_i^N)$, is adjacent to both A_i and B_i, then S contains an induced cycle of size four: $\big(f_1^{-1}(x_i^N), B_i, f_1^{-1}(x_i^1), A_i\big)$. Therefore, one representative of x_i^N must be adjacent to A_i and the other, to B_i. ◁

The previous claim implies that there are four possible configuration of $S[\hat{V}_i]$ such that it does not contain any induced cycles of length greater or equal to 4.

▶ **Lemma 8.** *Let (S, f) be a split of G_Ψ such that $S[\hat{V}_i]$ does not contain any induced cycles of length greater or equal to 4. Then, S satisfies one of the following conditions:*

1. *The vertex $f_1^{-1}(x_i^N)$ is adjacent to A_i and the vertex $f_2^{-1}(x_i^N)$ is adjacent to B_i, while for the rest of the literal vertices, there exists an element in the image via f^{-1} that is an isolated vertex.*
2. *The vertices $f_1^{-1}(x_i^1)$ and $f_1^{-1}(x_i^2)$ are adjacent to each other and to A_i, and the vertices $f_2^{-1}(x_i^1)$ and $f_2^{-1}(x_i^2)$ are adjacent to each other and to B_i, while $f^{-1}(x_i^N)$ contains an isolated vertex.*
3. *The images of x_i^1 and x_i^2 via f^{-1} are as in Case 1 and $f^{-1}(x_i^N)$ is as in Case 2 (see the graph in Figure 4).*
4. *Either the image of x_i^1 or the image of x_i^2 via f^{-1} is as in Case 1 (w.l.o.g., assume it is $f^{-1}(x_i^1)$) so that both representatives of x_i^1 are adjacent to the non-isolated representative of x_i^2; and $f^{-1}(x_i^N)$ is as in Case 2.*

Proof. We have already shown that one of the conditions of Claim 7 must hold. If condition 1 holds, then we have three possible configurations of $f^{-1}(x_i^1)$ and $f^{-1}(x_i^2)$: either both literal vertices have a representative that is isolated (Case 1), only one of them has a representative that is isolated (Case 4), or none of them has an isolated representative (Case 3). On the other hand, if condition 2 holds, the we only have two possible configurations of $f^{-1}(x_i^N)$: one representative of x_i^N is isolated (Case 2), or none of them is (Case 3). Finally, note that in Case 4, both representatives of x_i^1 need to be adjacent to the non-isolated representative of x_i^2 by Claim 6. ◀

The next two claims are devoted to proving that if (S, f) is a split of (G_Ψ, γ) contained in the family $\mathcal{S}_\mathcal{U}(G_\Psi)$, then Cases 3 and 4 of Lemma 8 are not possible. To do so, observe that by construction, since every variable appears exactly in three clauses (twice positive and once negated), we know that in G_Ψ, the vertices x_i^N, x_i^1 and x_i^2 all have two incident edges linking them with vertices outside of the variable gadget, called *external edges* in the following. The neighbors outside of the variable gadget are *external vertices*, and they constitute private neighbors of the vertices of the variable gadget, as it is not possible for two different vertices of the variable gadget to be incident to the same external neighbor. We will see that if S is as in Case 3 or Case 4, then the vertices of $S[\hat{V}_i]$ create an induced net with the external neighbors. Since nets are a forbidden induced subgraph for (unit) interval graphs, then S cannot be a unit 1-interval graph.

▷ Claim 9. Let S be an arbitrary graph in $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Then, for every variable x_i with $i \in \{1, \ldots, n\}$, the subgraph $S[\hat{V}_i]$ cannot be as in Case 3 of Lemma 8.

Proof. Suppose that $S[\hat{V}_i]$ is as in Case 3 of Lemma 8, i.e., as in Figure 4 (where C_i could be in the neighborhood of the other representatives of the vertices, but thanks to the symmetry, these cases are equivalent). We distinguish two cases:

- The two external edges incident to x_i^1 and x_i^2 are incident to two representatives that are adjacent. Then, either $f_1^{-1}(x_i^N), A_i, f_1^{-1}(x_i^1), f_1^{-1}(x_i^2)$, a private neighbor of $f_1^{-1}(x_i^1)$ and a private neighbor of $f_1^{-1}(x_i^2)$ form a net; or $f_2^{-1}(x_i^N), B_i, f_2^{-1}(x_i^1), f_2^{-1}(x_i^2)$, a private neighbor of $f_2^{-1}(x_i^1)$ and a private neighbor of $f_2^{-1}(x_i^2)$ form a net (see the red net in Figure 4).
- Otherwise, at least one of $f_1^{-1}(x_i^1)$ or $f_1^{-1}(x_i^2)$ will be incident to an external edge. Then, $C_i, A_i, f_1^{-1}(x_i^1)$ or $C_i, A_i, f_1^{-1}(x_i^2)$ will create a net together with $B_i, f_1^{-1}(x_i^N)$, and the corresponding external neighbor of $f_1^{-1}(x_i^1)$ or $f_1^{-1}(x_i^2)$, respectively (see the blue net in Figure 4).

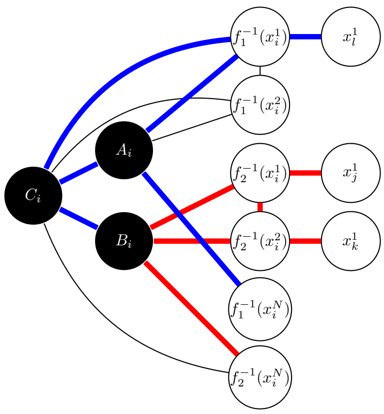

Figure 4 Configuration of $S[\hat{V}_i]$ described in Case 3 of Lemma 8. In red, the net created if both $f_2^{-1}(x_i^1)$ and $f_2^{-1}(x_i^2)$ have an external neighbor. In blue, the net created if $f_1^{-1}(x_i^1)$ has an external neighbor.

In both cases, we have a forbidden induced subgraph for (unit) interval graphs, contradicting the hypothesis that S is a unit interval graph.

◁

▷ Claim 10. (⋆) Let (S, f) be an arbitrary split in $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Then, for every variable x_i with $i \in \{1, \ldots, n\}$, the subgraph $S[\hat{V}_i]$ cannot be as in Case 4 of Lemma 8.

The proof of Claim 10 uses similar arguments to that of Claim 9 and is thus omitted here. Recall that in Case 1 of Lemma 8, one of the representatives of x_i^1 and one of the representatives of x_i^2 are isolated; and in Case 2 of Lemma 8, one of the representatives of x_i^N is isolated. Therefore, we obtain the following result.

▷ Claim 11. Let (S, f) be an arbitrary split in the family $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Then, for every variable x_i with $i \in \{1, \ldots, n\}$, the subgraph $S[\hat{V}_i]$ satisfies exactly one of the following two conditions:

1. There is a representative of x_i^1 and a representative of x_i^2 that are isolated vertices (they are either two non-adjacent vertices or they form a K_2).
2. One of the representatives of x_i^N is an isolated vertex.

(a) **(b)**

Figure 5 Representation of the variable gadget associated to the true value (left, 5a) or false value (right, 5b).

Proof. Combining Lemma 8 with Claims 9 and 10, it follows that $S[\hat{V}_i]$ is either as in Case 1 or as in Case 2 of Lemma 8, which means that either one representative of each of x_i^1 and x_i^2 is isolated, or that one representative of x_i^N is isolated, respectively. These options correspond to the interval representations in Figure 5a and Figure 5b, respectively. The reader can check the previous assertion observing the figures, and verify that the external edges incident to each of the vertices x_i^1, x_i^2 and x_i^N can be added in both representations, as we always have either a whole free interval (not depicted in the figures) or one extreme of the interval free for each of the vertices. ◁

The correctness of the reduction now follows from the two lemmas below.

▶ **Lemma 12.** *If Ψ is satisfiable, then the constructed graph $G_\Psi = (V, E)$, $V = V_{\mathtt{white}} \cup V_{\mathtt{black}}$, admits a colored unit 2-interval representation.*

Proof. Given a satisfying assignment ϕ of Ψ, we explain how to construct a colored unit 2-interval representation of G_Ψ, i.e., a collection of unit 2-intervals $\mathbf{D}_{\mathtt{white}} = \{(I_1(v), I_2(v)) \mid v \in V_{\mathtt{white}}\}$ and a collection of unit 1-intervals $\mathbf{I}_{\mathtt{black}} = \{I_1(v) \mid v \in V_{\mathtt{black}}\}$ such that $G \simeq \Omega(\mathbf{D}_{\mathtt{white}} \cup \mathbf{I}_{\mathtt{black}})$. Note that by Lemma 5, if G_Ψ is a colored unit 2-interval graph, then there exists a split (S, f) in the family $\mathcal{S}_\mathcal{U}(G_\Psi)$, and we know how to construct a colored unit 2-interval representation of G_Ψ given a unit 1-interval representation of S by defining the 2-interval associated to a white vertex $v \in V_{\mathtt{white}}$ as the union of the interval associated to $f_1^{-1}(v)$ and the interval associated to $f_2^{-1}(v)$; and the 1-interval associated to a black vertex $v \in V_{\mathtt{black}}$ as the interval associated to the single vertex $f^{-1}(v)$.

For each variable x_i with $i \in \{1, \ldots, n\}$, if $\Phi(x_i) = \mathit{true}$, we represent the variable gadget $\hat{V}_i$ as shown in Figure 5a, which corresponds exactly to Case 1 of Claim 11. On the other hand, if $\Phi(x_i) = \mathit{false}$, we represent $\hat{V}_i$ as in Figure 5b, which corresponds to Case 2 of Claim 11. Notice that in both representations, the literals that are true have an isolated representative, i.e., one of the intervals associated to them is unused in the representation of $\hat{V}_i$ and remains completely free to display intersections with external neighbors.

After this, it only remains to explain the connections introduced by the clauses.

▷ Claim 13. Given a 3-clause $(x_i \vee x_j \vee x_k)$, there exists a unit interval representation of the subgraph of G_Ψ induced by the vertices of the variable gadgets $\hat{V}_i, \hat{V}_j$ and $\hat{V}_k$.

Proof. Each of the variable gadgets can be represented as in Figure 5a or Figure 5b. To represent the edges associated to the 3-clauses, we first notice that, since the 3-clauses are positive monotone, true literals correspond to true variables. As we are assuming that we have a satisfying assignment, we only have three cases (up to symmetry), which correspond to the three variables being true; exactly two variables being true; and only one variable being true. The literals that are true have a whole free interval to display the intersection, whereas the literals that are false only have the extreme of an interval (while the other extreme is

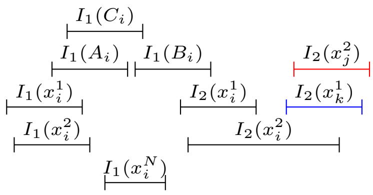

Figure 6 Representation of a 3-clause $(x_i \vee x_j \vee x_k)$, where x_i is set to false while x_j, x_k are set to true.

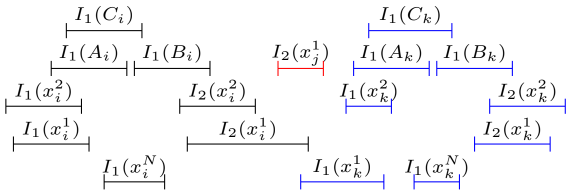

Figure 7 Representation of a 3-clause $(x_i \vee x_j \vee x_k)$, where x_i and x_k are set to false and x_j is set to true.

glued to the rest of the representation of the gadget, see Figure 5b). Let $(x_i \vee x_j \vee x_k)$ be a 3-clause, with $i, j, k \in \{1, \ldots, n\}$. If the three variables are true, we can easily represent the clause by making the three free intervals of the variables – w.l.o.g. $I_2(x_i^1), I_2(x_j^1), I_2(x_k^1)$ – intersect at the same time. On the other hand, if only one variable – say x_i – is false, we can add the two free intervals $-I_2(x_j^1), I_2(x_k^1)$ – to the corresponding extreme of the gadget of the false variable, as in Figure 6. Finally, if two variables are false – say x_i, x_k –, then we need to merge the two interval representations associated to their gadgets and add the free interval – $I_2(x_j^1)$ – in the middle, as in Figure 7. Note that the interval representations given in the figures are not unit, but they are proper, so at the end we will be able to use the algorithm described in [3] to turn it into a unit one. ◁

After representing all the 3-clauses, we can assume that the representations of some of the variable gadgets have been merged two by two (we will never have to merge a gadget more than once since a variable occurs in exactly one 3-clause in Ψ) and we can fix them in the real line separated from one another. The separation between them can be arbitrarily large, and needs to be at least greater than the space needed to place the remaining intervals. The variable gadgets that have not been merged can also be fixed in the real line, while the unused free intervals (corresponding to true literals), the intervals $I_1(L_{i,j}^{\alpha})$, and the intervals $I_1(p_{i,j}^{\alpha})$ remain unplaced.

Now, to display the 2-clauses, we distinguish two cases. First, if both literals are true, then there exists a free interval for each, and we can represent the clause in a separate part of the real line (there is one $L_{i,j}^{\alpha}$ and one $p_{i,j}^{\alpha}$ per clause, so these intervals will never cause a problem). Secondly, if one of the literals is false, then the free interval associated to the true literal needs to be glued to the extreme of the representation of the variable gadget of the false one. Note that there is always one free extreme because the 3-clauses use at most one extreme per variable gadget (and we can extend $I_j(x_i^2)$ to allow the intersection while keeping the representation proper). Note also that we will never need more than two extremes to obtain a representation because, since each variable occurs twice positive and once negated, we can have at most two false literals (when the variable is set to false).

Since we have constructed a proper interval representation, we can now use the algorithm described in [3] to turn the representation into a unit one, as mentioned before. ◀

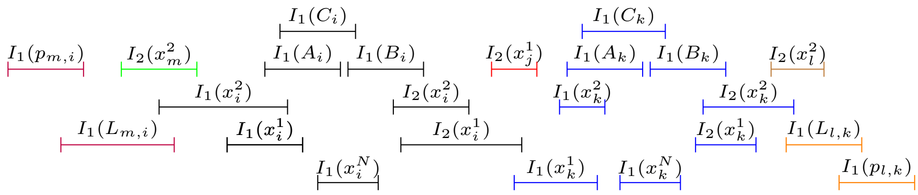

Figure 8 Representation of a longest contiguous block of intervals, where each color represents the intervals associated to a different variable. A longest contiguous block occurs when there is a clause $(x_i \vee x_j \vee x_k)$, where x_i and x_k are set to false and both of them also appear as positive literals in a 2-clause.

Let us now prove the converse implication.

▶ **Lemma 14.** *If the constructed graph $G_\Psi = (V, E)$, $V = V_{white} \cup V_{black}$, admits a colored unit 2-interval representation, then the original formula Ψ is satisfiable.*

Proof. Assume that the constructed graph G_Ψ admits a colored unit 2-interval representation where black vertices are represented by unit 1-intervals and white vertices are represented by unit 2-intervals. As in Claim 11, we study the splits $(S, f) \in \mathcal{S}_{\mathcal{U}}(G_\Psi)$.

We have already seen in Claim 11 that there are only two possible configurations for $S[\hat{V}_i]$, up to symmetry. Let us assign a truth value to each of the configurations. If $S[\hat{V}_i]$ satisfies condition 1 of Claim 11, we set $\Phi(x_i) = true$. Otherwise, if it satisfies condition 2 of Claim 11, then we set $\Phi(x_i) = false$. Recall that this implies that there is a representative of the vertices representing true literals which remains isolated from its variable gadget.

The following claims restrict the structure of a representable clause gadget. Both use similar arguments, so only the first proof is included here. Given a clause gadget $\hat{C_\alpha}$ in G, we define the clause gadget $S[\hat{C_\alpha}]$ in S as the set of representatives of the edges and vertices of $\hat{C_\alpha}$.

▷ Claim 15. Let (S, f) be an arbitrary split in $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Then, for every 3-clause, at least one of the representatives of the literal vertices incident to the clause gadget in S must be an isolated vertex.

Proof. Towards a contradiction, we assume that there exists a 3-clause gadget in S such that none of the representatives of the literal vertices adjacent to the clause gadget are isolated. Let $C_\alpha = x_i \vee x_j \vee x_k$, with $i, j, k \in \{1, \ldots, n\}$ be a (monotone positive) 3-clause. Each of the literal vertices has two external neighbors. In S, either the two external neighbors are incident to the same representative of the literal vertices (and thus only one representative is incident to the clause gadget), or each of them is incident to a different representative. We distinguish two cases, depending on whether only one representative of each literal vertex is incident to the clause gadget, or whether there is at least one literal vertex such that both of its representatives are incident to the clause gadget:

- If only one representative of each literal vertex is incident to the clause gadget in S, then w.l.o.g., the clause gadget is formed by edges $\{(f_1^{-1}(x_i^1), f_1^{-1}(x_j^1)), (f_1^{-1}(x_j^1), f_1^{-1}(x_k^1)), (f_1^{-1}(x_k^1), f_1^{-1}(x_i^1))\}$. By assumption, none of the vertices incident to the clause gadget in S are isolated, so they are all connected to at least one black vertex of their variable gadget. Thus, without loss of generality, $\{f_1^{-1}(x_i^1), f_1^{-1}(x_j^1), f_1^{-1}(x_k^1), A_i, A_j, A_k\}$ form a net (the readers can convince themselves looking at Figure 2). Note that when we say without loss of generality, we are using the symmetry between A_i and B_i.

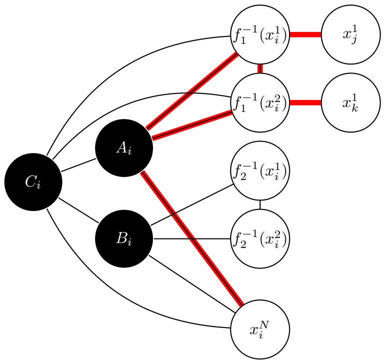

Figure 9 In red, the net created if both representatives of x_i^1 are incident to the clause gadget in S and $f_1^{-1}(x_i^2)$ is incident to an external edge.

- If there is at least one literal vertex such that both of its representatives are incident to the clause gadget, then w.l.o.g., the clause gadget in S contains edges $\{(f_1^{-1}(x_i^1), f_1^{-1}(x_j^1)), (f_1^{-1}(x_k^1), f_2^{-1}(x_i^1))\}$ (and eventually, edges between representatives of x_j^1 and x_k^1). Then, since one of the representative of x_i^2 also has a private neighbor outside of the variable gadget, either the subgraph induced by $\{A_i, f_1^{-1}(x_i^1), f_1^{-1}(x_i^2)\}$ or the subgraph induced by $\{B_i, f_2^{-1}(x_i^1), f_2^{-1}(x_i^2)\}$ (and one private neighbor of each of the three vertices, where the private neighbor of A_i and B_i is x_i^N) is a net. This situation is depicted in Figure 9.

In both cases, the resulting graph S would not be a unit 1-interval graph, contradicting the hypothesis. ◁

▷ Claim 16. (⋆) Let (S, f) be an arbitrary split in $\mathcal{S}_{\mathcal{U}}(G_\Psi)$. Then, for every 2-clause, at least one of the representatives of the literal vertices incident to the clause gadget in S must be an isolated vertex.

The previous claims imply that there is an isolated literal vertex incident to every 3-clause and to every 2-clause. Since literal vertices that have an isolated representative correspond to true literals in the assignment fixed before, it follows that there is a true literal per clause, and thus, all clauses are satisfied. This finishes the proof of the converse direction. ◀

As the problem is clearly in NP, the polynomial-time construction together with Lemmas 12 and 14 conclude the proof of Theorem 2. The bound on the degree follows because the constructed graph G has maximum degree 6 (the positive literal vertices have degree 4 in the variable gadget and are incident to 2 external edges).

3.2 Hardness of Unit 2-Interval Recognition

We show next that Colored Unit 2-Interval Recognition is polynomial-time reducible to Unit 2-Interval Recognition, which yields the main result of the paper:

▶ **Theorem 17.** *Unit 2-Interval Recognition is* NP*-complete, even for graphs of degree at most 7.*

Proof. We reduce from Colored Unit 2-Interval Recognition, which is NP-hard by Theorem 2. Given any instance (G, γ) of Colored Unit 2-Interval Recognition, where $G = (V, E)$ is a graph and $\gamma : V \to \{\texttt{white}, \texttt{black}\}$ is a vertex-coloring map, we construct an equivalent instance $G' = (V', E')$ of Unit 2-Interval Recognition. Define $n = |V|$ and $V_{\texttt{c}} = \{u \mid u \in V \ \wedge \ \gamma(u) = \texttt{c}\}$ for $c \in \{\texttt{white}, \texttt{black}\}$ (so that $n = |V_{\texttt{white}}| + |V_{\texttt{black}}|$).

We obtain $G' = (V', E')$ from G by replacing every vertex $v \in V_{\texttt{black}}$ by the gadget B_v depicted in Figure 10, which we also call black vertex gadget. Formally, for every $v \in V_{\texttt{black}}$, we add the vertices $V_v = \{a_v^i, b_v^i \mid 0 \leqslant i \leqslant 3\}$ and the edges $E_v = \{(v, a_v^0), (a_v^0, a_v^i), (v, b_v^0), (b_v^0, b_v^i), (a_v^0, b_v^0) \mid 1 \leqslant i \leqslant 3\}$. The gadget B_v is exactly the graph induced by the union of V_v and vertex v. Note that the vertex v of B_v is *public*, that is, it is adjacent to vertices of B_v and to vertices outside of B_v, while the rest of the vertices of B_v are *private*, i.e., they are only adjacent to vertices of B_v.

We have thus constructed a graph G' with vertex set $V' = V \cup \{V_v \mid v \in V_{\texttt{black}}\}$ and edge set $E' = E \cup \{E_v \mid v \in V_{\texttt{black}}\}$. Note that G' contains G as an induced subgraph, as $G'[V] = G$. Combining this with the replacement of every vertex in $V_{\texttt{black}}$ by a gadget with 9 vertices and 9 edges, it follows that $|V'| = |V_{\texttt{white}}| + 9\,|V_{\texttt{black}}|$ and $|E'| = |E| + 9\,|V_{\texttt{black}}|$.

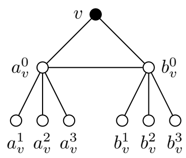

Figure 10 Gadget B_v used to replace every black vertex v of G in the construction of G'. Vertex v is a *public* vertex, as it is adjacent to vertices of the gadget (a_v^0 and b_v^0) and vertices outside the gadget (namely, its neighbors in the original graph G), whereas the rest of the vertices are *private*, as their only neighbors are vertices from the gadget (the ones shown in the figure).

The purpose of the black vertex gadget B_v used to replace every $v \in V_{\texttt{black}}$ in the construction of G' is to restrict the unit 2-interval representations of G'. Indeed, we will see that it forces one of the intervals associated to v to be used exclusively to represent the gadget, while the other interval is used exclusively to represent the rest of the neighborhood of v (which is exactly its neighborhood in the original graph G). Figure 11 shows a unit 2-interval representation $\mathbf{R} = \{(I_1(x), I_2(x)) \mid x \in V_v \cup \{v\}\}$ of B_v such that $I_1(v)$ does not have any points in common with the rest of the intervals of $\mathbf{R}$ (i.e., only $I_2(v)$ is used to represent the gadget). Furthermore, in the given representation, $I_2(v)$ cannot intersect any interval associated to a vertex outside of the gadget, as there is no point of $I_2(v)$ that does not intersect either $I_1(a_v^0)$ or $I_1(b_v^0)$, and both a_v^0 and b_v^0 are private vertices for v. The next claim proves that any unit 2-interval representation of B_v is as in Figure 11, up to symmetry.

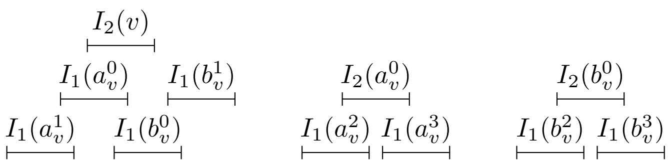

Figure 11 A unit 2-interval representation of B_v (Figure 10), i.e., $\mathbf{D}_{B_v}$ for an arbitrary $v \in V_{\texttt{black}}$. Note that only one interval of v is used ($I_2(v)$), while the other one remains free to display the rest of the neighborhood of v (and is not represented here).

▷ Claim 18. Let $\{(I_1(x), I_2(x)) \mid x \in V_v \cup \{v\}\}$ be a unit 2-interval representation of B_v. Then, there exist some indices $i, j, k \in \{1, 2\}$ such that the representation of $I_i(v), I_j(a_v^0), I_k(b_v^0)$ is contiguous (i.e., the union of the three intervals is an interval) and $I_i(v)$ is properly contained in the union $I_j(a_v^0) \cup I_k(b_v^0)$.

Proof. In the following, we denote an interval associated to a vertex by the name of the vertex if it refers to an arbitrary interval from the corresponding 2-interval (i.e., we will write v to denote $I_1(v)$ or $I_2(v)$ when the choice of interval is irrelevant).

Since a_v^0 and b_v^0 are both centers of an induced $K_{1,4}$, one of the intervals associated to a_v^0, say $I_1(a_v^0)$, needs to intersect v, b_v^0 and one of the a_v^i for some $i \in \{1,2,3\}$, say a_v^1 without loss of generality (because of the symmetry). Furthermore, the intervals of v and b_v^0 that intersect $I_1(a_v^0)$ also need to intersect each other, as otherwise $I_1(a_v^0)$ would intersect three disjoint intervals, contradicting the fact that the representation is unit. On the other hand, $I_2(a_v^0)$ has to intersect the two remaining a_v^i, that is, a_v^2 and a_v^3. Similarly, one of the intervals associated to b_v^0, say $I_1(b_v^0)$, needs to intersect v and a_v^0 (which also intersect each other), and one of the b_v^i for some $i \in \{1,2,3\}$, whereas $I_2(b_v^0)$ intersects the two remaining b_v^i. Again, without loss of generality, we can assume that $I(b_v^0)$ intersects b_v^1 while $I_2(b_v^0)$ intersects b_v^2 and b_v^3.

Thus, we have that $I_1(a_v^0)$ intersects v and $I_1(b_v^0)$ (which also intersect each other), and a_v^1; while $I(b_v^0)$ intersects v and $I_1(a_v^0)$ (which also intersect each other), and b_v^1. This implies that the representation of a_v^1, $I_1(a_v^0)$, b_v^1, $I_1(b_v^0)$ has to be contiguous. Finally, since vertex v is not adjacent to either a_v^1 nor b_v^1, the only possibility to represent the edges (v, a_v^0) and (v, b_v^0) is by placing an interval associated to v, say $I_2(v)$, properly contained in the union $I_1(a_v^0) \cup I_1(b_v^0)$, as in Figure 11. ◁

The next two claims now prove the correctness of the reduction.

▷ Claim 19. If G is a colored unit 2-interval graph, then G' is a unit 2-interval graph.

Proof. Suppose that G is a colored unit 2-interval graph. Then, by assumption, there exists a collection of unit 2-intervals $\mathbf{D}_{\texttt{white}} = \{(I_1(v), I_2(v)) \mid v \in V_{\texttt{white}}\}$ and a collection of unit intervals $\mathbf{I}_{\texttt{black}} = \{I_1(v) \mid v \in V_{\texttt{black}}\}$ such that $G \simeq \Omega\left(\mathbf{D}_{\texttt{white}} \cup \mathbf{I}_{\texttt{black}}\right)$.

From $\mathbf{D} = (\mathbf{D}_{\texttt{white}} \cup \mathbf{I}_{\texttt{black}})$, we show how to construct a unit 2-interval representation $\mathbf{D}'$ of G'. Recall that $(V_{\texttt{white}} \cup V_{\texttt{black}}) = V \subset V'$. Similarly, we will construct $\mathbf{D}'$ such that $\mathbf{D} \subset \mathbf{D}'$. In fact, we will have that $\mathbf{D}' = \mathbf{D} \cup \left(\bigcup_{v \in V_{\texttt{black}}} \mathbf{D}_{B_v}\right)$, where for every $v \in V_{\texttt{black}}$, $\mathbf{D}_{B_v}$ is the interval representation of the gadget B_v. More precisely, we construct $\mathbf{D}'$ as follows:

- For every $v \in V_{\texttt{white}}$, we add to $\mathbf{D}'$ the 2-interval $(I_1(v), I_2(v))$ from $\mathbf{D}$.
- For every $v \in V_{\texttt{black}}$, we add to $\mathbf{D}'$ the interval $I_1(v)$ from $\mathbf{D}$ together with $\mathbf{D}_{B_v}$, i.e., the interval $I_2(v)$ plus the 2-intervals $(I_1(a_v^k), I_2(a_v^k))$ and $(I_1(b_v^k), I_2(b_v^k))$ for $0 \leqslant k \leqslant 3$ as defined in Figure 11.

By construction, $\mathbf{D}'$ is a collection of unit 2-intervals. It is now a simple matter to verify that $G' \simeq \Omega(\mathbf{D}')$. ◁

▷ Claim 20. ($\star$) If G' is a unit 2-interval graph, then G is a colored unit 2-interval graph.

As the problem is clearly in NP, combining the fact that the construction of G' can be carried out in polynomial time with Claims 19 and 20, we obtain that Unit 2-Interval Recognition is NP-complete. The bound on the degree given in the statement of the theorem follows by construction, from adding the black vertex gadgets (Figure 10) to the graph constructed in the proof of Theorem 2 (Figure 2). Indeed, this results in a graph of maximum degree 7, as C_i is adjacent to 5 vertices in the variable gadget and to 2 vertices from the black vertex gadget. ◀

3.3 Consequences and generalizations

We now generalize the result for unit d-interval graphs, with $d \geqslant 2$, which is not directly implied in graph recognition problems, and for some specific cases of unit d-intervals.

▶ **Corollary 21.** ($\star$) *Recognizing unit d-interval graphs is* NP*-complete for every $d \geqslant 2$.*

▶ **Corollary 22.** ($\star$) *Recognizing* $(x, \ldots, x)$ d*-interval graphs is* NP*-complete for every* $x \geqslant 11$ *and every* $d \geqslant 2$*.*

▶ **Corollary 23.** ($\star$) *Recognizing depth* r *unit* d*-interval graphs is* NP*-complete for every* $r \geqslant 4$ *and every* $d \geqslant 2$*.*

The following corollary is based on the Exponential Time Hypothesis (ETH). More details on this notion that we are only touching here can be found in [9, Chapter 14].

▶ **Corollary 24.** ($\star$) *Unless the ETH fails,* UNIT d-INTERVAL RECOGNITION *does not admit an algorithm with running time* $2^{o(|V|+|E|)}$*.*

4 Concluding remarks

We have proven that recognizing unit d-interval graphs is NP-complete for any $d \geqslant 2$. Furthermore, our reduction implies that recognizing $(x, \ldots, x)$ d-interval graphs for any $x \geqslant 11$, and depth r unit d-interval graphs for any $r \geqslant 4$, is also hard. These results represent a significant step towards settling the landscape of the complexity of the recognition of the different subclasses of d-interval graphs.

However, some questions still remain open. Since we have shown that recognizing depth 4 unit d-interval graphs is NP-complete and it is known that the recognition of depth 2 unit d-interval graphs is polynomial-time solvable [18], it still remains to delineate the exact boundary, i.e., study the case of depth 3 unit d-interval graphs. On the other hand, the complexity of recognizing $(x, \ldots, x)$ d-interval graphs for $x < 11$ is also unknown. Finally, we have obtained a lower bound for the running time of an algorithm for recognizing unit 2-intervals. Since the brute-force algorithm, running in $\mathcal{O}(2^{n^2})$, is far from achieving it, it would be interesting to reduce this gap.

References

1 Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. *J. ACM*, 48(5):1069–1090, 2001. doi:10.1145/502102.502107.

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. doi:10.1137/S0097539703437843.

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), 2010. doi:10.1145/1721837.1721856.

6 Joel E. Cohen. *Food Webs and Niche Space*, volume 11 of *Monographs in Population Biology*. Princeton University Press, 1978.

7 Derek G. Corneil, Stephan Olariu, and Lorna Stewart. The LBFS structure and recognition of interval graphs. *SIAM J. Discret. Math.*, 23(4):1905–1953, 2010. doi:10.1137/S0895480100373455.

8 Margaret B. Cozzens. *Higher and Multi-Dimensional Analogues of Interval Graphs*. PhD thesis, Rutgers University, 1982.

9 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

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 Michael R. Fellows, Jan Kratochvíl, Matthias Middendorf, and Frank Pfeiffer. The complexity of induced minors and related problems. *Algorithmica*, 13(3):266–282, 1995. doi:10.1007/BF01190507.

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

13 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.

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

15 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.

16 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.

17 András Gyárfás and Douglas West. Multitrack interval graphs. *Congressus Numerantium*, pages 109–116, 1995.

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 Deborah Joseph, Joao Meidanis, and Prasoon Tiwari. Determining DNA sequence similarity using maximum independent set algorithms for interval graphs. In Otto Nurmi and Esko Ukkonen, editors, *Algorithm Theory - SWAT '92, Third Scandinavian Workshop on Algorithm Theory, Helsinki, Finland, July 8-10, 1992, Proceedings*, volume 621 of *LNCS*, pages 326–337. Springer, 1992. doi:10.1007/3-540-55706-7_29.

20 Jan Kratochvíl. A special planar satisfiability problem and a consequence of its NP-completeness. *Discret. Appl. Math.*, 52(3):233–252, 1994. doi:10.1016/0166-218X(94)90143-0.

21 Cornelis Gerrit Lekkerkerker and Johan Ch. Boland. Representation of a finite graph by a set of intervals on the real line. *Fundamenta Mathematicae*, 51:45–64, 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 Dieter Rautenbach and Jayme L Szwarcfiter. Unit interval graphs of open and closed intervals. *J. Graph Theory*, 72(4):418–429, 2013. doi:10.1002/jgt.21650.

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

26 Fred S. Roberts. On the boxicity and cubicity of a graph. In W. T. Tutte, editor, *Recent Progress in Combinatorics*, pages 301–310. Academic Press, NY, 1969.

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

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. *Disrecte Appl. Math.*, 8:295–305, 1984. doi:10.1016/0166-218X(84)90127-6.

32 Mihalis Yannakakis. The complexity of the partial order dimension problem. *SIAM Journal on Algebraic Discrete Methods*, 3(3):351–358, 1982.

33 Peisen Zhang, Eric A Schon, Stuart G Fischer, Eftihia Cayanis, Janie Weiss, Susan Kistler, and Philip E Bourne. An algorithm based on graph theory for the assembly of contigs in physical mapping of DNA. *Comput. Appl. Biosci.*, 10(3):309–317, 1994. doi:10.1093/bioinformatics/10.3.309.