Unit Interval Editing is Fixed-Parameter Tractable

Yixin Cao*

Abstract

Given a graph G and integers k_1, k_2, and k_3, the unit interval editing problem asks whether G can be transformed into a unit interval graph by at most k_1 vertex deletions, k_2 edge deletions, and k_3 edge additions. We give an algorithm solving this problem in time $2^{O(k \log k)} \cdot (n+m)$, where $k := k_1 + k_2 + k_3$, and n, m denote respectively the numbers of vertices and edges of G. Therefore, it is fixed-parameter tractable parameterized by the total number of allowed operations.

Our algorithm implies the fixed-parameter tractability of the unit interval edge deletion problem, for which we also present a more efficient algorithm running in time $O(4^k \cdot (n+m))$. Another result is an $O(6^k \cdot (n+m))$-time algorithm for the unit interval vertex deletion problem, significantly improving the algorithm of van 't Hof and Villanger, which runs in time $O(6^k \cdot n^6)$.

1 Introduction

A graph is a *unit interval graph* if its vertices can be assigned to unit-length intervals on the real line such that there is an edge between two vertices if and only if their corresponding intervals intersect. Most important applications of unit interval graphs were found in computational biology [5, 16, 18], where data are mainly obtained by unreliable experimental methods. Therefore, the graph representing the raw data is very unlikely to be a unit interval graph, and an important step before understanding the data is to find and fix the hidden errors. For this purpose various graph modification problems have been formulated: Given a graph G on n vertices and m edges, is there a set of at most k modifications that make G a unit interval graph. In particular, edge additions, also called completion, and edge deletions are used to fix false negatives and false positives respectively, while vertex deletions can be viewed as the elimination of outliers. We have thus three variants, which are all known to be NP-complete [21, 30, 16].

These modification problems to unit interval graphs have been well studied in the framework of parameterized computation, where the parameter is usually the number of modifications. Recall that a graph problem, with a nonnegative parameter k, is *fixed-parameter tractable (FPT)* if there is an algorithm solving it in time $f(k) \cdot (n+m)^{O(1)}$, where f is a computable function depending only on k [12]. The problems unit interval completion and unit interval vertex deletion have been shown to be FPT by Kaplan et al. [18] and van Bevern et al. [3] respectively. In contrast, however, the parameterized complexity of the edge deletion version remained open to date, which we settle here. Indeed, we devise single-exponential linear-time parameterized algorithms for both deletion versions.

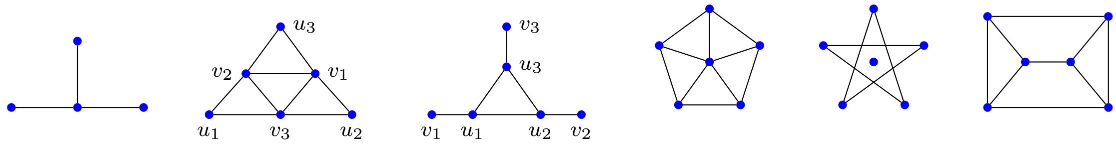
Theorem 1.1. *The problems unit interval vertex deletion and unit interval edge deletion can be solved in time* $O(6^k \cdot (n+m))$ *and* $O(4^k \cdot (n+m))$ *respectively.*

Our algorithm for unit interval vertex deletion significantly improves the currently best parameterized algorithm for it, which takes $O(6^k \cdot n^6)$ time [17]. Another algorithmic result of van 't Hof and Villanger [17] is an $O(n^7)$-time 6-approximation algorithm for the problem, which we improve to the following.

Theorem 1.2. *There is an* $O(nm + n^2)$*-time approximation algorithm of approximation ratio* 6 *for the minimization version of the unit interval vertex deletion problem.*

The structures and recognition of unit interval graphs have been well studied and well understood [11]. It is known that a graph is a unit interval graph if and only if it contains no claw, S_3, $\overline{S_3}$, (as depicted in Fig. 1,) or any hole (i.e., an induced cycle on at least four vertices) [26, 29]. Unit interval graphs are thus a subclass of chordal graphs, which are those graphs containing no holes. Modification problems

*Department of Computing, Hong Kong Polytechnic University, Hong Kong, China. yixin.cao@polyu.edu.hk. Supported in part by the Hong Kong Research Grants Council (RGC) under grant 252026/15E, the National Natural Science Foundation of China (NSFC) under grants 61572414 and 61420106009, the Hong Kong Polytechnic University (PolyU) under grant 4-ZZEZ, and the European Research Council (ERC) under grant 280152.



(a) claw (b) S_3 (c) $\overline{S_3}$ (d) W_5 (e) $C_5^* = \overline{W_5}$ (f) $\overline{C_6}$

Figure 1: Small forbidden induced graphs.

to chordal graphs and unit interval graphs are among the earliest studied problems in parameterized computation, and their study had been closely related. For example, the algorithm of Kaplan et al. [18] for unit interval completion is a natural spin-off of their algorithm for chordal completion, or more specifically, the combinatorial result of all minimal ways to fill holes in. A better analysis was shortly done by Cai [7], who also made explicit the use of bounded-depth search in disposing of finite forbidden induced subgraphs. This observation and the parameterized algorithm of Marx [24] for the chordal vertex deletion problem immediately imply the fixed-parameter tractability of the unit interval vertex deletion problem: One may break first all induced claws, S_3's, and $\overline{S_3}$'s, and then call Marx's algorithm. Here we are using the hereditary property of unit interval graphs,—recall that a graph class is *hereditary* if it is closed under taking induced subgraphs. However, neither approach can be adapted to the edge deletion version in a simple way. Compared to completion that needs to add $\Omega(\ell)$ edges to fill a C_ℓ (i.e., a hole of length ℓ) in, an arbitrarily large hole can be fixed by a single edge deletion. On the other hand, the deletion of vertices leaves an induced subgraph, which allows us to focus on holes once all claws, S_3's, and $\overline{S_3}$'s have been eliminated; however, the deletion of edges to fix holes of a {claw, S_3, $\overline{S_3}$}-free graph may introduce new claws, S_3's, and/or $\overline{S_3}$'s. Therefore, although a parameterized algorithm for the chordal edge deletion problem has also been presented by Marx [24], there is no obvious way to use it to solve the unit interval edge deletion problem.

Direct algorithms for unit interval vertex deletion were later discovered by van Bevern et al. [3] and van 't Hof and Villanger [17], both using a two-phase approach. The first phase of their algorithms breaks all forbidden induced subgraphs on at most six vertices. Note that this differentiates from the aforementioned simple approach in that it breaks not only claws, S_3's, and $\overline{S_3}$'s, but all C_ℓ's with $\ell \leqslant 6$. Although this phase is conceptually intuitive, it is rather nontrivial to efficiently carry it out, and the simple brute-force way introduces an n^6 factor to the running time. Their approaches diverse completely in the second phase. Van Bevern et al. [3] used a complicated iterative compression procedure that has a very high time complexity, while van 't Hof and Villanger [17] showed that after the first phase, the problem is linear-time solvable. The main observation of van 't Hof and Villanger [17] is that a connected {claw, S_3, $\overline{S_3}$, C_4, C_5, C_6}-free graph is a proper circular-arc graph, whose definition is postponed to Section 2. In the conference presentation where Villanger first announced the result, it was claimed that this settles the edge deletion version as well. However, the claimed result has not been materialized: It appears neither in the conference version [28] (which has a single author) nor in the significantly revised and extended journal version [17]. Unfortunately, this unsubstantiated claim did get circulated.

Although the algorithm of van 't Hof and Villanger [17] is nice and simple, its self-contained proof is excruciatingly complex. We revisit the relation between unit interval graphs and some subclasses of proper circular-arc graphs, and study it in a structured way. In particular, we observe that unit interval graphs are precisely those graphs that are both chordal graphs and proper Helly circular-arc graphs. As a matter of fact, unit interval graphs can also be viewed as "unit Helly interval graphs" or "proper Helly interval graphs," thereby making a natural subclass of proper Helly circular-arc graphs. The full containment relations are summarized in Fig. 2; the reader unfamiliar with some graph classes in this figure may turn to the appendix for a brief overview. These observations inspire us to show that a connected {claw, S_3, $\overline{S_3}$, C_4, C_5}-free graph is a proper Helly circular-arc graph. It is easy to adapt the linear-time certifying recognition algorithms for proper Helly circular-arc graphs [22, 9] to detect an induced claw, S_3, $\overline{S_3}$, C_4, or C_5 if one exists. Once all of them have been completely eliminated and the graph is a proper Helly circular-arc graph, it is easy to solve the unit interval vertex deletion problem in linear time. Likewise, using the structural properties of proper Helly circular-arc graphs, we can derive a linear-time algorithm for unit interval edge deletion on them. It is then straightforward to use simple branching to develop the parameterized algorithms stated in Theorem 1.1, though some nontrivial analysis is required to obtain the time bound for unit interval edge deletion.

Van Bevern et al. [3] showed that the unit interval vertex deletion problem remains NP-hard on

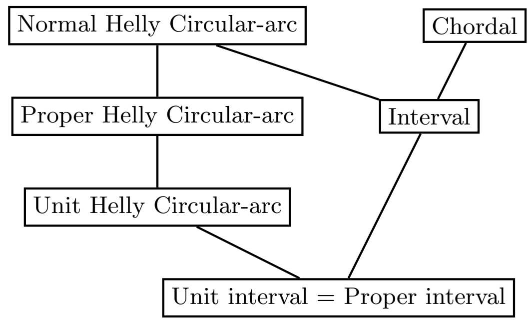


Figure 2: Containment relation of related graph classes.
Normal Helly circular-arc ∩ Chordal = Interval.
Proper Helly circular-arc ∩ Chordal = Unit Helly circular-arc ∩ Chordal = Unit interval.

{claw, $S_3, \overline{S_3}$}-free graphs. After deriving a polynomial-time algorithm for the problem on {claw, $S_3, \overline{S_3}, C_4, C_5, C_6$}-free graphs, van 't Hof and Villanger [17] asked for its complexity on {claw, $S_3, \overline{S_3}, C_4$}-free graphs. (It is somewhat intriguing that they did not mention the {claw, $S_3, \overline{S_3}, C_4, C_5$}-free graphs.) Note that a {claw, $S_3, \overline{S_3}, C_4$}-free graph is not necessarily a proper (Helly) circular-arc graph, evidenced by the W_5 (Fig. 1d). We answer this question by characterizing connected {claw, $S_3, \overline{S_3}, C_4$}-free graphs that are not proper Helly circular-arc graphs. We show that such a graph must be *like* a W_5: If we keep only one vertex from each twin class (all vertices in a twin class have the same closed neighborhood) of the original graph, then we obtain a W_5. It is then routine to solve the problem in linear time.

Theorem 1.3. *The problems unit interval vertex deletion and unit interval edge deletion can be solved in* $O(n+m)$ *time on {claw,* $S_3, \overline{S_3}$*,* C_4*}-free graphs.*

We remark that the techniques we developed in previous work [8] can also be used to derive Theorems 1.1 and 1.2. Those techniques, designed for interval graphs, are nevertheless far more complicated than necessary when applied to unit interval graphs. The approach we used in the current work, i.e., based on structural properties of proper Helly circular-arc graphs, is tailored for unit interval graphs, hence simpler and more natural. Another benefit of this approach is that it enables us to devise a parameterized algorithm for the general modification problem to unit interval graphs, which allows all three types of operations. This formulation generalizes all the three single-type modifications, and is also natural from the viewpoint of the aforementioned applications for de-noising data, where different types of errors are commonly found coexisting. Indeed, the assumption that the input data contain only a single type of errors is somewhat counterintuitive. Formally, given a graph G, the *unit interval editing* problem asks whether there are a set V_- of at most k_1 vertices, a set E_- of at most k_2 edges, and a set E_+ of at most k_3 non-edges, such that the deletion of V_- and E_- and the addition of E_+ make G a unit interval graph. We show that it is FPT, parameterized by the total number of allowed operations, $k := k_1 + k_2 + k_3$.

Theorem 1.4. *The unit interval editing problem can be solved in time* $2^{O(k \log k)} \cdot (n + m)$.

By and large, our algorithm for unit interval editing again uses the two-phase approach. However, we are not able to show that it can be solved in polynomial time on proper Helly circular-arc graphs. Therefore, in the first phase, we use brute force to remove not only claws, S_3's, $\overline{S_3}$'s, and C_4's, also all holes of length at most $k_3 + 3$. The high exponential factor in the running time is due to purely this phase. After that, every hole has length at least $k_3 + 4$, and has to be fixed by deleting a vertex or edge. We manage to show that an inclusion-wise minimal solution of this reduced graph does not add edges, and the problem can then be solved in linear time.

The study of general modification problems was initiated by Cai [7], who observed that the problem is FPT if the objective graph class has a finite number of minimal forbidden induced subgraphs. More challenging is thus to devise parameterized algorithms for those graph classes whose minimal forbidden induced subgraphs are infinite. Prior to this paper, the only known nontrivial graph class on which the general modification problem is FPT is the chordal graphs [10]. Theorem 1.4 extends this territory by including another well-studied graph class. As a corollary, Theorem 1.4 implies the fixed-parameter tractability of the unit interval edge editing problem, which allows both edge operations but not vertex deletions [6]. To see this we can simply try every combination of k_2 and k_3 as long as $k_2 + k_3$ does not exceed the given bound.

Organization. The rest of the paper is organized as follows. Section 2 presents combinatorial and algorithmic results on $\{\text{claw}, S_3, \overline{S_3}, C_4\}$-free graphs. Sections 3 and 4 present the algorithms for unit interval vertex deletion and unit interval edge deletion respectively (Theorems 1.1–1.3). Section 5 extends them to solve the general editing problem (Theorem 1.4). Section 6 closes this paper by discussing some possible improvement and new directions. The appendix provides a brief overview of related graph classes as well as their characterizations by forbidden induced subgraphs.

2 $\{$Claw, $S_3, \overline{S_3}$, $C_4\}$-free graphs

All graphs discussed in this paper are undirected and simple. A graph G is given by its vertex set $V(G)$ and edge set $E(G)$, whose cardinalities will be denoted by n and m respectively. All input graphs in this paper are assumed to be nontrivial ($n > 1$) and connected, hence $n = O(m)$. For $\ell \geqslant 4$, we use C_ℓ to denote a hole on ℓ vertices; if we add a new vertex to a C_ℓ and make it adjacent to no or all vertices in the hole, then we end with a C_ℓ^* or W_ℓ, respectively. For a hole H, we have $|V(H)| = |E(H)|$, denoted by $|H|$. The complement graph $\overline{G}$ of a graph G is defined on the same vertex set $V(G)$, where a pair of vertices u and v is adjacent if and only if $uv \notin E(G)$; e.g., $\overline{W_5} = C_5^*$, and depicted in Fig. 1f is the complement of C_6.

An *interval graph* is the intersection graph of a set of intervals on the real line. A natural way to extend interval graphs is to use arcs and a circle in the place of intervals and the real line, and the intersection graph of arcs on a circle is a *circular-arc graph*. The set of intervals or arcs is called an *interval model* or *arc model* respectively, and it can be specified by their $2n$ endpoints. In this paper, all intervals and arcs are closed, and no distinct intervals or arcs are allowed to share an endpoint in the same model; these restrictions do not sacrifice any generality. In a *unit interval model* or a *unit arc model*, every interval or arc has length one. An interval or arc model is *proper* if no interval or arc in it properly contains another interval or arc. A graph is a *unit interval graph, proper interval graph, unit circular-arc graph*, or *proper circular-arc graph* if it has a unit interval model, proper interval model, unit arc model, or proper arc model, respectively. The forbidden induced subgraphs of unit interval graphs have been long known.

Theorem 2.1 ([29])**.** *A graph is a unit interval graph if and only if it contains no claw, $S_3, \overline{S_3}$, or any hole.*

Clearly, any (unit/proper) interval model can be viewed as a (unit/proper) arc model leaving some point uncovered, and hence all (unit/proper) interval graphs are always (unit/proper) circular-arc graphs. A unit interval/arc model is necessarily proper, but the other way does not hold true in general. A well-known result states that a proper interval model can always be made unit, and thus these two graph classes coincide [26, 29].[1] This fact will be heavily used in the present paper; e.g., most of our proofs consist in modifying a proper arc model into a proper interval model, which represents the desired unit interval graph. On the other hand, it is easy to check that the S_3 is a proper circular-arc graph but not a unit circular-arc graph. Therefore, the class of unit circular-arc graphs is a proper subclass of proper circular-arc graphs.

An arc model is *Helly* if every set of pairwise intersecting arcs has a common intersection. A circular-arc graph is *proper Helly* if it has an arc model that is both proper and Helly.

Theorem 2.2 ([27, 22, 9])**.** *A graph is a proper Helly circular-arc graph if and only if it contains no claw, $S_3, \overline{S_3}$, W_4, W_5, $\overline{C_6}$, or C_ℓ^* for $\ell \geqslant 4$.*

The following is immediate from Theorems 2.1 and 2.2.

Corollary 2.3. *If a proper Helly circular-arc graph is chordal, then it is a unit interval graph.*

From Theorems 2.1 and 2.2 one can also derive the following combinatorial result. But since we will prove a stronger result in Theorem 2.5 that implies it, we omit its proof here.

Proposition 2.4. *Every connected $\{$claw, $S_3, \overline{S_3}$, C_4, $C_5\}$-free graph is a proper Helly circular-arc graph.*

Note that in Proposition 2.4, as well as most combinatorial statements to follow, we need the graph to be connected. Circular-arc graphs are not closed under taking disjoint unions. If a (proper Helly) circular-arc graph is not chordal, then it is necessarily connected. In other words, a disconnected (proper) circular-arc graph must be a (unit) interval graph.

[1] The reason we choose "unit" over "proper" in the title of this paper is twofold. On the one hand, the applications we are interested in are more naturally represented by unit intervals. On the other hand, we want to avoid the use of "proper interval subgraphs," which is ambiguous.

Proposition 2.4 can be turned into an algorithmic statement. We say that a recognition algorithm (for a graph class) is *certifying* if it provides a minimal forbidden induced subgraph of the input graph G when G is determined to be not in this class. Linear-time certifying algorithms for recognizing proper Helly circular-arc graphs have been reported by Lin et al. [22] and Cao et al. [9], from which one can derive a linear-time algorithm for detecting an induced claw, $\overline{S_3}$, S_3, C_4, or C_5 from a graph that is not a proper Helly circular-arc graph.

This would suffice for us to develop most of our main results. Even so, we would take pain to prove slightly stronger results (than Proposition 2.4) on $\mathcal{F}$-free graphs, where $\mathcal{F}$ denotes the set {claw, $S_3, \overline{S_3}$, C_4}. The purpose is threefold. First, they enable us to answer the question asked by van 't Hof and Villanger [17], i.e., the complexity of unit interval vertex deletion on $\mathcal{F}$-free graphs, thereby more accurately delimiting the complexity border of the problem. Second, as we will see, the disposal of C_5's would otherwise dominate the second phase of our algorithm for unit interval edge deletion, so excluding them enables us to obtain better exponential dependency on k in the running time. Third, the combinatorial characterization might be of its own interest.

A *(true) twin class* of a graph G is an inclusion-wise maximal set of vertices in which all have the same closed neighborhood. A graph is called a *fat* W_5 if it has precisely six twin classes and it becomes a W_5 after we remove all but one vertices from each twin class.[2] By definition, vertices in each twin class induce a clique. The five cliques corresponding to hole in the W_5 is the *fat hole*, and the other clique is the *hub*, of this fat W_5.

Theorem 2.5. *Let G be a connected graph.*

(1) If G is $\mathcal{F}$-free, then it is either a fat W_5 or a proper Helly circular-arc graph.

(2) In $O(m)$ time we can either detect an induced subgraph of G in $\mathcal{F}$, partition $V(G)$ into six cliques constituting a fat W_5, or build a proper and Helly arc model for G.

Proof. We prove only assertion (2) using the algorithm described in Fig. 3, and its correctness implies assertion (1). The algorithm starts by calling the certifying algorithm of Cao et al. [9] for recognizing proper Helly circular-arc graphs (step 0). It enters one of steps 1–4, or 6 based on the outcome of step 0. Here the subscripts of vertices in a hole C_ℓ should be understood to be modulo ℓ.

If the condition of any of steps 1–4 is satisfied, then either a proper and Helly arc model or a subgraph in $\mathcal{F}$ is returned. The correctness of steps 1–3 is straightforward. Step 4.1 can find the path because G is connected; possibly $v = x$, which is irrelevant in steps 4.2–4.4. Note also that $\ell > 4$ in step 4.

By Theorem 2.2, the algorithm passes steps 1–4 only when the outcome of step 0 is a W_5; let H be its hole and let v be the other vertex. Steps 5–7 either detect an induced subgraph of G in $\mathcal{F}$ or partition $V(G)$ into six cliques constituting a fat W_5. Step 6 scans vertices not in the W_5 one by one, and proceeds based on the adjacency between x and H. In step 6.1, H and x make a C_5^*, which means that we can proceed exactly the same as step 4. Note that the situation of step 6.4 is satisfied if x is adjacent to four vertices of H. If none of the steps 6.1 to 6.5 applies, then x has precisely three neighbors in H and they have to be consecutive. This is handled by step 6.6.

Steps 0 and 4 take $O(m)$ time. Steps 1, 2, 3, 5, and 7 need only $O(1)$ time. If the condition in step 6.1 is true, then it takes $O(m)$ time but it always terminates the algorithm after applying it. Otherwise, step 6.1 is never called, and the rest of step 6 scans the adjacency list of each vertex once, and hence takes $O(m)$ time in total. Therefore, the total running time of the algorithm is $O(m)$. This concludes the proof. □

Implied by Theorem 2.5, a connected {claw, $S_3, \overline{S_3}$, C_4, W_5}-free graph is a proper Helly circular-arc graph, which in turns implies Proposition 2.4. At this point a natural question appealing to us is the relation between connected {claw, $S_3, \overline{S_3}$, C_4, C_5}-free graphs and unit Helly circular-arc graphs. Recall that the class of unit interval graphs is a subclass of unit Helly circular-arc graphs, on which we have a similar statement as Corollary 2.3, i.e., a unit Helly circular-arc graph that is chordal is a unit interval graph. However, a connected {claw, $S_3, \overline{S_3}$, C_4, C_5}-free graph that is not a unit Helly circular-arc graph can be constructed as follows: Starting from a C_ℓ with $\ell \geqslant 6$, for each edge $h_i h_{i+1}$ in the hole add a new vertex v_i and two new edges $v_i h_i, v_i h_{i+1}$. (This is actually the $CI(\ell, 1)$ graph defined by Tucker [27]; see also [22].) Therefore, Proposition 2.4 and Theorem 2.5 are the best we can expect in this sense.

[2] Interestingly, in the study of proper circular-arc graphs that are chordal, Bang-Jensen and Hell [1] showed that if a connected {claw, $\overline{S_3}$, C_4, C_5}-free graph contains an S_3, then it must be a fat S_3, which is defined analogously as a fat W_5.

Algorithm recognize-$\mathcal{F}$-free(G)
Input: a connected graph G.
Output: a proper and Helly arc model, a subgraph in $\mathcal{F}$, or six cliques making a fat W_5.

0. call the recognition algorithm for proper Helly circular-arc graphs [9];
1. **if** a proper and Helly arc model $\mathcal{A}$ is found **then return** it;
2. **if** a claw, $\overline{S_3}$, or S_3 is found **then return** it;
3. **if** a W_4, C_4^*, or $\overline{C_6}$ is found **then return** a C_4 contained in it;
4. **if** a C_ℓ^* with hole H and isolated vertex v is found **then**
4.1. use breadth-first search to find a shortest path $v \cdots xyh_i$ from v to H;
4.2. **if** y has a single neighbor h_i in H **then return** claw $\{h_i, y, h_{i-1}, h_{i+1}\}$;
4.3. **if** y has only two neighbors on H that are consecutive, say, $\{h_i, h_{i+1}\}$ **then**
return $\overline{S_3}$ $\{x, y, h_{i-1}, h_i, h_{i+1}, h_{i+2}\}$;
4.4. **return** claw $\{y, x, h_j, h_{j'}\}$, where $h_j, h_{j'}$ are two nonadjacent vertices in $N[y] \cap H$;
$\backslash\backslash$ *The outcome of step 0 must be a* W_5*; let it be* H *and* v*. All subscripts of* h_i *and* K_i *are modulo* 5.
5. $K_0 \leftarrow \{h_0\}; K_1 \leftarrow \{h_1\}; K_2 \leftarrow \{h_2\}; K_3 \leftarrow \{h_3\}; K_4 \leftarrow \{h_4\}; K_v \leftarrow \{v\}$;
6. **for each** vertex x not in the W_5 **do**
6.1. **if** x is not adjacent to H **then** similar as step 4 (H and x make a C_5^*);
6.2. **if** x has a single neighbor h_i in H **then return** claw $\{h_i, x, h_{i-1}, h_{i+1}\}$;
6.3. **if** x is only adjacent to h_i, h_{i+1} in H **then**
if $xv \in E(G)$ **then return** claw $\{v, h_{i-1}, h_{i+2}, x\}$;
else return S_3 $\{x, h_i, h_{i+1}, h_{i-1}, v, h_{i+2}\}$;
6.4. **if** x is adjacent to h_{i-1}, h_{i+1} but not h_i **then return** $xh_{i-1}h_ih_{i+1}$ as a C_4;
6.5. **if** x is adjacent to all vertices in H **then**
if $xy \notin E(G)$ for some $y \in K_v$ or K_i **then return** $xh_{i-1}yh_{i+1}$ as a C_4;
else add x to K_v; $\backslash\backslash$ x *is adjacent to all vertices in the six cliques.*
6.6. **else** $\backslash\backslash$ *Hereafter* $|N(x) \cap H| = 3$*; let them be* h_{i-1}, h_i, h_{i+1}.
if $xy \notin E(G)$ for some $y \in K_i$ or K_v **then return** $xh_{i-1}yh_{i+1}$ as a C_4;
if $xy \notin E(G)$ for some $y \in K_{i-1}$ **then return** claw $\{v, y, h_{i+2}, x\}$;
if $xy \notin E(G)$ for some $y \in K_{i+1}$ **then return** claw $\{v, h_{i-2}, y, x\}$;
if $xy \in E(G)$ for some $y \in K_{i-2}$ **then return** $xh_{i+1}h_{i+2}y$ as a C_4;
if $xy \in E(G)$ for some $y \in K_{i+2}$ **then return** $xh_{i-1}h_{i-2}y$ as a C_4;
else add x to K_i; $\backslash\backslash$ $K_v, K_{i-1}, K_i, K_{i+1} \subseteq N(x)$ *and* $K_{i-2}, K_{i+2} \cap N(x) = \emptyset$.
7. **return** the six cliques.

Figure 3: Recognizing $\mathcal{F}$-free graphs.

Note that a C_4 is a proper Helly circular-arc graph. Thus, the algorithm of Theorem 2.5 is not yet a certifying algorithm for recognizing $\mathcal{F}$-free graphs. To detect an induced C_4 from a proper Helly circular-arc graph, we need to exploit its arc model. If a proper Helly circular-arc graph G is not chordal, then the set of arcs for vertices in a hole necessarily covers the circle, and it is minimal. Interestingly, the converse holds true as well—note that this is not true for chordal graphs.

Proposition 2.6. *[22, 9] Let* G *be a proper Helly circular-arc graph. If* G *is not chordal, then at least four arcs are needed to cover the whole circle in any arc model for* G.

Proposition 2.6 forbids among others two arcs from having two-part intersection.[3]

Corollary 2.7. *Let* G *be a proper Helly circular-arc graph that is not chordal and let* $\mathcal{A}$ *be an arc model for* G*. A set of arcs inclusion-wise minimally covers the circle in* $\mathcal{A}$ *if and only if the vertices represented by them induce a hole of* G.

Therefore, to find a shortest hole from a proper Helly circular-arc graph, we may work on an arc model of it, and find a minimum set of arcs covering the circle in the model. This is another important step of our algorithm for the unit interval editing problem. It has the detection of C_4's as a special case, because a C_4, if existent, must be the shortest hole of the graph.

[3] A model having no such intersection is called *normal*; see the appendix for more discussion.

Lemma 2.8. *There is an $O(m)$-time algorithm for finding a shortest hole of a proper Helly circular-arc graph.*

Before proving Lemma 2.8, we need to introduce some notation. In an interval model, the interval $I(v)$ for vertex v is given by $[\mathtt{lp}(v), \mathtt{rp}(v)]$, where $\mathtt{lp}(v)$ and $\mathtt{rp}(v)$ are its *left and right endpoints* respectively. It always holds $\mathtt{lp}(v) < \mathtt{rp}(v)$. In an arc model, the arc $A(v)$ for vertex v is given by $[\mathtt{ccp}(v), \mathtt{cp}(v)]$, where $\mathtt{ccp}(v)$ and $\mathtt{cp}(v)$ are its *counterclockwise and clockwise endpoints* respectively. All points in an arc model are assumed to be nonnegative; in particular, they are between 0 (inclusive) and ℓ (exclusive), where ℓ is the perimeter of the circle. We point out that possibly $\mathtt{ccp}(v) > \mathtt{cp}(v)$; such an arc $A(v)$ necessarily passes through the point 0. Note that rotating all arcs in the model does not change the intersections among them. Thus we can always assume that a particular arc contains or avoids the point 0. We say that an arc model (for an n-vertex circular-arc graph) is *canonical* if the perimeter of the circle is $2n$, and every endpoint is a different integer in $\{0, 1, \ldots, 2n-1\}$. Given an arc model, we can make it canonical in linear time: We sort all $2n$ endpoints by radix sort, and replace them by their indices in the order.

Each point α in an interval model $\mathcal{I}$ or arc model $\mathcal{A}$ defines a clique, denoted by $K_{\mathcal{I}}(\alpha)$ or $K_{\mathcal{A}}(\alpha)$ respectively, which is the set of vertices whose intervals or arcs contain α. There are at most $2n$ distinct cliques defined as such. If the model is Helly, then they include all maximal cliques of this graph [15]. Since the set of endpoints is finite, for any point ρ in an interval or arc model, we can find a small positive value ϵ such that there is no endpoint in $[\rho-\epsilon, \rho) \cup (\rho, \rho+\epsilon]$,—in other words, there is an endpoint in $[\rho-\epsilon, \rho+\epsilon]$ if and only if ρ itself is an endpoint. Note that the value of ϵ should be understood as a function, depending on the interval/arc model as well as the point ρ, instead of a constant.

Let G be a non-chordal graph and let $\mathcal{A}$ be a proper and Helly arc model for G. If $uv \in E(G)$, then exactly one of $\mathtt{ccp}(v)$ and $\mathtt{cp}(v)$ is contained in $A(u)$ (Proposition 2.6). Thus, we can define a "left-right relation" for each pair of intersecting arcs, which can be understood from the viewpoint of an observer placed at the center of the model. We say that arc $A(v)$ intersects arc $A(u)$ from the left when $\mathtt{cp}(v) \in A(u)$, denoted by $v \to u$. Any set of arcs whose union is an arc not covering the circle (the corresponding vertices induce a connected unit interval graph) can be ordered in a unique way such that $v_i \to v_{i+1}$ for all i. From it we can find the leftmost (most counterclockwise) and rightmost (most clockwise) arcs.

For any vertex v, let $h(v)$ denote the length of the shortest holes through v, which is defined to be $+\infty$ if no hole of G contains v. The following is important for the proof of Lemma 2.8.

Lemma 2.9. *Let $\mathcal{A}$ be a proper and Helly arc model for a non-chordal graph G. Let $v_1, v_2, \ldots, v_p$ be a sequence of vertices such that for each $i = 2, \ldots, p$, the arc $A(v_i)$ is the rightmost of all arcs containing $\mathtt{cp}(v_{i-1})$. If v_1 is contained in some hole and $v_i v_1 \notin E(G)$ for all $2 < i \leqslant p$, then there is a hole of length $h(v_1)$ containing $v_1, v_2, \ldots, v_p$ as consecutive vertices on it.*

Proof. Suppose that there is no such a hole, then there exists a smallest number i with $2 \leqslant i \leqslant p$ such that no hole of length $h(v_1)$ contains $v_1, v_2, \ldots, v_i$. By assumption, there is a hole $v_1 \cdots v_{i-1} u_i u_{i+1} \cdots u_{h(v_1)}$ of length $h(v_1)$ with $u_i \neq v_i$; let it be H. In case that $i = 2$, we assume that H is given in the way that $v_1 \to u_2$. By Corollary 2.7, the set of arcs for H cover the circle in $\mathcal{A}$. Since $v_{i-1} \to u_i$ and by the assumption on v_i, the arc $A(v_i)$ covers $[\mathtt{cp}(v_{i-1}), \mathtt{ccp}(u_{i+1})]$ (note that $i < h(v_1)$ as otherwise $v_i \to v_1$). Therefore, the arcs for $V(H) \setminus \{u_i\} \cup \{v_i\}$ cover the circle as well. By Corollary 2.7, a subset of these vertices induces a hole; since v_i is not adjacent to v_1 from the left, this subset has to contain v_1. But a hole containing v_1 cannot be shorter than H, and hence it contains $v_1, \ldots, v_i$, a contradiction. Therefore, such an i does not exist, and there is a hole of length $h(v_1)$ containing $v_1, v_2, \ldots, v_p$.

Since v_j and v_{j+1} are adjacent for all $1 \leqslant j \leqslant p-1$, vertices $v_1, v_2, \ldots, v_p$ have to be consecutive on this hole. □

Let α be any fixed point in a proper and Helly arc model. According to Corollary 2.7, every hole needs to visit some vertex in $K_{\mathcal{A}}(\alpha)$. Therefore, to find a shortest hole in G, it suffices to find a hole of length $\min\{h(x) : x \in K_{\mathcal{A}}(\alpha)\}$.

Proof of Lemma 2.8. The algorithm described in Fig. 4 finds a hole of length $\min\{h(x) : x \in K_{\mathcal{A}}(\mathtt{cp}(v) + 0.5)\}$. Step 1 creates $|K_{\mathcal{A}}(\mathtt{cp}(v) + 0.5)|$ arrays, each starting with a distinct vertex in $K_{\mathcal{A}}(\mathtt{cp}(v) + 0.5)$, and they are ordered such that their (counter)clockwise endpoints are increasing. The main job of this algorithm is done in step 3. During this step, w is the new vertex to be processed, and U is the current array. Each new vertex is added to at most one array, while each array is either dropped or extended. We use $\perp$ as a dummy vertex, which means that no new vertex has been met after the last one has been put into the previous array. Step 3.1 records the last scanned arc. Once the clockwise endpoint of the last vertex z of the current array U is met, w is appended to U (step 3.4); note that $A(w)$ is the most

Algorithm shortest-hole$(G, \mathcal{A})$
Input: a proper and Helly arc model $\mathcal{A}$ for a non-chordal graph G.
Output: a shortest hole of G.

0. make $\mathcal{A}$ canonical where 0 is $\mathtt{ccp}(v)$ for some v;
1. **for** $i = 1, \ldots, \mathtt{cp}(v) - 1$ **do**
1.1. **if** i is $\mathtt{ccp}(x)$ **then** create a new array $\{x\}$;
\\\\ *these arrays are circularly linked so that the next of the last array is the first one.*
2. $w \leftarrow \perp$; $U \leftarrow$ the first array;
3. **for** $i = \mathtt{cp}(v) + 1, \ldots, 2n - 1$ **do**
3.0. $z \leftarrow$ the last vertex of U;
3.1. **if** i is $\mathtt{ccp}(x)$ **then** $w \leftarrow x$; **continue**†;
3.2. **if** $i \neq \mathtt{cp}(z)$ **then continue**;
3.3. **if** $w = \perp$ **then** delete U; $U \leftarrow$ the next array of U;
3.4. **if** $w \neq \perp$ **then** append w to U; $w \leftarrow \perp$; $U \leftarrow$ the next array of U;
4. **for each** U till the last array **do**
4.1. **if** the first and last vertices of U are adjacent **then return** U;
5. **return** $U \cup \{v\}$. \\\\U *is the last array.*

†: i.e., ignore the rest of this iteration of the for-loop and proceed to the next iteration.

Figure 4: Finding a shortest hole in a proper Helly circular-arc graph.

clockwise arc that contains $\mathtt{cp}(z)$. On the other hand, if $w = \perp$, then we drop this array from further consideration (step 3.3). If after step 3, one of the arrays already induces a hole (i.e., the first and last vertices are adjacent), then it is returned in step 4.1. Otherwise, U does not induces a hole, and step 5 returns the hole induced by U and v.

We now verify the correctness of the algorithm. It suffices to show that the length of the found hole is $\min\{h(x) : x \in K_{\mathcal{A}}(\mathtt{cp}(v) + 0.5)\}$. The following hold for each array U:

(1) For any pair of consecutive vertices u, w of U, the arc $A(w)$ is the rightmost of all arcs containing $\mathtt{cp}(u)$.

(2) At the end of the algorithm, if U is not dropped, then $0 < \mathtt{cp}(z) < \mathtt{cp}(v)$, where z is the last vertex of U.

Let $v_1, v_2, \ldots, v_p$ be the vertices in U. From (1) and (2) it can be inferred that for $1 < i < p$, the vertex v_i is adjacent to only v_{i-1}, v_{i+1} in U, while v_1 and v_p may or may not be adjacent. If $v_1 v_p \in E(G)$, then vertices in U induce a hole of G; otherwise $U \cup \{v\}$ induces a hole of G. By Lemma 2.9, this hole has length $h(v_1)$.

Some of the $|K_{\mathcal{A}}(\mathtt{cp}(v) + 0.5)|$ arrays created in step 1 may be dropped in step 3. Note that step 3 processes arrays in a circular order starting from the first one, and each array is either deleted (step 3.3) or extended by adding one vertex (step 3.4).

(3) At any moment of the algorithm, the sizes of any two arrays differ by at most one. In particular, at the end of step 3, if the current array U is not the first, then U, as well as all the succeeding arrays, has one less element than its predecessor(s).

This ensures that the hole returned in step 4.1 or 5 is the shortest among all the holes decided by the remaining arrays after step 3.

It remains to argue that for any array U deleted in step 3.3, the found hole is not longer than $h(x)$, where x is the first vertex of U. All status in the following is referred at the moment before U is deleted (i.e., before step 3.3 that deletes U). Let z be the last vertex of U. Let U' be the array that is immediately preceding U, and let z' be its last vertex. Note that there is no arc with a counterclockwise endpoint between $\mathtt{cp}(z')$ and $\mathtt{cp}(z)$, as otherwise $w \neq \perp$ and U would not be deleted. Therefore, any arc intersecting $A(z)$ from the right also intersects $A(z')$ from the right. By Lemma 2.9, there is a hole H that has length $h(x)$ and contains $U \cup \{z'\}$. We find a hole through U' of the same length as follows. If U is not the first array, then $|U'| = |U| + 1$, and we replace $U \cup \{z'\}$ by U'. Otherwise, $|U'| = |U|$, and we replace $U \cup \{z'\}$ by $\{v\} \cup U'$. It is easy to verify that after this replacement, H remains a hole of the same length.

We now analyze the running time of the algorithm. Each of the $2n$ endpoints is scanned once, and each vertex belongs to at most one array. Using a linked list to store an array, the addition of a new vertex can be implemented in constant time. Using a circularly linked list to organize the arrays, we can find the next array or delete the current one in constant time. With the (endpoints of) all arcs given, the adjacency between any pair of vertices can be checked in constant time, and thus step 4 takes $O(n)$ time. It follows that the algorithm can be implemented in $O(m)$ time. □

3 Vertex deletion

We say that a set V_- of vertices is a *hole cover* of G if $G - V_-$ is chordal. The hole covers of proper Helly circular-arc graphs are characterized by the following lemma.

Lemma 3.1. *Let $\mathcal{A}$ be a proper and Helly arc model for a non-chordal graph G. A set $V_- \subseteq V(G)$ is a hole cover of G if and only if it contains $K_{\mathcal{A}}(\alpha)$ for some point α in $\mathcal{A}$.*

Proof. For any vertex set V_-, the subgraph $G - V_-$ is also a proper Helly circular-arc graph, and the set of arcs $\{A(v) : v \in V(G) \setminus V_-\}$ is a proper and Helly arc model for $G - V_-$. For the "if" direction, we may rotate $\mathcal{A}$ to make $\alpha = 0$, and then setting $I(v) = A(v)$ for each $v \in V(G) \setminus V_-$ gives a proper interval model for $G - V_-$. For the "only if" direction, note that if there is no α with $K_{\mathcal{A}}(\alpha) \subseteq V_-$, then we can find a minimal set X of vertices from $V(G) \setminus V_-$ such that $\bigcup_{v \in X} A(v)$ covers the whole circle in $\mathcal{A}$. According to Corollary 2.7, X induces a hole of G, which remains in $G - V_-$. □

Noting that any "local" part of a proper and Helly arc model "behaves similarly" as an interval model, Lemma 3.1 is an easy extension of the clique separator property of interval graphs [14]. On the other hand, to get a unit interval graph out of a fat W_5, it suffices to delete a smallest clique from the fat hole. Therefore, Theorem 2.5 and Lemma 3.1 imply the following linear-time algorithm.

Corollary 3.2. *The unit interval vertex deletion problem can be solved in $O(m)$ time (1) on proper Helly circular-arc graphs and (2) on $\mathcal{F}$-free graphs.*

We are now ready to prove the main results of this section.

Theorem 3.3. *There are an $O(6^k \cdot m)$-time parameterized algorithm for the unit interval vertex deletion problem and an $O(nm)$-time approximation algorithm of approximation ratio 6 for its minimization version.*

Proof. Let (G, k) be an instance of unit interval vertex deletion; we may assume that G is not a unit interval graph and $k > 0$. The parameterized algorithm calls first Theorem 2.5(2) to decide whether it has an induced subgraph in $\mathcal{F}$, and then based on the outcome, it solves the problem by making recursive calls to itself, or calling the algorithm of Corollary 3.2. If an induced subgraph F in $\mathcal{F}$ is found, it calls itself $|V(F)|$ times, each with a new instance $(G - v, k - 1)$ for some $v \in V(F)$; since we need to delete at least one vertex from $V(F)$, the original instance (G, k) is a yes-instance if and only if at least one of the instances $(G - v, k - 1)$ is a yes-instance. Otherwise, G is $\mathcal{F}$-free and the algorithm calls Corollary 3.2 to solve it. The correctness of the algorithm follows from discussion above and Corollary 3.2. On each subgraph in $\mathcal{F}$, which has at most 6 vertices, at most 6 recursive calls are made, all with parameter value $k - 1$. By Theorem 2.5, each recursive call is made in $O(m)$ time; each call of Corollary 3.2 takes $O(m)$ time. Therefore, the total running time is $O(6^k \cdot m)$.

The approximation algorithm is adapted from the parameterized algorithm as follows. For the subgraph F found by Theorem 2.5, we delete all its vertices. We continue the process until the remaining graph is $\mathcal{F}$-free, and then we call Corollary 3.2 to solve it optimally. Each subgraph in $\mathcal{F}$ has 4 or 6 vertices, and thus at most $n/4$ such subgraphs can be detected and deleted, each taking $O(m)$ time, hence $O(nm)$ in total. Corollary 3.2 takes another $O(m)$ time. The total running time is thus $O(nm) + O(m) = O(nm)$, and the approximation ratio is clearly 6. □

4 Edge deletion

Inspired by Lemma 3.1, one may expect a similarly nice characterization—being "local" to some point in an arc model for G—for a minimal set of edges whose deletion from a proper Helly circular-arc graph G makes it chordal. This is nevertheless not the case; as shown in Fig. 5, they may behave in a very strange or pathological way.

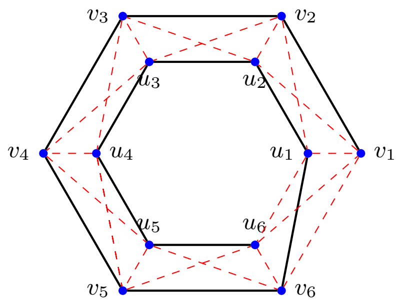


Figure 5: The set of all 30 edges (both solid and dashed) spans a proper Helly circular-arc graph. After the set of 19 dashed edges deleted (we rely on the reader to verify its minimality), the remaining graph on the 11 solid edges is a unit interval graph.
Note that four edges would suffice, e.g., $\{u_2u_3, u_2v_3, v_2u_3, v_2v_3\}$.

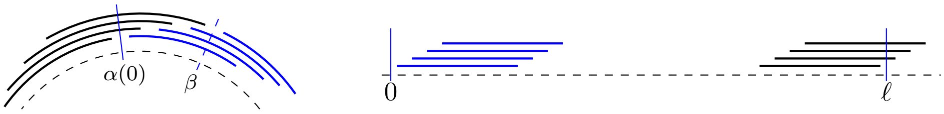
Recall that $v \to u$ means arc $A(v)$ intersecting arc $A(u)$ from the left, or $\mathtt{cp}(v) \in A(u)$. For each point α in a proper and Helly arc model $\mathcal{A}$, we can define the following set of edges:

$$\overrightarrow{E}_{\mathcal{A}}(\alpha) = \{vu : v \in K_{\mathcal{A}}(\alpha),\ u \notin K_{\mathcal{A}}(\alpha), v \to u\}. \tag{1}$$

One may symmetrically view $\overrightarrow{E}_{\mathcal{A}}(\alpha)$ as $\{vu : v \notin K_{\mathcal{A}}(\beta),\ u \in K_{\mathcal{A}}(\beta), v \to u\}$, where $\beta := \max\{\mathtt{cp}(x) : x \in K_{\mathcal{A}}(\alpha)\} + \epsilon$. It is easy to verify that the following gives a proper interval model for $G - \overrightarrow{E}_{\mathcal{A}}(0)$:

$$I(v) := \begin{cases} [\mathtt{ccp}(v), \mathtt{cp}(v) + \ell] & \text{if } v \in K_{\mathcal{A}}(0), \\ [\mathtt{ccp}(v), \mathtt{cp}(v)] & \text{otherwise,} \end{cases} \tag{2}$$

where ℓ is the perimeter of the circle in $\mathcal{A}$; see Fig. 6. For an arbitrary point α, the model $G - \overrightarrow{E}_{\mathcal{A}}(\alpha)$ can be given analogously, e.g., we may rotate the model first to make $\alpha = 0$.



(a) An arc model for G.

(b) The interval model for $G - \overrightarrow{E}_{\mathcal{A}}(\alpha)$ given by (2).

Figure 6: Illustration for Proposition 4.1.

Proposition 4.1. *Let $\mathcal{A}$ be a proper and Helly arc model for a non-chordal graph G. For any point α in $\mathcal{A}$, the subgraph $G - \overrightarrow{E}_{\mathcal{A}}(\alpha)$ is a unit interval graph.*

The other direction is more involved and very challenging. A unit interval graph $\underline{G}$ is called a *spanning unit interval subgraph* of G if $V(\underline{G}) = V(G)$ and $E(\underline{G}) \subseteq E(G)$; it is called *maximum* if it has the largest number of edges among all spanning unit interval subgraphs of G. To prove that all maximum spanning unit interval subgraphs have a certain property, we use the following argument by contradiction. Given a spanning unit interval subgraph $\underline{G}$ not having the property, we locally modify a unit interval model $\mathcal{J}$ for $\underline{G}$ to a *proper* interval model $\mathcal{J}'$ such that the represented graph $\underline{G}'$ satisfies $E(\underline{G}') \subseteq E(G)$ and $|E(\underline{G}')| > |E(\underline{G})|$. Recall that we always select ϵ in a way that there cannot be any endpoint in $[\rho - \epsilon, \rho) \cup (\rho, \rho + \epsilon]$, and thus an arc covering $\rho + \epsilon$ or $\rho - \epsilon$ must contain ρ.

Lemma 4.2. *Let $\mathcal{A}$ be a proper and Helly arc model for a non-chordal graph G. For any maximum spanning unit interval subgraph $\underline{G}$ of G, the deleted edges, $E(G) \setminus E(\underline{G})$, are $\overrightarrow{E}_{\mathcal{A}}(\rho)$ for some point ρ in $\mathcal{A}$.*

Proof. Let $\mathcal{J}$ be a unit interval model for $\underline{G}$, and let $E_- = E(G) \setminus E(\underline{G})$, i.e., the set of deleted edges from G. We find first a vertex v satisfying at least one of the following conditions.

(C1) The sets $N_{\underline{G}}[v]$ and $K_{\mathcal{A}}(\mathtt{ccp}(v) - \epsilon)$ are disjoint and all edges between them are in E_-.

(C2) The sets $N_{\underline{G}}[v]$ and $K_{\mathcal{A}}(\mathtt{cp}(v) + \epsilon)$ are disjoint and all edges between them are in E_-.

Recall that a vertex $u \in K_{\mathcal{A}}(\mathtt{ccp}(v) - \epsilon)$ if and only if $u \to v$, and a vertex $w \in K_{\mathcal{A}}(\mathtt{cp}(v) + \epsilon)$ if and only if $v \to w$; see Fig. 7a. These two conditions imply $N_{\underline{G}}[v] \subseteq K_{\mathcal{A}}(\mathtt{cp}(v))$ and $N_{\underline{G}}[v] \subseteq K_{\mathcal{A}}(\mathtt{ccp}(v))$

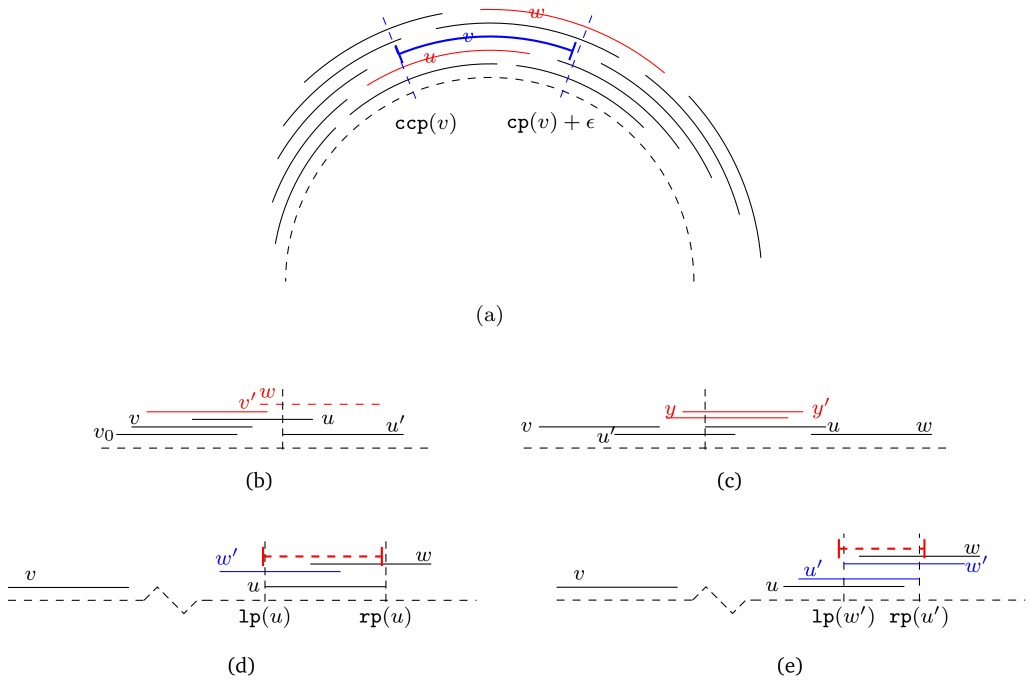


Figure 7: Illustration for the proof of Lemma 4.2.

respectively: All edges between ν itself, which belongs to $N_{\underline{G}}[\nu]$, and $K_{\mathcal{A}}(\mathtt{ccp}(\nu) - \epsilon)$ or $K_{\mathcal{A}}(\mathtt{cp}(\nu) + \epsilon)$ are in E_{-}.

Let $I(\nu_0)$ be the leftmost interval in $\mathcal{I}$. Note that by Proposition 2.6, arcs for $N_{\underline{G}}[\nu_0]$ cannot cover the whole circle. If $N_{\underline{G}}[\nu_0]$ is a separate component in $\underline{G}$, then we can take ν to be the vertex from $N_{\underline{G}}[\nu_0]$ with the leftmost arc; it satisfies condition (C1). Otherwise, let $I(u)$ be the last interval containing $\mathtt{rp}(\nu_0)$ and let $I(u')$ be the next interval (i.e., $I(u')$ is the leftmost interval that does not intersect $I(\nu_0)$). See Fig. 7b. Intervals $I(u)$ and $I(u')$ intersect ($N_{\underline{G}}[\nu_0]$ is not isolated), but $\nu_0 u' \notin E(G)$ because $\underline{G}$ is maximum: Moving $I(\nu_0)$ to the right to intersect $I(u')$ would otherwise make a unit interval model that represents a subgraph of G having one more edge than $\underline{G}$.

- If $u' \to u$, then $u' \to u \to \nu_0$. We argue by contradiction that there cannot be vertices $\nu' \in K_{\mathcal{A}}(\mathtt{cp}(\nu_0) + \epsilon)$ and $w \notin N_G(\nu_0)$ with $u\nu', \nu' w \in E(\underline{G})$. If $u\nu' \in E(\underline{G}) \subseteq E(G)$, then by the position of $A(\nu_0)$, we must have $u \to \nu'$ and $\nu_0 \to \nu'$. Then $\nu_0 u' \notin E(G)$ excludes the possibility $u' \to \nu'$; on the other hand, $\nu' \to u'$ is excluded by Proposition 2.6: The arcs for u', u, ν' cannot cover the whole circle. Therefore, $\nu' u' \notin E(G)$, and likewise $uw \notin E(G)$. They cannot be in $E(\underline{G})$ either, but this is impossible; See Fig. 7b. Let ν be the vertex in $N_{\underline{G}}[\nu_0]$ such that $A(\nu)$ is the rightmost; it satisfies condition (C2).
- If $u \to u'$, then $\nu_0 \to u \to u'$. Similarly as above, the vertex ν in $N_{\underline{G}}[\nu_0]$ such that $A(\nu)$ is the leftmost satisfies condition (C1).

Noting that conditions (C1) and (C2) are symmetric, we assume that the vertex ν found above satisfies condition (C2), and a symmetric argument would apply to condition (C1). Note that by the selection of ν, which has the leftmost arc among $N_{\underline{G}}[\nu_0]$, we have $N_{\underline{G}}[\nu] \subseteq N_G[\nu']$ for every ν' satisfying $\mathtt{lp}(\nu') < \mathtt{lp}(\nu)$; thus setting their intervals to be $[\mathtt{lp}(\nu) + \epsilon, \mathtt{rp}(\nu) + \epsilon]$ would produce another unit interval model for $\underline{G}$. In the rest of the proof we may assume without loss of generality that $I(\nu)$ is the first interval in $\mathcal{I}$.

Since the model $\mathcal{A}$ is proper and Helly, no arc in $\mathcal{A}$ can contain both $\mathtt{ccp}(\nu)$ and $\mathtt{cp}(\nu) + \epsilon$. In other words, $K_{\mathcal{A}}(\mathtt{ccp}(\nu))$ and $K_{\mathcal{A}}(\mathtt{cp}(\nu) + \epsilon)$ are disjoint, and $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(\nu))$ comprises precisely edges between them; see Fig. 7a. By Proposition 4.1, $|E_{-}| \leqslant |\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(\nu))|$. If $K_{\mathcal{A}}(\mathtt{ccp}(\nu))$ is not adjacent to $K_{\mathcal{A}}(\mathtt{cp}(\nu) + \epsilon)$ in $\underline{G}$, then $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(\nu)) \subseteq E_{-}$, and they have to be equal. In this case the proof is complete: $\rho = \mathtt{ccp}(\nu)$. We are hence focused on edges between $K_{\mathcal{A}}(\mathtt{ccp}(\nu))$ and $K_{\mathcal{A}}(\mathtt{cp}(\nu) + \epsilon)$.

Claim 1. *Let $u \in K_{\mathcal{A}}(\mathtt{ccp}(\nu))$. If u is adjacent to $K_{\mathcal{A}}(\mathtt{cp}(\nu) + \epsilon)$ in $\underline{G}$, then $I(u)$ does not intersect the intervals for $N_{\underline{G}}[\nu]$.*

Proof. Recall that $uv \notin E(\underline{G})$ by condition (C2) and the fact that u is adjacent to $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ in $\underline{G}$. Let u' be the vertex in $N_{\underline{G}}[v]$ with the rightmost interval, and let $I(y)$ be the leftmost interval not intersecting $I(v)$. Note that $vy \notin E(G)$: Otherwise moving $I(v)$ to the right to intersect $I(y)$ would make a unit interval model that represents a subgraph of G with one more edge than $\underline{G}$. Suppose to the contrary of this claim that $I(u)$ intersects some interval for $N_{\underline{G}}[v]$, then it intersects $I(u')$. See Fig. 7c. Since v satisfies condition (C2) and by Proposition 2.6, $u' \to v$ and $y \to u, u'$.

Let w be the vertex in $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ that has the leftmost interval. Since v satisfies condition (C2), $\mathtt{lp}(w) > \mathtt{rp}(u')$; by the assumption that u is adjacent to $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ in $\underline{G}$, we have $\mathtt{lp}(w) < \mathtt{rp}(u)$. By Proposition 2.6, $yw \notin E(G)$. Let $A(y')$ be the leftmost arc such that $y' \in K_{\mathcal{A}}(\mathtt{ccp}(u))$ and $I(y')$ lies in $(\mathtt{rp}(v), \mathtt{lp}(w))$; this vertex exists because y itself is a candidate for it. Again by Proposition 2.6, $y'w \notin E(G)$.

Let X denote the set of vertices x with $\mathtt{lp}(x) < \mathtt{lp}(w)$. We make a new interval model by resetting intervals for these vertices. Since every vertex in X is adjacent to at least one of u and u', by Proposition 2.6, the union of arcs for X do not cover the circle in $\mathcal{A}$. These arcs can thus be viewed as a proper interval model for $G[X]$. The new intervals for X, adapted from these arcs, are formally specified as follows. The left endpoint of each $x \in X$ is set as $\mathtt{lp}'(x) = \mathtt{lp}(w) - (\mathtt{ccp}(w) - \mathtt{ccp}(x))$. For each vertex $x \in X \setminus N_G(w)$, we set

$$I'(x) = [\mathtt{lp}(w) - (\mathtt{ccp}(w) - \mathtt{ccp}(x)), \mathtt{lp}(w) - (\mathtt{ccp}(w) - \mathtt{cp}(x))].$$

By the selection of w, we have $x \to w$ for all $x \in X \cap N_G(w)$. Arcs for $X \cap N_G(w)$ are thus pairwise intersecting; by Proposition 2.6, they cannot cover the whole circle. We can thus number vertices in $X \cap N_G(w)$ as $u_1, \ldots, u_p$ such that $u_i \to u_{i+1}$ for each $i = 1, \ldots, p-1$. Their right endpoints are set as

$$\begin{aligned} \mathtt{rp}'(u_1) &= \max\left\{\mathtt{lp}(w)+\epsilon, \mathtt{rp}(u_1)\right\}, && \text{and} \\ \mathtt{rp}'(u_i) &= \max\left\{\mathtt{rp}'(u_{i-1})+\epsilon, \mathtt{rp}(u_i)\right\} && \text{for } i = 2, \ldots, p. \end{aligned}$$

Let $\mathcal{I}'$ denote the resulting new interval model. To see that $\mathcal{I}'$ is proper, note that (a) no new interval can contain or be contained by an interval $I(z)$ for $z \in V(G) \setminus X$; and (b) the left and right endpoints of the intervals for X have the same ordering as the counterclockwise and clockwise endpoints of the arcs $\{A(x) : x \in X\}$, hence necessarily proper. Let G' denote the proper interval graph represented by $\mathcal{I}'$. We want to argue that $E(\underline{G}) \subset E(G') \subseteq E(G)$, which would contradict that $\underline{G}$ is a maximum unit interval subgraph of G, and conclude the proof of this claim.

By construction, $G' - X$ is the same as $\underline{G} - X$, while $G'[X]$ is the same as $G[X]$. Thus, we focus on edges between X and $V(G) \setminus X$, which are all incident to $X \cap N_G(w)$. For each $i = 1, \ldots, p$, we have $\mathtt{rp}'(u_i) \geqslant \mathtt{rp}(u_i)$, and thus $E(\underline{G}) \subseteq E(G')$; on the other hand, they are not equal because $uv \in E(G') \setminus E(\underline{G})$. We show by induction that for every $i = 1, \ldots, p$, the edges incident to u_i in G' is a subset of G. The base case is clear: $N_{G'}(u_1) \setminus X$ is either $\{w\}$ or $N_{\underline{G}}(u_1) \setminus X$. For the inductive step, if $\mathtt{rp}'(u_i) = \mathtt{rp}(u_i)$, then $N_{G'}(u_i) \setminus X = N_{\underline{G}}(u_i) \setminus X \subseteq N_G(u_i)$; otherwise, $N_{G'}(u_i) \setminus X \subseteq N_{\underline{G}}(u_{i-1}) \setminus X$, which is a subset of $N_G(u_i)$ because $u_{i-1} \to u_i$. This verifies $E(G') \subseteq E(G)$. ⌟

We consider then edges deleted from each vertex in $K_{\mathcal{A}}(\mathtt{ccp}(v))$, i.e., edges in E_- that is incident to $K_{\mathcal{A}}(\mathtt{ccp}(v))$.

Claim 2. *Let $u \in K_{\mathcal{A}}(\mathtt{ccp}(v))$. If u is adjacent to $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ in $\underline{G}$, then there are strictly more edges incident to u in E_- than in $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(v))$ (i.e., between u and $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$.).*

Proof. The vertices in $N_G(u)$ consists of three parts, $N_{\underline{G}}[v]$, those in $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$, and others. By Claim 1, edges between u and all vertices in $N_{\underline{G}}[v]$ are in E_-; they are not in $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(v))$. All edges between u and $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ are in $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(v))$. In $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(v))$ there is no edge between u and the other vertices. Therefore, to show the claim, it suffices to show $|N_{\underline{G}}[u] \cap K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)| \leqslant |N_{\underline{G}}(v)| < |N_{\underline{G}}[v]|$.

Let uw be an edge of $\underline{G}$ with $w \in K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$. Since v satisfies condition (C2), both uv and vw are in E_-. We show only the case that $I(w)$ intersects $I(u)$ from the right; a similar argument works for the other case: Note that all the intervals used below are disjoint from $I(v)$.

Consider first that there exists another vertex $w' \in K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ with interval $I(w')$ intersecting $I(u)$ from the left. See Fig. 7d. Any interval that intersects $I(u)$ necessarily intersects at least one of $I(w)$ and $I(w')$, and thus by Proposition 2.6, its vertex must be in $N_G(v)$. As a result, setting $I(v)$ to $[\mathtt{lp}(u)-\epsilon, \mathtt{rp}(u)-\epsilon]$ gives another proper interval model and it represents a subgraph of G. Since $\underline{G}$ is maximum, we can conclude $|N_{\underline{G}}[u]| \leqslant |N_{\underline{G}}(v)| < |N_{\underline{G}}[v]|$. In this case the proof of this claim is concluded.

In the second case there is no vertex in $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ whose interval intersects $I(u)$ from the left. Note that every interval intersecting $[\mathtt{lp}(w), \mathtt{rp}(u)]$ represents a vertex in $N_G[v]$. Let w' be vertex in

$K_{\mathcal{A}}(\mathtt{cp}(v)) \cap K_{\mathcal{J}}(\mathtt{rp}(u))$ that has the leftmost interval, and let u' be vertex in $K_{\mathcal{A}}(\mathtt{ccp}(v)) \cap K_{\mathcal{J}}(\mathtt{lp}(w))$ that has the rightmost interval; these two vertices exist because u and w are candidates for them, respectively. See Fig. 7e. There cannot be any vertex whose interval contains $[\mathtt{lp}(w'), \mathtt{rp}(u')]$: Such a vertex, if it exists, is in $N_G(v)$, but then it contradicts the selection of u' and w'. Also by the selection of u' and w', no interval contains $[\mathtt{lp}(w') - \epsilon, \mathtt{rp}(u') + \epsilon]$. Thus, setting $I(v)$ to $[\mathtt{lp}(w') - \epsilon, \mathtt{rp}(u') + \epsilon]$ gives another proper interval model and it represents a subgraph of G. Since $\underline{G}$ is maximum, we can conclude $|N_{\underline{G}}[u] \cap K_{\mathcal{A}}(\mathtt{cp}(v) + \epsilon)| \leqslant |N_{\underline{G}}(v)| < |N_{\underline{G}}[v]|$. ⌟

Therefore, for every vertex $u \in K_{\mathcal{A}}(\mathtt{ccp}(v))$, there are no less edges incident to u in E_- than in $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(v))$. Moreover, as there is at least one vertex in $K_{\mathcal{A}}(\mathtt{ccp}(v))$ adjacent to $K_{\mathcal{A}}(\mathtt{cp}(v)+\epsilon)$ in $\underline{G}$, (noting that no edge in $\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(v))$ is incident to two vertices in $K_{\mathcal{A}}(\mathtt{ccp}(v))$,) it follows that $|E_-| > |\overrightarrow{E}_{\mathcal{A}}(\mathtt{ccp}(v))|$, contradicting that $\underline{G}$ is a maximum unit interval subgraph of G. □

It is worth stressing that a thinnest place in an arc model with respect to edges is not necessarily a thinnest place with respect to vertices; see Fig. 8 for an example. There is a linear number of different places to check, and thus the edge deletion problem can also be solved in linear time on proper Helly circular-arc graphs. The problem is also simple on fat W_5's.

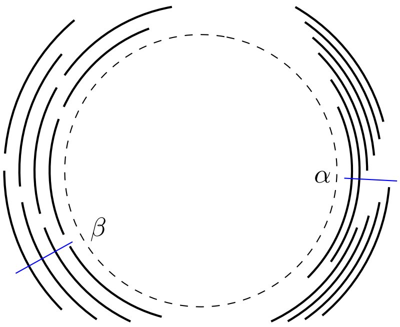


Figure 8: The thinnest points for vertices and edges are α and β respectively: $K_{\mathcal{A}}(\alpha) = 2 < K_{\mathcal{A}}(\beta) = 3$; while $\overrightarrow{E}_{\mathcal{A}}(\alpha) = 8 > \overrightarrow{E}_{\mathcal{A}}(\beta) = 6$.

Theorem 4.3. *The unit interval edge deletion problem can be solved in* $O(m)$ *time (1) on proper Helly circular-arc graphs and (2) on* $\mathcal{F}$*-free graphs.*

Proof. For (1), we may assume that the input graph G is not an unit interval graph; according to Corollary 2.3, it is not chordal. We build a proper and Helly arc model $\mathcal{A}$ for G; without loss of generality, assume that it is canonical. According to Lemma 4.2, the problem reduces to finding a point α in $\mathcal{A}$ such that $\overrightarrow{E}_{\mathcal{A}}(\alpha)$ is minimized. It suffices to consider the $2n$ points $i + 0.5$ for $i \in \{0, \ldots, 2n-1\}$. We calculate first $\overrightarrow{E}_{\mathcal{A}}(0.5)$, and then for $i = 1, \ldots, 2n-1$, we deduce $\overrightarrow{E}_{\mathcal{A}}(i+0.5)$ from $\overrightarrow{E}_{\mathcal{A}}(i-0.5)$ as follows. If i is a clockwise endpoint of some arc, then $\overrightarrow{E}_{\mathcal{A}}(i+0.5) = \overrightarrow{E}_{\mathcal{A}}(i-0.5)$. Otherwise, $i = \mathtt{ccp}(v)$ for some vertex v, then the difference between $\overrightarrow{E}_{\mathcal{A}}(i+0.5)$ and $\overrightarrow{E}_{\mathcal{A}}(i-0.5)$ is the set of edges incident to v. In particular, $\{uv : u \to v\} = \overrightarrow{E}_{\mathcal{A}}(i-0.5) \setminus \overrightarrow{E}_{\mathcal{A}}(i+0.5)$, while $\{uv : v \to u\} = \overrightarrow{E}_{\mathcal{A}}(i+0.5) \setminus \overrightarrow{E}_{\mathcal{A}}(i-0.5)$. Note that the initial value $\overrightarrow{E}_{\mathcal{A}}(0.5)$ can be calculated in $O(m)$ time, and then each vertex and its adjacency list is scanned exactly once. It follows that the total running time is $O(m)$.

For (2), we may assume that the input graph G is connected, as otherwise we work on its components one by one. According to Theorem 2.5(1), G is either a proper Helly circular-arc graph or a fat W_5. The former case has been considered above, and now assume G is a fat W_5. Let $K_0, \ldots, K_4$ be the five cliques in the fat hole, and let K_5 be the hub. We may look for a maximum spanning unit interval subgraph $\underline{G}$ of G such that $N_{\underline{G}}[u] = N_{\underline{G}}[v]$ for any pair of vertices u, v in K_i, where $i \in \{0, \ldots, 5\}$. We now argue the existence of such a subgraph. By definition, $N_G[u] = N_G[v]$. Let G' be a maximum spanning unit interval subgraph of G where $N_{G'}[u] \neq N_{G'}[v]$, and assume without loss of generality, $|N_{G'}[u]| \geqslant |N_{G'}[v]|$. We may change the deleted edges that are incident to v to make another subgraph where v has the same neighbors as $N_{G'}[u]$; this graph is clearly a unit interval graph and has no less edges than G'. This operation can be applied to any pair of u, v in the same twin class, and it will not violate an earlier pair. Repeating it we will finally end with a desired maximum spanning unit interval subgraph. Therefore, there is always some $i \in \{0, \ldots, 4\}$ (all subscripts are modulo 5) such that deleting all edges between cliques K_i and K_{i+1} together with edges between one of them and K_5 leaves a maximum spanning unit interval subgraph. Once the sizes of all six cliques have been calculated, which can be done in $O(m)$ time, the minimum set of edges can be decided in constant time. Therefore, the total running time is $O(m)$. The proof is now complete. □

Indeed, it is not hard to see that in the proof of Theorems 4.3(2), *every* maximum spanning unit interval subgraph of a fat W_5 keeps its six twin classes, but we are satisfied with the weaker statement that is sufficient for our algorithm.

Theorems 4.3 and 2.5 already imply a branching algorithm for the unit interval edge deletion problem running in time $O(9^k \cdot m)$. Here the constant 9 is decided by the S_3, which has 9 edges. However, a closer look at it tells us that deleting any single edge from an S_3 introduces either a claw or a C_4, which forces us to delete some other edge(s). The disposal of an $\overline{S_3}$ is similar. The labels for an S_3 and a $\overline{S_3}$ used in the following proof are as given in Fig. 1.

Proposition 4.4. *Let $\underline{G}$ be a spanning unit interval subgraph of a graph G, and let $F = E(G) \setminus E(\underline{G})$.*

(1) For an S_3 of G, there must be some $i = 1, 2, 3$ such that F contains at least two edges from the triangle involving u_i.

(2) For an $\overline{S}_3$ of G, the set F contains either an edge $u_i v_i$ for some $i = 1, 2, 3$, or at least two edges from the triangle $u_1 u_2 u_3$.

Proof. (1) We consider the intervals for v_1, v_2, and v_3 in a unit interval model $\mathcal{I}$ for $\underline{G}$. If $v_1 v_2 v_3$ remains a triangle of $\underline{G}$, then the interval $I(v_1) \cup I(v_2) \cup I(v_3)$ has length less than 2, and it has to be disjoint from $I(u_i)$ for at least one $i \in \{1, 2, 3\}$. In other words, both edges incident to this u_i are in F. Otherwise, $v_1 v_2 v_3$ is not a triangle of $\underline{G}$. Assume without loss of generality $\mathtt{lp}(v_1) < \mathtt{lp}(v_2) < \mathtt{lp}(v_3)$. Then $v_1 v_3 \notin E(\underline{G})$ and u_2 cannot be adjacent to both v_1 and v_3. Therefore, at least two edges from the triangle $u_2 v_1 v_3$ are in F; other cases are symmetric.

(2) If F contains none of the three edges $v_1 u_1$, $v_2 u_2$, and $v_3 u_3$, then it contains at least two edges from the triangle $u_1 u_2 u_3$: Otherwise there is a claw. □

This observation and a refined analysis will yield the running time claimed in Theorem 1.1. The algorithm goes similarly as the parameterized algorithm for unit interval vertex deletion used in the proof of Theorem 3.3.

Theorem 4.5. *The unit interval edge deletion problem can be solved in time $O(4^k \cdot m)$.*

Proof. The algorithm calls first Theorem 2.5(2) to decide whether there exists an induced subgraph in $\mathcal{F}$, and then based on the outcome, it solves the problem by making recursive calls to itself, or calling the algorithm of Theorem 4.3. When a claw or C_4 is found, the algorithm makes respectively 3 or 4 calls to itself, each with a new instance with parameter value $k - 1$ (deleting one edge from the claw or C_4). For an S_3, the algorithm branches on deleting two edges from a triangle involving a vertex u_i with $i = 1, 2, 3$. Since there are three such triangles, and each has three options, the algorithm makes 9 calls to itself, all with parameter value $k - 2$. For an $\overline{S_3}$, the algorithm makes 6 calls to itself, of which 3 with parameter value $k - 1$ (deleting edge $v_i u_i$ for $i = 1, 2, 3$), and another 3 with parameter value $k - 2$ (deleting two edges from the triangle $v_1 v_2 v_3$).

To verify the correctness of the algorithm, it suffices to show that for any spanning unit interval subgraph $\underline{G}$ of G, there is at least one recursive call that generates a graph G' satisfying $E(\underline{G}) \subseteq E(G') \subseteq E(G)$. This is obvious when the recursive calls are made on a claws or C_4. It follows from Proposition 4.4 when the recursive calls are made on an S_3 or $\overline{S_3}$. With standard technique, it is easy to verify that $O(4^k)$ recursive calls are made, each in $O(m)$ time. Moreover, the algorithm for Theorem 4.3 is called $O(4^k)$ times. It follows that the total running time of the algorithm is $O(4^k \cdot m)$. □

What dominates the branching step is the disposal of C_4's. With the technique the author developed in [23], one may (slightly) improve the running time to $O(c^k \cdot m)$ for some constant $c < 4$. To avoid blurring the focus of the present paper, we omit the details.

5 General editing

Let $V_- \subseteq V(G)$, and let E_- and E_+ be a set of edges and a set of non-edges of $G - V_-$ respectively. We say that (V_-, E_-, E_+) is an *editing set* of G if the deletion of E_- from and the addition of E_+ to $G - V_-$ create a unit interval graph. Its *size* is defined to be the 3-tuple $(|V_-|, |E_-|, |E_+|)$, and we say that it is *smaller* than (k_1, k_2, k_3) if all of $|V_-| \leqslant k_1$ and $|E_-| \leqslant k_2$ and $|E_+| \leqslant k_3$ hold true and at least one inequality is strict. The unit interval editing problem is formally defined as follows.

Input:	A graph G and three nonnegative integers k_1, k_2, and k_3.
Task:	Either construct an editing set (V_-, E_-, E_+) of G that has size at most (k_1, k_2, k_3), or report that no such set exists.

We remark that it is necessary to impose the quotas for different modifications in the stated, though cumbersome, way. Since vertex deletions are clearly preferable to both edge operations, the problem would be computationally equivalent to unit interval vertex deletion if we have a single budget on the total number of operations.

By and large, our algorithm for the unit interval editing problem also uses the same two-phase approach as the previous algorithms. The main discrepancy lies in the first phase, when we are not satisfied with a proper Helly circular-arc graph or an $\mathcal{F}$-free graph. In particular, we also want to dispose of all holes C_ℓ with $\ell \leqslant k_3 + 3$, which are precisely those holes fixable by merely adding edges (recall that at least $\ell - 3$ edges are needed to fill a C_ℓ in). In the very special cases where $k_3 = 0$ or 1, a fat W_5 is $\mathcal{F} \cup \{C_\ell : \ell \leqslant k_3 + 3\}$-free. It is not hard to solve fat W_5's, but to make the rest more focused and also simplify the presentation, we also exclude these cases by disposing of all C_5's in the first phase.

A graph is called *reduced* if it contains no claw, S_3, $\overline{S_3}$, C_4, C_5, or C_ℓ with $\ell \leqslant k_3 + 3$. By Proposition 2.4, a reduced graph G is a proper Helly circular-arc graph. Hence, if G happens to be chordal, then it must be a unit interval graph (Corollary 2.3), and we terminate the algorithm. Otherwise, our algorithm enters the second phase. Now that G is reduced, every minimal forbidden induced subgraph is a hole C_ℓ with $\ell > k_3 + 3$, which can only be fixed by deleting vertices and/or edges. Here we again exploit a proper and Helly arc model $\mathcal{A}$ for G. According to Lemma 3.1, if there exists some point ρ in the model such that $|K_{\mathcal{A}}(\rho)| \leqslant k_1$, then it suffices to delete all vertices in $K_{\mathcal{A}}(\rho)$, which results in a subgraph that is a unit interval graph. Therefore, we may assume hereafter that no such point exists, then G remains reduced and non-chordal after at most k_1 vertex deletions. As a result, we have to delete edges as well.

Consider an (inclusion-wise minimal) editing set (V_-, E_-, E_+) to a reduced graph G. It is easy to verify that $(\emptyset, E_-, E_+)$ is an (inclusion-wise minimal) editing set of the reduced graph $G - V_-$. In particular, E_- needs to intersect all holes of $G - V_-$. We use $\mathcal{A} - V_-$ as a shorthand for $\{A(v) \in \mathcal{A} : v \notin V_-\}$, an arc model for $G - V_-$ that is proper and Helly. One may want to use Lemma 4.2 to find a minimum set E_- of edges (i.e., $\overrightarrow{E}_{\mathcal{A}-V_-}(\alpha)$ for some point α) to finish the task. However, Lemma 4.2 has not ruled out the possibility that we delete less edges to break all long holes, and subsequently add edges to fix the incurred subgraphs in {claw, $\overline{S_3}$, S_3, C_4, C_5, C_ℓ} with $\ell \leqslant k_3 + 3$. So we need the following lemma.

Lemma 5.1. *Let (V_-, E_-, E_+) be an inclusion-wise minimal editing set of a reduced graph G. If $|E_+| \leqslant k_3$, then $E_+ = \emptyset$.*

Proof. We may assume without loss of generality $V_- = \emptyset$, as otherwise it suffices to consider the inclusion-wise minimal editing set $(\emptyset, E_-, E_+)$ to the still reduced graph $G - V_-$. Let $\mathcal{A}$ be a proper and Helly arc model for G. Let E'_- be an inclusion-wise minimal subset of E_- such that for every hole in $G - E'_-$, the union of arcs for its vertices does not cover the circle of $\mathcal{A}$. We argue the existence of E'_- by showing that E_- itself satisfies this condition. Suppose for contradiction that there exists in $G - E_-$ a hole whose arcs cover the circle of $\mathcal{A}$. Then we can find a minimal subset of them that covers the circle of $\mathcal{A}$. By Corollary 2.7, this subset has at least $k_3 + 4$ vertices, and thus the length of the hole in $G - E_-$ is at least $k_3 + 4$. But then it cannot be fixed by the addition of the at most k_3 edges from E_+.

Now for the harder part, we argue that $\underline{G} := G - E'_-$ is already a unit interval graph. Together with the inclusion-wise minimality, it would imply $E_- = E'_-$ and $E_+ = \emptyset$.

Suppose for contradiction that $\underline{G}[X]$ is a claw, $S_3, \overline{S_3}$, or a hole for some $X \subseteq V(G)$. We find three vertices $u, v, w \in X$ such that $uw \in E'_-$ and $uv, vw \in E(\underline{G})$ as follows. By Corollary 2.7 and the fact that G is $\{C_4, C_5\}$-free, at least six arcs are required to cover the circle. As a result, if the arcs for a set Y of at most six vertices covers the circle, then $G[Y]$ must be a C_6, and its subgraph $\underline{G}[Y]$ cannot be a claw, $\overline{S_3}$, or S_3. Therefore, $\bigcup_{v \in X} A(v)$ cannot cover the whole circle when $\underline{G}[X]$ is a claw, S_3, or $\overline{S_3}$. On the other hand, from the selection of E'_-, this is also true when $\underline{G}[X]$ is a hole. Thus, $G[X]$ is a unit interval graph, and we can find two vertices x, z from X having $xz \in E'_-$. We find a shortest x-z path in $\underline{G}[X]$. If the path has more than one inner vertex, then it makes a hole together with xz; as $G[X]$ is a unit interval graph, this would imply that there exists an inner vertex y of this path such that $xy \in E'_-$ or $yz \in E'_-$. We consider then the new pair x, y or y, z accordingly. Note that their distance in $\underline{G}[X]$ is smaller than xz, and hence repeating this argument (at most $|X| - 3$ times) will end with two vertices with distance precisely 2 in $\underline{G}[X]$. They are the desired u and w, while any common neighbor of them in $\underline{G}[X]$ can be v.

By the minimality of E'_-, in $\underline{G} + uw$ there exists a hole H such that arcs for its vertices cover the circle in $\mathcal{A}$. This hole H necessarily passes uw, and we denote it by $x_1 x_2 \cdots x_{\ell-1} x_\ell$, where $x_1 = u$ and $x_\ell = w$.

Note that $A(u)$ intersects $A(w)$, and since $\mathcal{A}$ is proper and Helly, $A(u), A(v), A(w)$ cannot cover the circle; moreover, it cannot happen that $A(v)$ intersects all the arcs $A(x_i)$ for $1 < i < \ell$ simultaneously. From $x_1 x_2 \cdots x_{\ell-1} x_\ell$ we can find p and q such that $1 \leqslant p < p+1 < q \leqslant \ell$ and $vx_p, vx_q \in E(\underline{G})$ but $vx_i \notin E(\underline{G})$ for every $p < i < q$. Here possibly $p = 1$ and/or $q = \ell$. Then $vx_p \cdots x_q$ makes a hole of $\underline{G}$, and the union of its arcs covers the circle, contradicting the definition of E'_-. This concludes the proof. □

Therefore, a yes-instance on a reduced graph always has a solution that does not add any edge. By Lemma 4.2, for any editing set $(V_-, E_-, \emptyset)$, we can always find some point α in the model and use $\overrightarrow{E}_{\mathcal{A}-V_-}(\alpha)$ to replace E_-. After that, we can use the vertices "close" to this point to replace V_-. Therefore, the problem again boils down to find some "weak point" in the arc model. This observation is formalized in the following lemma. We point out that this result is stronger than required by the linear-time algorithm, and we present in the current form for its own interest (see Section 6 for more discussions).

Lemma 5.2. *Given a proper Helly circular-arc graph G and a nonnegative integer p, we can calculate in $O(m)$ time the minimum number q such that G has an editing set of size $(p, q, 0)$. In the same time we can find such an editing set.*

Proof. We may assume that G is not chordal; otherwise, by Corollary 2.3, G is a unit interval graph and the problem becomes trivial because an empty set will suffice. Let us fix a proper and Helly arc model $\mathcal{A}$ for G. The lemma follows from Lemma 3.1 when there is some point ρ satisfying $K_{\mathcal{A}}(\rho) \leqslant p$. Hence, we may assume that no such point exists, and for any subset V_- of at most p vertices, $G - V_-$ remains a proper Helly circular-arc graph and is non-chordal. Hence, $q > 0$. For each point ρ in $\mathcal{A}$, we can define an editing set $(V^\rho_-, E^\rho_-, \emptyset)$ by taking the p vertices in $K_{\mathcal{A}}(\rho)$ with the most clockwise arcs as V^ρ_- and $\overrightarrow{E}_{\mathcal{A}-V^\rho_-}(\rho)$ as E^ρ_-. We argue first that the minimum cardinality of this edge set, taken among all points in $\mathcal{A}$ is the desired number q. See Fig. 9.

Let $(V^*_-, E^*_-, \emptyset)$ be an editing set of G with size $(p, q, 0)$. According to Lemma 4.2, there is a point α such that the deletion of $E'_- := \overrightarrow{E}_{\mathcal{A}-V^*_-}(\alpha)$ from $G - V^*_-$ makes it a unit interval graph and $|E'_-| \leqslant |E^*_-|$. We now consider the original model $\mathcal{A}$. Note that a vertex in V^*_- is in either $K_{\mathcal{A}}(\alpha)$ or $\{v \notin K_{\mathcal{A}}(\alpha) : u \to v, u \in K_{\mathcal{A}}(\alpha)\}$; otherwise replacing this vertex by any end of an edge in E^*_-, and removing this edge from E^*_- gives an editing set of size $(p, q-1, 0)$. Let V_- comprise the $|V^*_- \cap K_{\mathcal{A}}(\alpha)|$ vertices of $K_{\mathcal{A}}(\alpha)$ whose arcs are the most clockwise in them, as well as the first $|V^*_- \setminus K_{\mathcal{A}}(\alpha)|$ vertices whose arcs are immediately to the right of α. And let $E_- := \overrightarrow{E}_{\mathcal{A}-V_-}(\alpha)$. It is easy to verify that $|E_-| \leqslant |E^*_-| = q$ and $(V_-, E_-, \emptyset)$ is also an editing set of G (Lemma 4.1). Note that arcs for V_- are consecutive in $\mathcal{A}$. Let v be the vertex in V_- with the most clockwise arc, and then $\mathtt{ccp}(v) + \epsilon$ is the desired point ρ.

We give now the $O(m)$-time algorithm for finding the desired point, for which we assume that $\mathcal{A}$ is canonical. It suffices to consider the $2n$ points $i + 0.5$ for $i \in \{0, \ldots, 2n-1\}$. We calculate first the $V^{0.5}_-$ and $E^{0.5}_-$, and maintain a queue that is initially set to be $V^{0.5}_-$. For $i = 1, \ldots, 2n-1$, we deduce the new sets $V^{i+0.5}_-$ and $E^{i+0.5}_-$ from $V^{i-0.5}_-$ and $E^{i-0.5}_-$ as follows. If i is a clockwise endpoint of some arc, then $V^{i+0.5}_- = V^{i-0.5}_-$ and $E^{i+0.5}_- = E^{i-0.5}_-$. Otherwise, $i = \mathtt{ccp}(v)$ for some vertex v, then we enqueue v, and dequeue u. We set $V^{i+0.5}_-$ to be the vertices in the queue, whose size remains p. The different edges between $E^{i+0.5}_-$ and $E^{i-0.5}_-$ are those incident to u and v. In particular, $E^{i+0.5}_- \setminus E^{i-0.5}_- = \{ux : u \to x, x \notin V^{i+0.5}_-\}$, while $E^{i-0.5}_- \setminus E^{i+0.5}_- = \{xv : x \to v, x \notin V^{i-0.5}_-\}$. Note that the initial sets $V^{0.5}_-$ and $E^{0.5}_-$ can be found in $O(m)$ time, and then each vertex and its adjacency is scanned exactly once. The total running time is $O(m)$. This concludes the proof. □

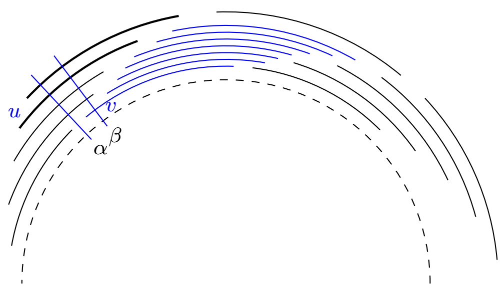


Figure 9: Illustration for the proof of Lemma 5.2. Here $p = 2$ and $q = 3$. Then V^α_- consists of the two thick arcs, and $|E^\alpha_-| = 3$. Moving from point α to β $(= \alpha + 1)$ gives $|E^\beta_-| = |E^\alpha_-| + 4 - 2 = 5$.

Again, one should note that in the general case (when both $p, q > 0$), the point identified by Lemma 5.2 may not be the thinnest point for vertices or the thinnest point for edges, as specified by respectively

Lemmas 3.1 and 4.2. Indeed, for different values of p, the thinnest points found by Lemma 5.2 may be different.

The mixed hole covers consists of both vertices and edges, and thus the combinatorial characterization given in Lemma 5.2 extends Lemmas 3.1 and 4.2. The algorithm used in the proof is similar as that of Theorem 4.3. Recall that a reduced graph is a proper Helly circular-arc graph. Thus, Lemmas 5.1 and 5.2 have the following consequence: It suffices to call the algorithm with $p = k_1$, and returns the found editing set if $q \leqslant k_2$, or "NO" otherwise.

Corollary 5.3. *The unit interval editing problem can be solved in* $O(m)$ *time on reduced graphs.*

Putting together these steps, the fixed-parameter tractability of unit interval editing follows. Note that to fill a hole, we need to add an edge whose ends have distance 2.

Proof of Theorem 1.4. We start by calling Theorem 2.5. If a subgraph in $\mathcal{F}$ or W_5 is detected, then we branch on all possible ways of destroying it or the contained C_5. Otherwise, we have in our disposal a proper and Helly arc model for G, and we call Lemma 2.8 to find a shortest hole C_ℓ. If $\ell \leqslant k_3 + 3$, then we either delete one of its ℓ vertices and ℓ edges, or add one of ℓ edges $h_i h_{i+2}$ (the subscripts are modulo ℓ). One of the three parameters decreases by 1. We repeat these two steps until some parameter becomes negative, then we terminate the algorithm by returning "NO"; or the graph is reduced, and then call the algorithm of Corollary 5.3 to solve it. The correctness of this algorithm follows from Lemma 2.8 and Corollary 5.3. In the disposal of a subgraph of $\mathcal{F}$, at most 21 recursive calls are made, while 3ℓ for a C_ℓ, each having a parameter $k - 1$. Therefore, the total number of instances (with reduced graphs) made in the algorithm is $O((3k_3 + 21)^k)$. It follows that the total running time of the algorithm is $2^{O(k \log k)} \cdot m$. □

It is worth mentioning that Lemma 5.2 actually implies a linear-time algorithm for the unit interval deletion problem (which allows k_1 vertex deletions and k_2 edge deletions) on the proper Helly circular-arc graphs and an $O(10^{k_1+k_2} \cdot m)$-time algorithm for it on general graphs. The constant 10 can be even smaller if we notice that (1) the problem is also easy on fat W_5's, and (2) the worst cases for vertex deletions (S_3's and $\overline{S_3}$'s) and edge deletions (C_4's) are different.

6 Concluding remarks

All aforementioned algorithms exploit the characterization of unit interval graphs by forbidden induced subgraphs [29]. Very recently, Bliznets et al. [4] used a different approach to produce a subexponential-time parameterized algorithm for unit interval completion (whose polynomial factor is however not linear). Using a parameter-preserving reduction from vertex cover [21], one can show that the vertex deletion version cannot be solved in $2^{o(k)} \cdot n^{O(1)}$ time, unless the Exponent Time Hypothesis fails [7]. Now that the edge deletion version is FPT as well, one may want to ask to which side it belongs. The evidence we now have is in favor of the hard side: In all related graph classes, the edge deletion versions seem to be harder than their vertex deletion counterparts.

As said, it is not hard to slightly improve the constant c in the running time $O(c^k \cdot m)$, but a significant improvement would need some new observation(s). More interesting would be to fathom their limits. In particular, can the deletion problems be solved in time $O(2^k \cdot m)$?

Polynomial kernels for unit interval completion [2] and unit interval vertex deletion [13] have been known for a while. Using the approximation algorithm of Theorem 1.2, we [19] recently developed an $O(k^4)$-vertex kernel for unit interval vertex deletion, improving from the $O(k^{53})$ one of Fomin et al. [13]. We conjecture that the unit interval edge deletion problem also has a small polynomial kernel.

The algorithm for unit interval editing is the second nontrivial FPT algorithm for the general editing problem. The main ingredient of our algorithm is the characterization of the mixed deletion of vertices and edges to break holes. A similar study has been conducted in the algorithm for the chordal editing problem [10]. In contrast to that, Lemmas 5.1 and 5.2 are somewhat stronger. For example, we have shown that once small forbidden subgraphs have been all fixed, no edge additions are further needed. Together with Marx, we had conjectured that this is also true for the chordal editing problem, but we failed to find a proof. Very little study had been done on the mixed deletion of vertices and edges [25]. We hope that our work will trigger more studies on this direction, which will further deepen our understanding of various graph classes.

We point out that although we start by breaking small forbidden induced subgraphs, our major proof technique is instead manipulating (proper/unit) interval models. The technique of combining

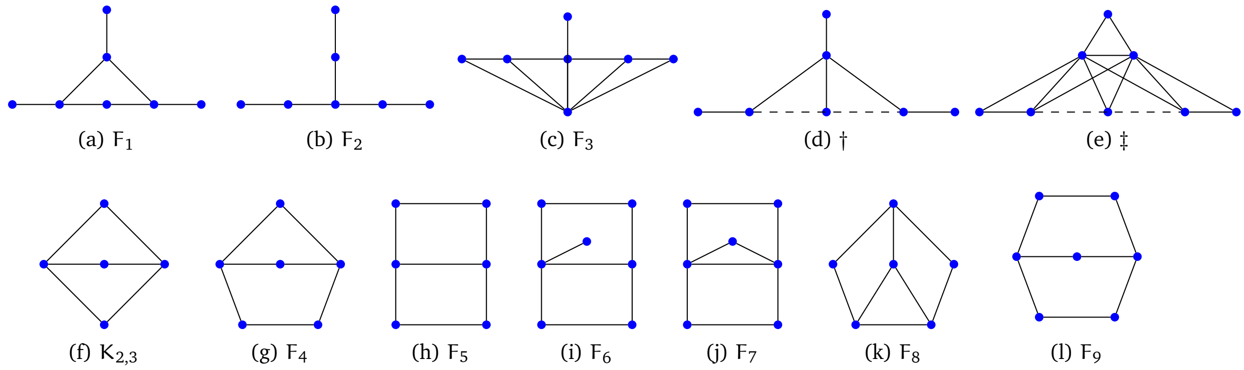


Figure 10: Forbidden induced graphs.

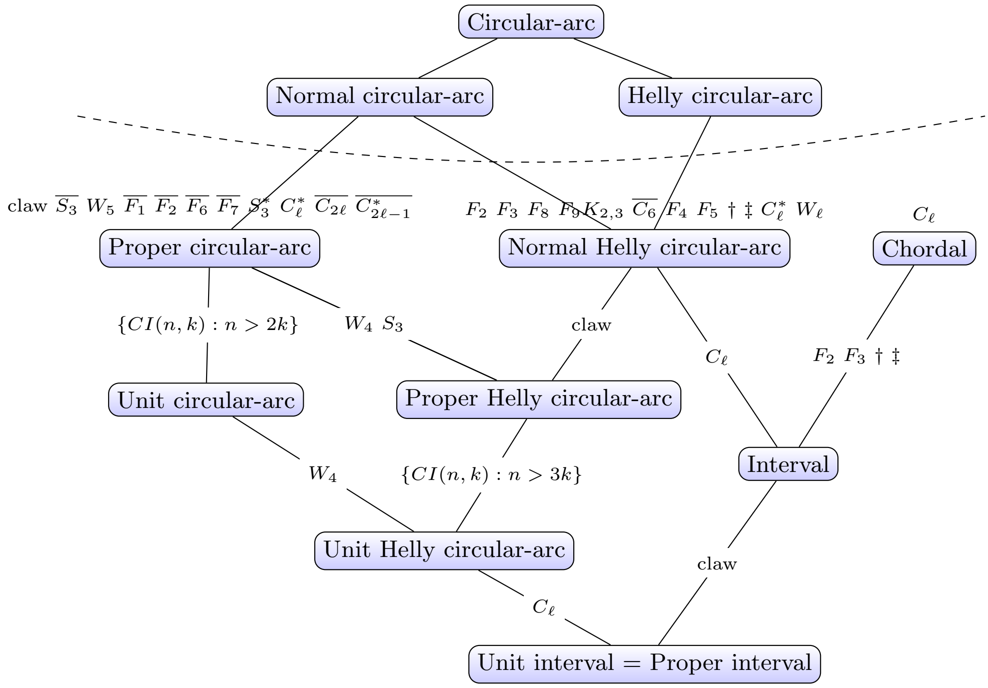


Figure 11: Forbidden induced subgraphs and containment relations of related graph classes ($\ell \geqslant 4$)

(constructive) interval models and (destructive) forbidden induced subgraphs is worth further study on related problems.

Appendix

For the convenience of the reader, we collect related graph classes and their containment relations in Fig. 11, which is adapted from Lin et al. [22] (note that some of these graph classes are not used in the present paper). The $\mathrm{CI}(n, k)$ graphs are defined by Tucker [27]; see also [22]. Other subgraphs that have not been introduced in the main text are depicted in Fig. 10. The relations in Fig. 11 can be viewed both from the intersection models, arcs or intervals, and forbidden induced subgraphs, every minimal forbidden induced subgraph of a super-class being a (not necessarily minimal) forbidden induced subgraph of its subclass. For example, Proposition 2.6 and Corollary 2.7 are actually properties of normal Helly circular-arc graphs. For a normal Helly circular-arc graph that is not chordal, every arc model has to be normal and Helly [22, 9]. This is also true for the subclass of proper Helly circular-arc graphs, but an arc model for a proper Helly circular-arc graph may not be proper.

A word of caution is worth on the definition of proper Helly circular-arc graphs. One graph might admit two arc models, one being proper and the other Helly, but no arc model that is both proper and

Helly, e.g., the S_3 and the W_4. Therefore, the class of proper Helly circular-arc graphs does not contain all those graphs being both proper circular-arc graphs and Helly circular-arc graphs, but a proper subclass of it. A similar remark applies to normal Helly circular-arc graphs.

For the three classes at the top of Fig. 11, their characterizations by minimal forbidden induced subgraphs are still open. At the third level, the minimal forbidden induced subgraphs for proper circular-arc graphs and normal Helly circular-arc graphs are completely determined by Tucker [27] and Cao et al. [9]. For all the classes at lower levels, their forbidden induced subgraphs with respect to its immediate super-classes are given. From them we are able to derive all the minimal forbidden induced subgraphs for each of these classes.

For example, the characterization of unit interval graphs (Theorem 2.1) follows from the characterization of interval graphs and that we can find a claw in an F_2, an F_3, a † that is not an $\overline{S_3}$, or a ‡ that is not an S_3. Likewise, the minimal forbidden induced subgraphs of proper Helly circular-arc graphs stated in Theorem 2.2 can be derived from those of proper circular-arc graphs and by Corollary 5 of [22]: A proper circular-arc graph that is not a proper Helly circular-arc graph must contain a W_4 or S_3. Clearly, S_3^* contains an S_3. To see that each of $\overline{F_1}, \overline{F_2}, \overline{F_6}, \overline{F_7}$, and $\overline{C_{2\ell}}, \overline{C_{2\ell-1}^*}$ for $\ell \geqslant 4$ contain a W_4, it is equivalent to check that each of F_1, F_2, F_6, F_7, and $C_{2\ell}, C_{2\ell-1}^*$ for $\ell \geqslant 4$ contains a $\overline{W_4}$, i.e., two non-incident edges and another independent vertex v. This can be directly read from Fig. 10 for F_1, F_2, F_6, F_7. Let $h_1, h_2, \dots$ denote the vertices in the hole of $C_{2\ell}$ and $C_{2\ell-1}^*$. Then edges h_1h_2 and h_4h_5 are non-incident. In a C_7^*, the vertex not in the hole can be the v, while in all other holes longer than 7, the vertex h_7 can be the v.

References

[1] Jørgen Bang-Jensen and Pavol Hell. On chordal proper circular arc graphs. *Discrete Mathematics*, 128(1):395–398, 1994. doi:10.1016/0012-365X(94)90130-9.

[2] Stéphane Bessy and Anthony Perez. Polynomial kernels for proper interval completion and related problems. *Information and Computation*, 231:89–108, 2013. doi:10.1016/j.ic.2013.08.006.

[3] René van Bevern, Christian Komusiewicz, Hannes Moser, and Rolf Niedermeier. Measuring indifference: Unit interval vertex deletion. In Dimitrios M. Thilikos, editor, *Graph-Theoretic Concepts in Computer Science (WG)*, volume 6410 of *LNCS*, pages 232–243. Springer, 2010. doi:10.1007/978-3-642-16926-7_22.

[4] Ivan Bliznets, Fedor V. Fomin, Marcin Pilipczuk, and Michał Pilipczuk. A subexponential parameterized algorithm for proper interval completion. *SIAM Journal on Discrete Mathematics*, 29(4):1961–1987, 2015. A preliminary version appeared in ESA 2014. doi:10.1137/140988565.

[5] Hans L. Bodlaender and Babette van Antwerpen-de Fluiter. On intervalizing k-colored graphs for DNA physical mapping. *Discrete Applied Mathematics*, 71(1-3):55–77, 1996. A preliminary version appeared in ICALP 1995. doi:10.1016/S0166-218X(96)00057-1.

[6] Pablo Burzyn, Flavia Bonomo, and Guillermo Durán. NP-completeness results for edge modification problems. *Discrete Applied Mathematics*, 154(13):1824–1844, 2006. doi:10.1016/j.dam.2006.03.031.

[7] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters*, 58(4):171–176, 1996. doi:10.1016/0020-0190(96)00050-6.

[8] Yixin Cao. Linear recognition of almost interval graphs. In Krauthgamer [20], pages 1096–1115. Full version available at arXiv:1403.1515. doi:10.1137/1.9781611974331.ch77.

[9] Yixin Cao, Luciano N. Grippo, and Martín D. Safe. Forbidden induced subgraphs of normal Helly circular-arc graphs: Characterization and detection. *Discrete Applied Mathematics*, 216:67–83, 2017. doi:10.1016/j.dam.2015.08.023.

[10] Yixin Cao and Dániel Marx. Chordal editing is fixed-parameter tractable. *Algorithmica*, 75(1):118–137, 2016. A preliminary version appeared in STACS 2014. doi:10.1007/s00453-015-0014-x.

[11] Xiaotie Deng, Pavol Hell, and Jing Huang. Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. *SIAM Journal on Computing*, 25(2):390–403, 1996. doi:10.1137/S0097539792269095.

[12] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Undergraduate texts in computer science. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

[13] Fedor V. Fomin, Saket Saurabh, and Yngve Villanger. A polynomial kernel for proper interval vertex deletion. *SIAM Journal on Discrete Mathematics*, 27(4):1964–1976, 2013. A preliminary version appeared in ESA 2012. doi:10.1137/12089051X.

[14] Delbert R. Fulkerson and Oliver A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15(3):835–855, 1965. doi:10.2140/pjm.1965.15.835.

[15] Fănică Gavril. Algorithms on circular-arc graphs. *Networks*, 4:357–369, 1974. doi:10.1002/net.3230040407.

[16] Paul W. Goldberg, Martin C. Golumbic, Haim Kaplan, and Ron Shamir. Four strikes against physical mapping of DNA. *Journal of Computational Biology*, 2(1):139–152, 1995. doi:10.1089/cmb.1995.2.139.

[17] Pim van 't Hof and Yngve Villanger. Proper interval vertex deletion. *Algorithmica*, 65(4):845–867, 2013. doi:10.1007/s00453-012-9661-3.

[18] Haim Kaplan, Ron Shamir, and Robert Endre Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. *SIAM Journal on Computing*, 28(5):1906–1922, 1999. A preliminary version appeared in FOCS 1994. doi:10.1137/S0097539796303044.

[19] Yuping Ke, Yixin Cao, Xiating Ouyang, and Jianxin Wang. Unit interval vertex deletion: Fewer vertices are relevant. arXiv:1607.01162, 2016.

[20] Robert Krauthgamer, editor. *Proceedings of the 27th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. SIAM, 2016. doi:10.1137/1.9781611974331.

[21] John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219–230, 1980. Preliminary versions independently presented in STOC 1978. doi:10.1016/0022-0000(80)90060-4.

[22] Min Chih Lin, Francisco J. Soulignac, and Jayme L. Szwarcfiter. Normal Helly circular-arc graphs and its subclasses. *Discrete Applied Mathematics*, 161(7-8):1037–1059, 2013. doi:10.1016/j.dam.2012.11.005.

[23] Yunlong Liu, Jianxin Wang, Jie You, Jianer Chen, and Yixin Cao. Edge deletion problems: Branching facilitated by modular decomposition. *Theoretical Computer Science*, 573:63–70, 2015. doi:10.1016/j.tcs.2015.01.049.

[24] Dániel Marx. Chordal deletion is fixed-parameter tractable. *Algorithmica*, 57(4):747–768, 2010. A preliminary version appeared in WG 2006. doi:10.1007/s00453-008-9233-8.

[25] Dániel Marx, Barry O'Sullivan, and Igor Razgon. Finding small separators in linear time via treewidth reduction. *ACM Transactions on Algorithms*, 9(4):30.1–30.35, 2013. A preliminary version appeared in STACS 2010. doi:10.1145/2500119.

[26] Fred S. Roberts. Indifference graphs. In Frank Harary, editor, *Proof Techniques in Graph Theory (Proc. Second Ann Arbor Graph Theory Conf., 1968)*, pages 139–146. Academic Press, New York, 1969.

[27] Alan C. Tucker. Structure theorems for some circular-arc graphs. *Discrete Mathematics*, 7(1-2):167–195, 1974. doi:10.1016/S0012-365X(74)80027-0.

[28] Yngve Villanger. Proper interval vertex deletion. In Venkatesh Raman and Saket Saurabh, editors, *Parameterized and Exact Computation - IPEC 2010*, volume 6478 of *LNCS*, pages 228–238. Springer, 2010. doi:10.1007/978-3-642-17493-3_22.

[29] Gerd Wegner. *Eigenschaften der Nerven homologisch-einfacher Familien im* $\mathbb{R}^n$. PhD thesis, Universität Göttingen, 1967.

[30] Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic and Discrete Methods*, 2(1):77–79, 1981. doi:10.1137/0602010.