Algorithmica (2013) 65:845–867
DOI 10.1007/s00453-012-9661-3

Proper Interval Vertex Deletion

Pim van 't Hof · Yngve Villanger

Received: 29 December 2010 / Accepted: 14 May 2012 / Published online: 30 May 2012

Abstract The NP-complete problem PROPER INTERVAL VERTEX DELETION is to decide whether an input graph on n vertices and m edges can be turned into a proper interval graph by deleting at most k vertices. Van Bevern et al. (In: Proceedings WG 2010. Lecture notes in computer science, vol. 6410, pp. 232–243, 2010) showed that this problem can be solved in $\mathcal{O}((14k+14)^{k+1}kn^6)$ time. We improve this result by presenting an $\mathcal{O}(6^k kn^6)$ time algorithm for PROPER INTERVAL VERTEX DELETION. Our fixed-parameter algorithm is based on a new structural result stating that every connected component of a {claw, net, tent, C_4, C_5, C_6}-free graph is a proper circular arc graph, combined with a simple greedy algorithm that solves PROPER INTERVAL VERTEX DELETION on {claw, net, tent, C_4, C_5, C_6}-free graphs in $\mathcal{O}(n+m)$ time. Our approach also yields a polynomial-time 6-approximation algorithm for the optimization variant of PROPER INTERVAL VERTEX DELETION.

Keywords Proper interval graphs · Proper circular arc graphs · Vertex deletion · FPT algorithms

1 Introduction

Many NP-hard problems can be solved in polynomial time on restricted graph classes. Classical examples are the polynomial-time algorithms for GRAPH COLORING, MAXIMUM CLIQUE and MAXIMUM INDEPENDENT SET on chordal graphs by Gavril [9] and on perfect graphs by Grötschel, Lovász and Schrijver [11], the

P. van 't Hof · Y. Villanger (✉)
Department of Informatics, University of Bergen, 5020 Bergen, Norway
e-mail: yngve.villanger@ii.uib.no

P. van 't Hof
e-mail: pim.vanthof@ii.uib.no

polynomial-time algorithm for computing the branchwidth of a planar graph by Seymour and Thomas [21], and Courcelle's Theorem for solving problems expressible in monadic second-order logic on graphs of bounded treewidth [7]. We refer to the book of Golumbic [10] for further references on algorithms for special graph classes.

Often, algorithms for solving problems on specific graph classes can be generalized to also work for graphs that are somehow close to these graph classes. The closeness of a graph G to a graph class Π can be defined in several ways, leading to different types of graph modification problems. Let us mention three types of graph modification problems here, covering many of the most natural problems in algorithmic graph theory. The Π-VERTEX DELETION problem takes as input a graph G and an integer k, and the task is to decide whether G can be transformed into a graph that belongs to class Π by deleting at most k vertices from G. The Π-EDGE DELETION problem is defined similarly, but now the question is whether a member of Π can be obtained by deleting at most k edges from G. The problem of deciding whether a graph in Π can be obtained by adding at most k edges to G is known as the Π-COMPLETION problem.

It is hardly surprising that the problems Π-VERTEX DELETION, Π-EDGE DELETION, and Π-COMPLETION are NP-hard for almost all natural graph classes Π. For example, Yannakakis [26] showed that Π-EDGE DELETION is NP-complete for many different classes Π, such as forests, bipartite graphs and outerplanar graphs. Lewis and Yannakakis [16] showed that Π-VERTEX DELETION is NP-complete for any nontrivial and hereditary graph class Π, where a graph class is called hereditary if every induced subgraph of every graph in the class also belongs to the same graph class. Arguably the most famous NP-complete example of Π-COMPLETION is the MINIMUM FILL-IN problem, which is the problem of determining the smallest number of edges whose addition makes the input graph chordal [27].

From a parameterized complexity perspective, the situation is much less clear. We are still far from understanding what properties make a graph modification problem fixed-parameter tractable (FPT), which means that there exists an algorithm that solves the problem on n-vertex input graphs in $f(k) \cdot n^c$ time, for some constant c and some function f that depends only on k. Examples of graph classes Π for which Π-VERTEX DELETION is known to be FPT are edgeless graphs [6], forests [4], chordal graphs [18], planar graphs [19] and proper interval graphs [23]. On the other hand, Π-VERTEX DELETION is known to be $W[2]$-hard when Π is the class of wheel-free graphs [17] or the class of perfect graphs [12], rendering it highly unlikely that an FPT algorithm exists for these classes. The Π-EDGE DELETION problem is known to be FPT when Π is the class of planar graphs [14], forests [5], or chordal graphs [18]. For the Π-COMPLETION problem, FPT algorithms are known for the case where Π is the class of chordal graphs [3, 13], interval graphs [24] or proper interval graphs [13].

All the graph classes mentioned above can be characterized by a (not necessarily finite) set of forbidden induced subgraphs. Cai [3] showed that the problems Π-VERTEX DELETION, Π-EDGE DELETION and Π-COMPLETION are all FPT for any hereditary graph class Π, as long as Π can be characterized by a finite number of forbidden induced subgraphs. The result of Cai leaves the parameterized complexity open for all hereditary graph classes that cannot be characterized by a finite number of forbidden induced subgraphs. A typical example, corresponding to the famous

FEEDBACK VERTEX SET problem, is the class of forests, where all the cycles are forbidden induced subgraphs.

In this paper, a *hole* is defined as an induced cycle of length at least 4, and thus chordal graphs are exactly the graphs that are hole-free. Despite this simple characterization, it was not until 2006 that Marx [18] proved the CHORDAL VERTEX DELETION problem to be FPT. Wegner [25] (see also Brandstädt et al. [2]) showed that proper interval graphs are exactly the class of graphs that are {claw, net, tent, hole}-free, where the claw, the net, and the tent are three well-known graphs on at most 6 vertices (see Fig. 1). Proper interval graphs are hereditary, since each induced subgraph of a {claw, net, tent, hole}-free graph is also {claw, net, tent, hole}-free. Hence, by combining the aforementioned results of Wegner, Cai, and Marx, it can be shown that PROPER INTERVAL VERTEX DELETION is FPT. Recently, van Bevern et al. [23] presented an algorithm for PROPER INTERVAL VERTEX DELETION using the structure of a problem instance that is already {claw, net, tent, C_4, C_5, C_6}-free. Using this approach, they obtained an algorithm with running time $\mathcal{O}((14k + 14)^{k+1}kn^6)$. Furthermore, they showed that the PROPER INTERVAL VERTEX DELETION problem remains NP-complete when restricted to {claw, net, tent}-free input graphs.

Our Results We present a simpler and faster algorithm for the PROPER INTERVAL VERTEX DELETION problem than the one presented by van Bevern et al. [23]. Our algorithm, which runs in time $\mathcal{O}(6^k kn^6)$, is based on a new combinatorial result of independent interest, stating that every connected component of a {claw, net, tent, C_4, C_5, C_6}-free graph is a proper circular arc graph. Using the structure of proper circular arc graphs, we show that PROPER INTERVAL VERTEX DELETION can be solved on {claw, net, tent, C_4, C_5, C_6}-free graphs in linear time. Our FPT algorithm for PROPER INTERVAL VERTEX DELETION on general graphs is obtained by combining this linear-time algorithm with a branching algorithm that produces a family of {claw, net, tent, C_4, C_5, C_6}-free problem instances. Like the algorithm by Villanger et al. [24] for INTERVAL COMPLETION, this is a branching algorithm where the leaves of the search tree correspond to polynomial-time solvable problem instances, rather than potential solutions. To complement the study of modifying a graph into a proper interval graph by deleting vertices, we also give a polynomial-time 6-approximation algorithm for the optimization variant of PROPER INTERVAL VERTEX DELETION.

2 Preliminaries

All graphs considered in this text are simple and undirected. For a graph $G = (V, E)$, we use n to denote the number of vertices and m to denote the number of edges in G. Two vertices $u, v \in V$ are *adjacent* if $\{u, v\} \in E$. The *neighborhood* of a vertex u in G is denoted by $N_G(u)$, and a vertex $v \in N_G(u)$ if $\{u, v\} \in E$. The *closed* neighborhood of u is the set $N_G[u] = N_G(u) \cup \{u\}$. Two vertices $u, v \in V$ are *twins* if $N_G[u] = N_G[v]$. Let $S \subseteq V$. We write $G[S]$ to denote the subgraph of G *induced* by S, i.e., $G[S] := (S, \{\{u, v\} \in E \mid u, v \in S\})$. The graph $G[V \setminus S]$ is denoted by $G - S$, and we write $G - v$ instead of $G - \{v\}$ for a vertex $v \in V$.

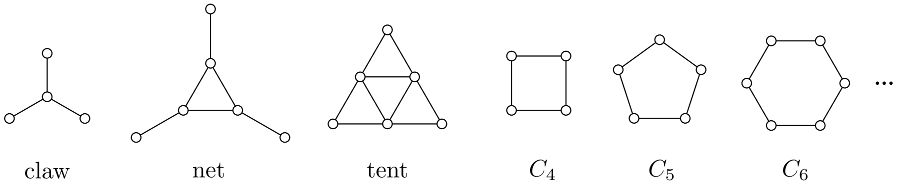


Fig. 1 The (infinite) family of forbidden induced subgraphs of proper interval graphs: the claw, the net, the tent, and all the holes, i.e., all induced cycles of length at least 4

A *path* P in G is a subgraph of G whose vertices can be ordered $v_1, v_2, \dots, v_r$ such that $\{v_i, v_{i+1}\} \in E$ for $1 \le i < r$. A path P is called *induced* if $\{v_i, v_j\} \notin E$ whenever $i + 1 < j$. If $\{v_1, v_r\} \in E$, then P is a *cycle* in G, and this cycle is *induced* if 1 and r are the only values of i and j for which $i + 1 < j$ and $\{v_i, v_j\} \in E$. A *chord* of a path (cycle) is an edge of G between two vertices of the path (cycle) that are not consecutive on the path (cycle). The *length* of a path (cycle) is the number of edges in that path (cycle). A *hole* is an induced cycle of length at least 4. We use C_r to denote a hole of length $r \ge 4$. A graph is *hole*-free if it does not contain a hole as an induced subgraph. More generally, we say that a graph G is F*-free* for some graph F if G does not contain an induced subgraph isomorphic to F. For any family $\mathcal{F}$ of graphs, we say that G is $\mathcal{F}$*-free* if G is F-free for every $F \in \mathcal{F}$. If a graph class $\mathcal{G}$ is $\mathcal{F}$-free, then the graphs in $\mathcal{F}$ are called the *forbidden induced subgraphs* of $\mathcal{G}$.

A graph G is an *interval* graph if it is the intersection graph of intervals on the real line, i.e., if each vertex of G can be assigned an interval on the real line, such that two vertices are adjacent in G if and only if their corresponding intervals intersect. The collection of intervals $\mathcal{I}$ is called an *interval model* of G. A *proper interval graph*, also known as a *unit interval graph* or an *indifference graph*, is an interval graph that has an interval model $\mathcal{I}$ where no interval is properly contained in another interval, i.e., where no interval is a subinterval of another interval. Alternatively, a graph is a proper interval graph if and only if it is {claw, net, tent, hole}-free [25]; see Fig. 1 for a depiction of the forbidden induced subgraphs of proper interval graphs.

A graph G is a *circular arc graph* if it is the intersection graph of arcs on a circle, i.e., if each vertex of G can be assigned an interval (arc) on a circle such that two vertices are adjacent in G if and only if their corresponding intervals intersect. Like for interval graphs, the intervals in $\mathcal{I}$ constitute a *(circular arc) model* of G. A circular arc graph is a *proper circular arc graph* if it has a model $\mathcal{I}$ where no interval of the model is a subinterval of another interval. An example of a proper circular arc graph (the tent) and a corresponding proper circular arc model is given in Fig. 2. A circular arc graph is a *unit circular arc graph* if it has a circular arc model where all the arcs are of unit length. It is easy to see that every unit circular arc graph is a proper circular arc graph. The reverse, however, is not true: Tucker [22] showed that the tent (see Fig. 2) is not a unit circular arc graph. This is an interesting contrast to the well-known fact that the classes of proper interval graphs and unit interval graphs do coincide [20].

Let G be a proper circular arc graph and let $\mathcal{I}$ be a proper circular arc model of G. For any subset $\mathcal{X} \subseteq \mathcal{I}$ of intervals, we say that (the union of the intervals in) $\mathcal{X}$ *covers the circle* if every point of the circle in the model $\mathcal{I}$ is contained in at least one

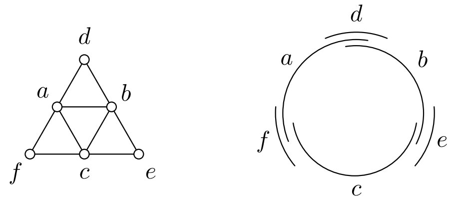


Fig. 2 A proper circular arc graph (the tent) on the left, with a corresponding proper circular arc model on the right. The three vertices of degree 2 in the tent form a so-called asteroidal triple, which implies that the tent is not an interval graph [15]

interval in $\mathcal{X}$. Recall that in any proper circular arc model, no interval is a subinterval of another interval. Consequently, we may assume that no single interval in $\mathcal{I}$ covers the circle. In fact, it is known that every proper circular arc graph has a proper circular arc model in which no pair of intervals covers the circle [10, 22]. Moreover, we may always assume that no two intervals in a proper circular arc model share an endpoint, i.e., that all $2n$ endpoints of intervals in $\mathcal{I}$ are distinct [22]. Throughout the paper, we will tacitly assume that all proper circular arc models we consider satisfy these properties.

Given a proper circular arc model $\mathcal{I}$ of a proper circular arc graph G, we refer to the *clockwise* direction along the circle as the *right* direction, whereas the *left* direction refers to the *counter-clockwise* direction. Each vertex v of G corresponds to an interval I_v in $\mathcal{I}$ whose leftmost point v^s is called its *start point*, and whose rightmost point v^e is called its *end point*; note that v^s and v^e are well-defined due to the assumption that no interval in $\mathcal{I}$ covers the circle. For convenience, we will label an interval I_v simply with v in any figure of a circular arc model in this paper.

For two points p_1 and p_2 on an interval I_v, we say that p_1 is *left* of p_2, denoted by $p_1 < p_2$, if p_2 is not a point on the subinterval of I_v from v^s to p_1. We write $p_1 \leq p_2$ if either $p_1 = p_2$ or p_1 is to the left of p_2. An interval I_v is *to the left* of an interval I_w if $v^s < w^s < v^e < w^e$, and in this case we say that I_w is *to the right* of I_v. Note that, for any two intersecting intervals $I_v, I_w \in \mathcal{I}$, either I_v is to the left of I_w or vice versa, since we assume that no two intervals in $\mathcal{I}$ cover the circle. For obvious reasons, we cannot define "to the left" and "to the right" if the intervals I_v and I_w do not intersect. For two adjacent vertices v and w in G, we say that v is *to the left* of w (and w is to the right of v) if the interval I_v is to the left of the interval I_w in $\mathcal{I}$. A vertex v is the *leftmost* neighbor of a vertex w if v is to the left of w, and no other neighbor of w is to the left of v. The *rightmost* neighbor of a vertex is defined analogously.

Let I_v and I_v be two intersecting intervals in $\mathcal{I}$, where I_v is to the left of I_w. We define the *union* of I_v and I_w to be the interval with start point v^s and end point w^e. Let $X \subseteq V$ be a set such that $G[X]$ is connected. Then there exists a vertex ordering $v_1, v_2, \ldots, v_{|X|}$ of the vertices in X such that $G[\{v_1, v_2, \ldots, v_i\}]$ is connected for $1 \leq i \leq |X|$. Let $\mathcal{X} = \{I_v \mid v \in X\}$ be the set of intervals in $\mathcal{I}$ corresponding to the vertices in X. The *union* of the intervals in $\mathcal{X}$ is the interval $I_{1,|X|}$, where the interval $I_{1,i}$ is defined recursively as the union of the interval $I_{1,i-1}$ and the interval I_{v_i}, and where $I_{1,2}$ is the union of intervals I_{v_1} and I_{v_2}. Equivalently, the union of the intervals of $\mathcal{X}$ is the interval that contains exactly those points of the circle of model $\mathcal{I}$ that are contained in at least one interval in $\mathcal{X}$.

3 Almost Proper Interval Graphs

Adopting the terminology introduced by van Bevern et al. [23], we define an *almost proper interval graph* to be a {claw, net, tent, C_4, C_5, C_6}-free graph. The main result of this section is a theorem stating that every connected component of an almost proper interval graph is a proper circular arc graph. Before presenting this main result, we prove some structural properties of proper circular arc graphs and almost proper interval graphs.

Lemma 1 *Let G be a proper circular arc graph and let $\mathcal{I}$ be a proper circular arc model of G. If G contains an induced cycle of length $r \geq 4$, then $|\mathcal{X}| \geq \lceil r/2 \rceil$ for any subset $\mathcal{X} \subseteq \mathcal{I}$ of intervals whose union covers the circle.*

Proof Let us first recall that we may assume that all $2n$ start and end points of the intervals in $\mathcal{I}$ are distinct. Let ϵ be the smallest distance on the circle between any two of these $2n$ start and end points. We now construct a new proper circular arc model $\mathcal{I}'$ from $\mathcal{I}$ as follows. First, for each $I_v \in \mathcal{I}$, we add an identical interval $I_{v'}$ to the model; we say that each added interval $I_{v'}$ is *marked*. Then, for each pair of identical intervals I_v and $I_{v'}$, we shift the start and end points of the marked interval $I_{v'}$ by $\epsilon/2$ to the right. This finishes the construction of $\mathcal{I}'$. Note that, by construction, $\mathcal{I}'$ is a proper circular arc model, and that all $4n$ start and end points of the intervals in $\mathcal{I}'$ are distinct.

Let G' be the proper circular arc graph represented by the model $\mathcal{I}'$. We say that a vertex v' in G' is *marked* if its corresponding interval $I_{v'}$ is marked. Note that, for each vertex v of G, the vertices v and v' are twins in G', since the intervals I_v and $I_{v'}$ intersect exactly the same set of intervals in $\mathcal{I}'$.

Suppose G' contains an induced cycle C on $r \geq 4$ unmarked vertices, and let $\mathcal{Y}$ be the set of intervals in the model $\mathcal{I}'$ representing the vertices of C. Notice that the union of the intervals in $\mathcal{Y}$ covers the circle of the model $\mathcal{I}'$, as in any proper circular arc model, only induced cycles of length 3 can be represented by intervals whose union does not cover the circle. Let $\mathcal{X}'$ be a minimal subset of marked intervals in $\mathcal{I}'$ such that the union of the intervals in $\mathcal{X}'$ covers the circle. Due to the minimality of $\mathcal{X}'$, the set of vertices represented by the intervals in $\mathcal{X}'$ induces a cycle in G'.

Let the intervals of $\mathcal{Y}$ be numbered I_{v_1} to I_{v_r} in such a way that $I_{v_{i-1}}$ is to the left of I_{v_i} for $1 < i \leq r$, and I_{v_r} is to the left of I_{v_1}. In a similar way, let the intervals of $\mathcal{X}'$ be numbered I_{u_1} to $I_{u_{|\mathcal{X}'|}}$ in such a way that I_{u_1} is to the left of I_{v_1}, I_{v_1} is to the left of I_{u_2}, $I_{u_{i-1}}$ is to the left of interval I_{u_i} for $1 < i \leq |\mathcal{X}'|$, and $I_{u_{|\mathcal{X}'|}}$ is to the left of I_{u_1}. Notice that these two orderings exist since $\mathcal{Y}$ and $\mathcal{X}'$ both cover the circle, and there are no two intervals in $\mathcal{Y} \cup \mathcal{X}'$ such that one is a subinterval of another.

For any two points x and y, we will say that x is to the left of y, denoted by $x < y$, if x is to the left of y on the interval that is the union of the intervals in $\mathcal{X}' \setminus \{I_{u_{|\mathcal{X}'|}}\}$. Note that "to the left" is well-defined here, since the union of the intervals in $\mathcal{X}' \setminus \{I_{u_{|\mathcal{X}'|}}\}$ does not cover the circle due to the minimality of $\mathcal{X}'$. We now distinguish between two cases. □

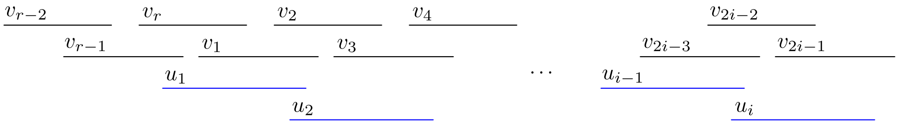


Fig. 3 Figure for Case 1: $v_2^s < u_1^e$. We prove that, in this case, $u_i^e < v_{2i-1}^e$ for $1 < i \leq r$

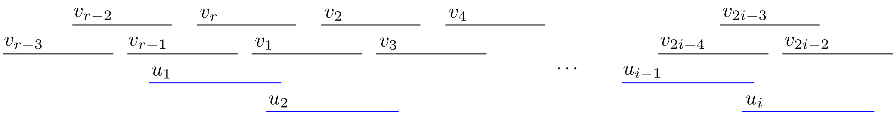


Fig. 4 Figure for Case 2: $u_1^e < v_2^s$. We prove that, in this case, $u_i^e < v_{2i-2}^e$ for $1 < i \leq r$

Case 1: $v_2^s < u_1^e$ We first prove that $u_i^e < v_{2i-1}^e$ for $1 < i \leq r$ by induction on i. Note that $u_2^e < v_3^e$, since the definition of $\mathcal{I}'$ implies that $u_2^s < u_1^e < v_1^e < v_3^s$, and I_{v_3} is not a subinterval of I_{u_2} (see Fig. 3). Hence the claim holds for $i = 2$. For the induction hypothesis, suppose that $u_{i-1}^e < v_{2i-3}^e$ for some $i \geq 2$. Then $u_i^s < u_{i-1}^e < v_{2i-3}^e < v_{2i-1}^s$. This, together with the observation that $I_{v_{2i-1}}$ is not a subinterval of I_{u_i}, implies that $u_i^e < v_{2i-1}^e$.

Since we are considering the case where $v_2^s < u_1^e$ and I_{v_r} is not a subinterval of I_{u_1}, we have $v_{r-2}^e < u_1^s$. By the above induction proof, we know that $u_{r-1}^e < v_{2r-4}^e$, and hence $u_{\lfloor (r-1)/2 \rfloor}^e < v_{r-2}^e$. Consequently, interval $I_{u_{\lfloor (r-1)/2 \rfloor + 1}}$ is required to cover point v_{r-2}^e, which implies that $|\mathcal{X}'| \geq \lfloor (r-1)/2 \rfloor + 1 = \lceil r/2 \rceil$.

Case 2: $u_1^e < v_2^s$ Since $u_1^e < v_2^s$, we know that $u_2^s < u_1^e < v_2^s < u_2^e < v_2^e$, where the last inequality follows from the fact that I_{v_2} is not a subinterval of I_{u_2} (see Fig. 4). Using similar arguments as in Case 1, we can use induction on i to prove that $u_i^e < v_{2i-2}^e$ for $1 < i \leq r$ as follows. We already observed that the base case $u_2^e < v_2^e$ holds. Assume that $u_{i-1}^e < v_{2i-4}^e$ for some $i \geq 2$. Then $u_i^s < u_{i-1}^e < v_{2i-4}^e < v_{2i-2}^s$. Since $I_{v_{2i-2}}$ is not a subinterval of I_{u_i}, we get $u_i^e < v_{2i-2}^e$.

The assumption $u_1^e < v_2^s$ implies that $v_{r-3}^e < u_1^s$, since otherwise $I_{v_{r-1}}$ would be a subinterval of I_{u_1}. It follows from the above induction proof that $u_{\lfloor (r-1)/2 \rfloor}^e < v_{r-3}^e$, which means that the interval $I_{\lfloor (r-1)/2 \rfloor + 1}$ is required to cover the point v_{r-3}^e. Just as in Case 1, we conclude that $|\mathcal{X}'| \geq \lfloor (r-1)/2 \rfloor + 1 = \lceil r/2 \rceil$. □

The following result is due to Brandstädt and Dragan [1].

Proposition 1 ([1]) *Let C be an induced cycle of length at least* 5 *in a* {*claw*, *net*}*-free graph G of diameter greater than* 3. *Then every vertex in $V(G) \setminus V(C)$ is adjacent exactly to two*, *three or four consecutive vertices of C*.

The following result, which will be used in the proofs of some of the lemmas below, easily follows from Proposition 1. Note that the class of {claw, net, C_4}-free graphs is a superclass of the class of almost proper interval graphs.

Corollary 1 *Let C be an induced cycle of length at least* 6 *in a* $\{claw, net, C_4\}$*-free graph G. Then every vertex in $V(G) \setminus V(C)$ is adjacent exactly to two, three or four consecutive vertices of C.*

Proof Brandstädt and Dragan [1] proved that a {claw, net}-free graph of diameter greater than 3 cannot contain the C_5, the tent, or the graph S_3^- as an induced subgraph, where S_3^- is the graph obtained from the tent by removing one edge between two vertices of degree 4 (such as the edge *ab* in Fig. 2). Moreover, it follows from the proof of Proposition 1 in [1] that we can replace "in a {claw, net}-free graph G of diameter greater than 3" in Proposition 1 by "in a $\{\text{claw}, \text{net}, S_3^-\}$-free graph G". Since the graph S_3^- contains a C_4 as an induced subgraph, this suffices to prove Corollary 1. □

Recall that an almost proper interval graph is a $\{\text{claw}, \text{net}, \text{tent}, C_4, C_5, C_6\}$-free graph. In Theorem 1 below, we prove that every connected almost proper interval graph is a proper circular arc graph. The proof of Theorem 1 is constructive: given a connected almost proper interval graph G, a proper circular arc model of G is constructed in an incremental way as follows. The algorithm starts with an induced subgraph of G that is an induced cycle of length at least 7. (Note that if G does not contain such a cycle, G is a proper interval graph, and hence a proper circular arc graph.) At the start of the k-th iteration, the algorithm considers an induced subgraph G_{k-1} of G that is a connected proper circular arc graph that contains an induced cycle of length at least 7. A new vertex v_k of G is added to the graph G_{k-1} in such a way that the obtained subgraph G_k of G is connected. The algorithm then constructs a proper circular arc model of G_k from the proper circular arc model of G_{k-1}, and proceeds to the next iteration. The algorithm continues until a proper circular arc model of G has been constructed.

The crucial step in each iteration of the algorithm is the construction of a proper circular arc model of G_k from the proper circular arc model of G_{k-1}. In order to prove that we can always do this, we need a few structural lemmas. In the three lemmas below, we consider the case where G_k is a connected almost proper interval graph that has a vertex $v := v_k$, such that the graph $G_{k-1} = G_k - v$ is a proper circular arc graph that contains a cycle $C = w_1, w_2, \dots, w_r$ for some $r \geq 7$. Let $\mathcal{I}_{k-1}$ be a proper circular arc model of G_{k-1}.

The first lemma below shows that, due to the presence of the long cycle C in G_{k-1}, no subset of six or less intervals in $\mathcal{I}_{k-1}$ can cover the circle.

Lemma 2 *For any subset $\mathcal{X} \subseteq \mathcal{I}_{k-1}$ of intervals whose union covers the circle of model $\mathcal{I}_{k-1}$, we have that $|\mathcal{X}| \geq 7$.*

Proof Let $\mathcal{X} \subseteq \mathcal{I}$ be a set of intervals whose union covers the circle. Since $\mathcal{X}$ covers the circle, there is a subset $\mathcal{X}' \subseteq \mathcal{X}$ such that $\mathcal{X}'$ covers the circle and the vertices of G_{k-1} corresponding to the intervals in $\mathcal{X}'$ induce a cycle in G_{k-1}. By Lemma 1, any subset of intervals in $\mathcal{I}_{k-1}$ that covers the circle must contain at least 4 intervals, so $|\mathcal{X}'| \geq 4$. In fact, since G_{k-1} is $\{C_4, C_5, C_6\}$-free, we know that $|\mathcal{X}'| \geq 7$. Since $|\mathcal{X}| \geq |\mathcal{X}'|$, the lemma follows. □

The following lemma states that there are two consecutive vertices on cycle C such that v is adjacent to both of them, and every other neighbor of v in G_k is adjacent to at least one of them.

Lemma 3 *There exist two consecutive vertices w_t, w_{t+1} on cycle C such that $w_t, w_{t+1} \in N_{G_k}(v)$ and $N_{G_k}(v) \subseteq N_{G_k}(w_t) \cup N_{G_k}(w_{t+1})$.*

Proof Since G_k is an induced subgraph of an almost proper interval graph, G_k is $\{\text{claw}, \text{net}, \text{tent}, C_4, C_5, C_6\}$-free, and in particular $\{\text{claw}, \text{net}, C_4\}$-free. Hence, by Corollary 1, the neighbors of v in G_k on cycle C are $w_i, w_{i+1}, \dots, w_j$, where $j \le i+3$. Let us first argue that every vertex $x \in N_{G_k}(v) \setminus \{w_i, w_{i+1}, \dots, w_j\}$ has a neighbor w_q such that $i \le q \le j$. For contradiction, let us assume that x is a vertex of $N_{G_k}(v) \setminus \{w_i, w_{i+1}, \dots, w_j\}$ that does not have such a neighbor. Then $j = i+1$, since otherwise $G_k[\{x, u, w_i, w_j\}]$ would be a claw. If $j = i+1$ and x is adjacent to neither w_{i-1} nor w_{i+2}, then $G_k[\{w_{i-1}, w_i, w_{i+1}, w_{i+2}, u, x\}]$ is a net. On the other hand, if x is adjacent to w_{i-1} or w_{i+2}, then $G_k[\{x, u, w_i, w_{i-1}\}]$ or $G_k[\{x, u, w_{i+1}, w_{i+2}\}]$ is a C_4, respectively. In each case, we obtain the desired contradiction.

Now let us assume, for contradiction, that there exists no integer t with $i \le t < j$ such that $N_{G_k}(v) \subseteq N_{G_k}(w_t) \cup N_{G_k}(w_{t+1})$. By Corollary 1, the neighbors of any $x \in N_{G_k}(v) \setminus V(C)$ on cycle C are consecutive on the cycle, and by the argument above, x has at least one neighbor in $\{w_i, w_{i+1}, \dots, w_j\}$. Let i' and j' be the largest and smallest integers, respectively, such that $i \le i' \le j' \le j$ and $N_{G_k}(v) \subseteq \bigcup_{i' \le \tau \le j'} N_{G_k}(w_\tau)$. Let $x, y \in N_{G_k}(v)$ be vertices such that $N_{G_k}(x) \cap \{w_{i'}, w_{i'+1}, \dots, w_{j'}\} = \{w_{i'}\}$ and $N_{G_k}(y) \cap \{w_{i'}, w_{i'+1}, \dots, w_{j'}\} = \{w_{j'}\}$; note that such vertices x and y exist by the definition of i' and j'. Recall that we assumed that there is no $i \le t < j$ such that $N_{G_k}(v) \subseteq N_{G_k}(w_t) \cup N_{G_k}(w_{t+1})$, which means that $i' + 1 < j'$. Hence, vertices x and y are not adjacent, since otherwise $G_k[\{x, w_{i'}, w_{i'+1}, \dots, w_{j'}, y\}]$ would be a C_5 or a C_6. But then we obtain a contradiction, since $G_k[\{u, x, w_{i'+1}, y\}]$ is a claw. □

Before presenting the main result of this section, we prove one additional lemma that describes two useful properties of the neighborhood of v in G_k.

Lemma 4 *The graph $G_k[N_{G_k}(v)]$ is connected, and the union of the intervals representing the vertices in $N_{G_k}(v)$ does not cover the circle of model $\mathcal{I}_{k-1}$.*

Proof By Lemma 3, there exist two consecutive vertices $w_t, w_{t+1} \in N_{G_k}(v)$ on cycle C such that $N_{G_k}(v)$ is contained in $N_{G_k}(w_t) \cup N_{G_k}(w_{t+1})$. Since the vertices w_t and w_{t+1} are adjacent and $N_{G_k}(v) \subseteq N_{G_k}(w_t) \cup N_{G_k}(w_{t+1})$, the graph $G_k[N_{G_k}(v)]$ is connected.

Consider now the model $\mathcal{I}_{k-1}$ of the graph G_{k-1}. Let z_1 be the vertex in $N_{G_k}(v)$ whose interval ends in the union of I_{w_t} and $I_{w_{t+1}}$ and starts furthest to the left, and let z_2 be the vertex in $N_{G_k}(v)$ whose interval starts in the union of I_{w_t} and $I_{w_{t+1}}$ and ends furthest to the right; note that it is possible that $z_1 = w_t$ or $z_2 = w_{t+1}$, which implies that such vertices z_1 and z_2 exist. Since $\mathcal{I}_{k-1}$ is a proper circular arc model, we have $w^s_{t-2} < z^s_1 \le w^s_t < z^e_1$ and $z^s_2 < w^e_{t+1} \le z^e_2 < w^e_{t+3}$ (see Fig. 5 for an illustration).

Let I^* be the union of the intervals $I_{z_1}, I_{w_t}, I_{w_{t+1}}, I_{z_2}$, i.e., I^* is the interval with start point z^s_1 and end point z^e_2. Since I^* is the union of four intervals, we know that I^*

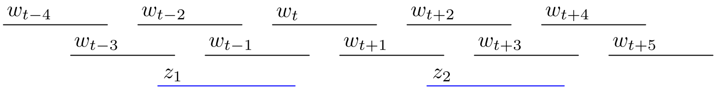


Fig. 5 Illustration for the proof of Lemma 4

does not cover the circle of model $\mathcal{I}_{k-1}$ due to Lemma 2. Moreover, by the definition of z_1 and z_2 and the fact that every neighbor of v in G_k is adjacent to w_t or w_{t+1}, we know that I^* contains I_x as a subinterval for each neighbor $x \in N_{G_k}(v)$; in fact, I^* is precisely the union of the intervals representing the vertices in $N_{G_k}(v)$. This completes the proof of Lemma 4. □

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

Theorem 1 *Every connected almost proper interval graph is a proper circular arc graph.*

Proof Let $G = (V, E)$ be a connected almost proper interval graph. If G does not contain an induced cycle of length at least 7, i.e., if G is hole-free, then G is a proper interval graph due to the aforementioned result by Wegner [25]. Since every proper interval graph is a proper circular arc graph, we are done in this case.

Suppose G contains an induced cycle C of length $r \geq 7$. Let $v_1, v_2, \dots, v_r$ be an ordering of the vertices of C and let $v_{r+1}, v_{r+2}, \dots, v_n$ be an ordering of the vertices in $V \setminus V(C)$ such that the graph $G_k := G[\{v_1, v_2, \dots, v_k\}]$ is connected for $1 \leq k \leq n$. We prove, by induction on k, that G_k is a proper circular arc graph for $1 \leq k \leq n$.

For the base case, let us observe that the cycle C is a proper circular arc graph. In fact, due to the choice of the ordering $v_1, \dots, v_r$, it is easy to see that the graph G_k is a proper circular arc graph for $1 \leq k \leq r$. Moreover, we can easily construct a proper circular arc model $\mathcal{I}_r$ for the graph G_r. For the induction hypothesis, assume that G_{k-1} is a proper circular arc graph for some $k \geq r + 1$, and let $\mathcal{I}_{k-1}$ be a proper circular arc model of G_{k-1}. Recall that v_k is the vertex that is added to G_{k-1} in order to obtain G_k. We will show that G_k is a proper circular arc graph as follows. We first add an interval I_{v_k} to the model $\mathcal{I}_{k-1}$ (in the way described below) to obtain a new model $\mathcal{I}^*$. We then modify $\mathcal{I}^*$, if necessary, such that it becomes a proper circular arc model of the graph G_k.

Let us first explain how the interval I_{v_k} is added to the proper circular arc model $\mathcal{I}_{k-1}$ in order to obtain the (not necessarily proper) circular arc model $\mathcal{I}^*$. By Lemma 3, $N_{G_k}(v_k)$ contains two consecutive vertices w_t, w_{t+1} of cycle C, such that all vertices of $N_{G_k}(v_k)$ are contained in $N_{G_k}(w_t) \cup N_{G_k}(w_{t+1})$. Let z_1 be the vertex in $N_{G_k}(v_k)$ with the leftmost end point z_1^e ($z_1 \in N_{G_k}[w_t]$), and let z_2 be the vertex in $N_{G_k}(v_k)$ with the rightmost start point z_2^s ($z_2 \in N_{G_k}[w_{t+1}]$). Vertices z_1 and z_2 are well-defined as a result of Lemma 4. Notice that $z_1 \neq z_2$, as both w_t and w_{t+1} are contained in $N_{G_k}(v_k)$ by definition. The interval I_{v_k} is now added to the model $\mathcal{I}_{k-1}$, where the positions of the start point v_k^s and the end point v_k^e of I_{v_k} depend on whether or not z_1 and z_2 are adjacent in the following way (see Fig. 6 for an illustration):

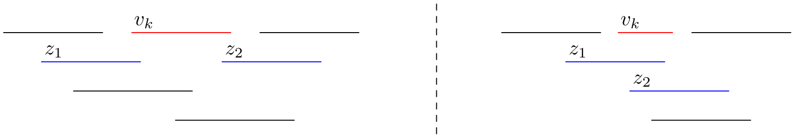


Fig. 6 Illustration of the way interval I_{v_k} is added to proper circular arc model $\mathcal{I}_{k-1}$ in order to obtain model $\mathcal{I}^*$, in case $\{z_1, z_2\} \notin E$ (*left figure*) or $\{z_1, z_2\} \in E$ (*right figure*)

- If $\{z_1, z_2\} \notin E$, then v_k^s is placed immediately to the left of z_1^e and v_k^e is placed immediately to the right of z_2^s, such that there exists no start or end point p_1 in model $\mathcal{I}_{k-1}$ where $v_k^s < p_1 < z_1^e$ and there exists no start or end point p_2 where $z_2^s < p_2 < v_k^e$.
- If $\{z_1, z_2\} \in E$, then v_k^s is placed immediately to the left of z_2^s and v_k^e is placed immediately to the right of z_1^e, such that there exists no start or end point p_1 in model $\mathcal{I}_{k-1}$ where $v_k^s < p_1 < z_2^s$ and there exists no start or end point p_2 where $z_1^e < p_2 < v_k^e$.

Let $\mathcal{I}^*$ be the model we obtain by adding interval I_{v_k} to $\mathcal{I}_{k-1}$ in the way described above. It is clear that $\mathcal{I}^*$ is a circular arc model. However, $\mathcal{I}^*$ is not necessarily a *proper* circular arc model, as there might be an interval I_x in $\mathcal{I}^*$ such that I_x is a subinterval of I_{v_k}, or vice versa. If such a vertex x exists, we call it an *obstruction vertex of type 1*. Moreover, it is not guaranteed that $\mathcal{I}$ is a circular arc model of the graph G_k, as there might be a vertex x such that $\{x, v_k\} \notin E(G_k)$ and I_x intersects I_{v_k}. Such a vertex x is called an *obstruction vertex of type 2*. Note that a vertex x can be both an obstruction vertex of type 1 and an obstruction vertex of type 2 at the same time.

If no obstruction vertex of type 1 or 2 exists, then $\mathcal{I}^*$ is a proper circular arc model for G_k, and thus G_k is a proper circular arc graph. In that case, we set $\mathcal{I}_k := \mathcal{I}^*$, and we proceed to the next subgraph G_{k+1} (unless $k = n$, in which case we are done). If there exists an obstruction vertex of type 1 or 2, then we will show below how the model $\mathcal{I}^*$ can be modified in such a way that the number of obstruction vertices strictly decreases, eventually leading to a proper circular arc model $\mathcal{I}_k$ of G_k. It is clear that this suffices to prove Theorem 1.

Suppose there exists an obstruction vertex x of type 1 or 2. As usual, we write I_x to denote the interval in $\mathcal{I}^*$ that corresponds to x, and x^s and x^e to denote the start and end points of I_x, respectively. Let us consider all possible positions of the points x^s and x^e in the model $\mathcal{I}^*$ with respect to the positions of the points z_1^e and z_2^s. There are 24 permutations of the four points x^s, x^e, z_1^e, z_2^s. Since I_x is an interval in the model $\mathcal{I}_{k-1}$, we know that $x^s < x^e$. Hence 12 permutations of the four points x^s, x^e, z_1^e, z_2^s remain, each satisfying $x^s < x^e$. Consider the cases where x^s and x^e are both to the left of z_1^s or both to the right of z_2^e. Since in those cases I_x and I_{v_k} do not intersect, x is not an obstruction vertex of type 1. Moreover, since we also know that $x \notin N_{G_k}(v_k)$ by the definition of z_1 and z_2, x is not an obstruction vertex of type 2 either. Since we assumed x to be an obstruction vertex, we can therefore safely ignore the permutations where $x^s < x^e < z_1^e < z_2^s$, $x^s < x^e < z_2^s < z_1^e$, $z_1^e < z_2^s < x^s < x^e$, or $z_2^s < z_1^e < x^s < x^e$. The eight remaining permutations are listed and illustrated in Fig. 7.

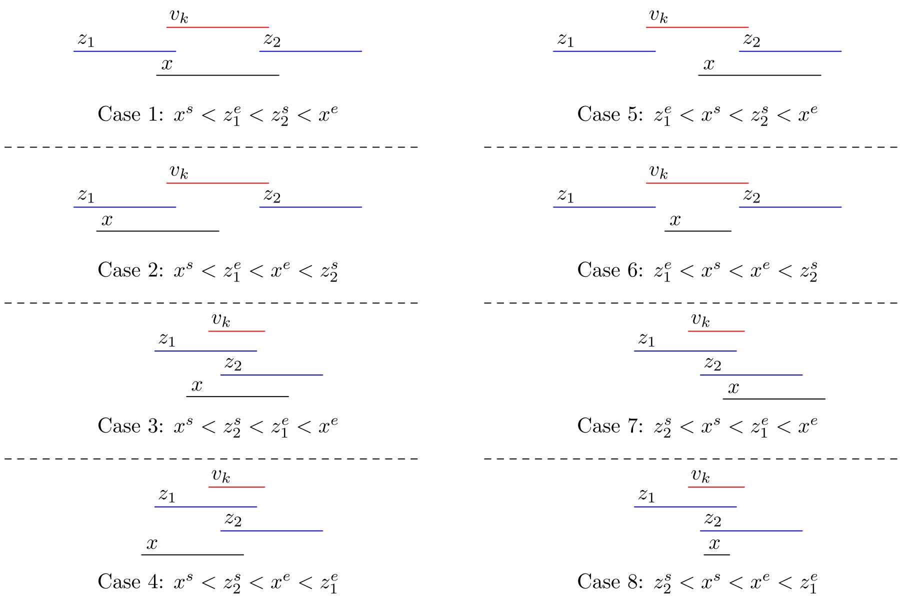


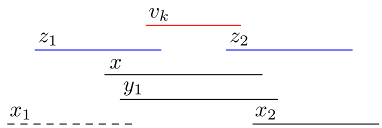
Fig. 7 Eight possible ways in which the points x^s and x^e of an obstruction vertex x can appear in model $\mathcal{I}^*$ with respect to the points z_1^e and z_2^s. (In fact, we prove that Cases 4, 6, 7, and 8 do not occur in $\mathcal{I}^*$)

In the remainder of this proof, we will consider each of these eight cases separately. In fact, for each of the cases, we will consider two subcases, indicated with "a" and "b", depending on the existence of the edge $\{x, v_k\}$ in G_k. For example, Case 1a is the case where $x^s < z_1^e < z_2^s < x^e$ and $x \notin N_{G_k}(v_k)$, whereas Case 1b is the case where $x^s < z_1^e < z_2^s < x^e$ and $x \in N_{G_k}(v_k)$. Note that Case 2 is symmetric to Case 5, and Case 4 is symmetric to Case 7.

We will use two different proof strategies in the case analysis below. For each of the Cases 4, 6, 7, and 8 (and therefore for both of their subcases), as well as for the Cases 1a, 2b, 3a, and 5b, we prove that they cannot occur in $\mathcal{I}^*$, as this would contradict the assumption that G_k is {claw, net, tent, C_4, C_5, C_6}-free. For the Cases 1b and 3b, we show how we can manipulate model $\mathcal{I}^*$ in such a way that the number of obstruction vertices strictly decreases. In Case 2a (and by symmetry also in Case 5a), we can either manipulate the model and strictly reduce the number of obstruction vertices, or obtain a contradiction, depending on whether or not x and z_1 (respectively z_2) are twins in G_{k-1}. Note that once we have successfully decreased the number of obstruction vertices to 0, we have obtained a proper circular arc model $\mathcal{I}_k$ of G_k.

Before we start with the case analysis, let us make some observations. First, due to Lemma 4, the intervals I_{z_1} and I_{z_2} can only intersect if $z_2^s < z_1^e$. Furthermore, due to Lemma 2, we need the union of at least 7 intervals in order to cover the circle of model $\mathcal{I}^*$. This means that we can interpret a section of a proper circular arc model as a proper interval model if the section contains a subset of at most six intervals that cover the entire section. Since this is the case for each of the cases we discuss below (as is apparent from the corresponding figures), this observation justifies the horizontal drawing of the section of model $\mathcal{I}^*$ that we consider in each case. Finally, in the anal-

Fig. 8 Illustration for Case 1b



ysis below, one should keep in mind that the graph G_k is {claw, net, tent, C_4, C_5, C_6}-free, as it is an induced subgraph of the {claw, net, tent, C_4, C_5, C_6}-free graph G.

Case 1a: $x^s < z_1^e < z_2^s < x^e$, $x \notin N_{G_k}(v_k)$ Vertices z_1 and z_2 are not adjacent, but both of them are adjacent to x and to v_k. Hence $G_k[\{z_1, x, z_2, v_k\}]$ is a C_4. This contradiction implies that Case 1a cannot occur in $\mathcal{I}^*$.

Case 1b: $x^s < z_1^e < z_2^s < x^e$, $x \in N_{G_k}(v_k)$ Suppose x has neighbors x_1 and x_2 such that $x_1^s < x^s < x_1^e < z_1^e < z_2^s < x_2^s < x^e < x_2^e$ (see Fig. 8). Then $x_1, x_2 \notin N_{G_k}(v_k)$ by the definition of z_1 and z_2. Moreover, the vertices x_1 and x_2 are not adjacent, since $x_1^e < x_2^s$. But then $G[\{x_1, x, x_2, v_k\}]$ is a claw, yielding a contradiction.

Now suppose that x does not have such neighbors x_1 and x_2, which implies that z_1 is the leftmost neighbor of x or z_2 is the rightmost. Assume that z_1 is the leftmost neighbor of x (see Fig. 8); the case where z_2 is the rightmost neighbor of x can be dealt with analogously. The assumption that z_1 is the leftmost neighbor of x implies that, for every point p that is a start or end point of an interval in model $\mathcal{I}^*$ such that $x^s < p < v_k^s$, p is a start point.

Let $Y := \{y \in V(G_k) \mid x^s \leq y^s < v_k^s\}$. Note that Y is non-empty, since $x \in Y$. Since we are considering the case where $x^s < z_1^e < z_2^s < x^e$, x is an obstruction vertex of type 1. We claim that the same holds for every $y \in Y \setminus \{x\}$. Let $y \in Y \setminus \{x\}$. Since y is to the right of x in the proper circular arc model $\mathcal{I}_{k-1}$, we must have $x^e < y^e$, and hence $v_k^e < y^e$. This implies that I_{v_k} is a subinterval of I_y, so y is an obstruction vertex of type 1 by definition. We also claim that none of the vertices in Y is an obstruction vertex of type 2. For x, this follows from the fact that I_x and I_{v_k} intersect in model $\mathcal{I}^*$, together with the assumption that $x \in N_{G_k}(v_k)$. Let $y \in Y \setminus \{x\}$. Notice that $y \in N_{G_k}(v_k)$, since otherwise $G_k[\{x, y, z_1, z_2\}]$ would be a C_4 (see also Case 1a). Since I_y and I_x intersect, y is not an obstruction vertex of type 2.

We will now explain how we can modify model $\mathcal{I}^*$ in such a way that the number of obstruction vertices strictly decreases. We extend interval I_{v_k} by moving the point v_k^s to the immediate left of x^s, making sure that there is no start or end point p of any interval in $\mathcal{I}^*$ with $v_k^s < p < x^s$. Note that after extending I_{v_k} this way, none of the vertices in Y is an obstruction vertex of type 1 anymore, since we already saw that $x^e < y^e$ for every $y \in Y$, implying that I_y did not become a subinterval of the extended interval I_{v_k}. Also note that no new obstruction vertices of type 1 or type 2 are created, due to the assumption that no interval in $\mathcal{I}^*$ has an end point p with $x^s < p < v_k^s$. Since Y is non-empty, the number of obstruction vertices strictly decreased.

Case 2a: $x^s < z_1^e < x^e < z_2^s$, $x \notin N_{G_k}(v_k)$ Let us first argue why x is not an obstruction vertex of type 1. Recall that the points v_k^s and v_k^e were placed in model $\mathcal{I}_{k-1}$ immediately to the left of z_1^e and immediately to the right of z_2^s, respectively. This, together with the assumption $x^s < z_1^e < x^e < z_2^s$, implies that $x^s < v_k^s < z_1^e < x^e < z_2^s < v_k^e$ in this case. Hence I_x is not a subinterval of I_{v_k} or vice versa, which means that x is not an obstruction vertex of type 1.

On the other hand, x is an obstruction vertex of type 2, as $x \notin N_{G_k}(v_k)$ and the intervals I_x and I_{v_k} intersect in model $\mathcal{I}_{k-1}$. We distinguish between two cases, depending on whether or not x is a twin of z_1 in the graph G_{k-1}.

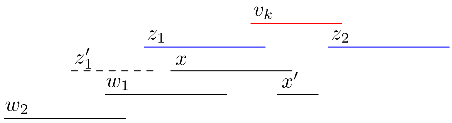
First suppose that x and z_1 are twins in G_{k-1}, i.e., $N_{G_{k-1}}[x] = N_{G_{k-1}}[z_1]$. We swap the intervals assigned to x and z_1, i.e., we relabel interval I_x as I_{z_1} and relabel interval I_{z_1} as I_x. Let $\mathcal{I}'$ denote the model obtained after the swap. Note that the swap did not change any of the adjacencies in the graph G_{k-1}, as x and z_1 are twins. However, it is possible that z_1 is no longer the leftmost neighbor of v_k in model $\mathcal{I}'$, as there might exist a neighbor z' of v_k that was to the left of x and to the right of z_1 in model $\mathcal{I}_{k-1}$, which means that z' is to the left of z_1 in model $\mathcal{I}'$. Hence, we redefine z_1 to be the leftmost neighbor of v_k in model $\mathcal{I}'$. Note that the interval representing the new vertex z_1 is to the right of the interval that was labeled I_{z_1} in model $\mathcal{I}_{k-1}$, since that interval corresponds to vertex x in model $\mathcal{I}'$, and x is not adjacent to v_k by assumption. Recall that, when interval I_{v_k} was added to model $\mathcal{I}_{k-1}$, the positions of v_k^s and v_k^e were chosen with respect to the positions of z_1^e and z_2^s in $\mathcal{I}_{k-1}$. Hence, as z_1 was redefined in model $\mathcal{I}'$, we also need to redefine the start point of interval I_{v_k} accordingly, i.e., we must place the point v_k^s immediately to the left of the new point z_1^e. For convenience, we again use $\mathcal{I}'$ to denote the obtained model.

We claim that the number of obstruction vertices in model $\mathcal{I}'$ is strictly smaller than in model $\mathcal{I}_{k-1}$. First, we observe that x is no longer an obstruction vertex, as $x^s < z_1^s < x^e < v_k^s < z_1^e$ implies that I_x does not intersect I_{v_k} in model $\mathcal{I}'$. It remains to show that the swap did not create any new obstruction vertices. Let us first observe that since x and z_1 are twins in G_{k-1}, every vertex y that is to the right of z_1 and to the left of x in model $\mathcal{I}_{k-1}$ is a twin of x and z_1. This, together with the definition of the new vertex z_1 as the leftmost neighbor of v_k in model $\mathcal{I}'$, implies that the swap did not change any of the adjacencies in G_k, and consequently no new obstruction vertices of type 2 were created. Suppose, for contradiction, that there is a vertex y that was not an obstruction vertex of type 1 in model $\mathcal{I}_{k-1}$, but is an obstruction vertex of type 1 in model $\mathcal{I}'$. Note that the length of the interval I_{v_k} strictly decreased, as the point v_k^s was moved to the right. Hence, I_{v_k} is a subinterval of I_y in model $\mathcal{I}'$ (and not the other way around). Since y was not an obstruction vertex of type 1 in model $\mathcal{I}_{k-1}$, but is an obstruction vertex in model $\mathcal{I}'$, we must have $z_1^s < y^s < x^s$ in the proper circular arc model $\mathcal{I}_{k-1}$. This contradicts the assumption that x and z_1 are twins in G_{k-1}.

Now consider the case where x and z_1 are not twins in G_{k-1}. Then there exists a vertex $z_1' \in N_{G_{k-1}}[z_1] \setminus N_{G_{k-1}}[x]$ or a vertex $x' \in N_{G_{k-1}}[x] \setminus N_{G_{k-1}}[z_1]$ (see Fig. 9). Suppose there exists a vertex $z_1' \in N_{G_{k-1}}[z_1] \setminus N_{G_{k-1}}[x]$. Interval $I_{z_1'}$ is to the left of interval I_{z_1} in model $\mathcal{I}_{k-1}$. Hence, by the definition of z_1, we have $z_1' \notin N_{G_{k-1}}(v_k)$. This yields a contradiction, since $G_k[\{z_1', z_1, x, v_k\}]$ is a claw.

Now suppose that x has a neighbor x' not adjacent to z_1, which means that x' is to the right of x (see Fig. 9). Vertex x' is not adjacent to v_k, since otherwise $G_k[\{v_k, z_1, x, x'\}]$ would be a C_4. Recall that G, and by construction also G_{k-1}, contains the induced cycle C of length at least 7. Since $\mathcal{I}_{k-1}$ is a proper circular arc

Fig. 9 Illustration for Case 2a



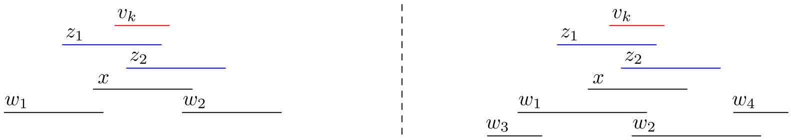


Fig. 10 Illustration for Case 3a. Vertices w_1 and w_1 are either not adjacent (*left figure*) or adjacent (*right figure*) in G_k

model, the intervals representing the vertices of C cover the circle. Let w_1 be the left-most neighbor of z_1 contained in C, and let w_2 be the left neighbor of w_1 on C. As there is no vertex in $N_{G_{k-1}}[z_1] \setminus N_{G_{k-1}}[x]$, we get $w_1 \in N_{G_{k-1}}(x)$. Vertex w_1 is not adjacent to v_k, as it is to the left of z_1 and z_1 is the leftmost neighbor of v_k. Moreover, since w_1^e is to the left of v_k^s and the start point of interval $I_{x'}$ is to the right of v_k^s, w_1 is not adjacent to x'. Finally, as w_2 is to the left of w_1, vertex w_2 is not adjacent to any vertex in $\{z_1, x, v_k, x', z_2\}$. Then $G_k[\{w_2, w_1, z_1, x, x', v_k\}]$ is a net, yielding a contradiction.

Case 2b: $x^s < z_1^e < x^e < z_2^s$, $x \in N_{G_k}(v_k)$ In this case, since $x \in N_{G_k}(v_k)$ and the intervals I_x and I_{v_k} intersect in model $\mathcal{I}_{k-1}$, x is not an obstruction vertex of type 2. As we explained at the start of Case 2a, x is also not an obstruction vertex of type 1. We conclude that Case 2b cannot occur in model $\mathcal{I}^*$.

Case 3a: $x^s < z_2^s < z_1^e < x^e$, $x \notin N_{G_k}(v_k)$ As the intervals in $\mathcal{I}_{k-1}$ representing the vertices of cycle C cover the circle, there exist two vertices w_1 and w_2 on C such that w_1 the leftmost neighbor of z_1 contained in C and w_2 is the rightmost neighbor of z_2 contained in C (see Fig. 10). Vertices w_1 and w_2 are not adjacent to v_k, as z_1 and z_2 are defined as the leftmost and rightmost neighbors of v_k, respectively. If w_1 (respectively w_2) is not adjacent to x, then $G_k[\{w_1, z_1, v_k, x\}]$ (respectively $G_k[\{w_2, z_2, v_k, x\}]$) is a claw. Hence, both w_1 an w_2 must be adjacent to x.

Suppose w_1 and w_2 are not adjacent (see the left side of Fig. 10). If w_1 is adjacent to z_2 or if w_2 is adjacent to z_1, then $G_k[\{w_1, z_2, v_k, w_2\}]$ or $G_k[\{w_1, z_1, w_2, v_k\}]$) is a claw, respectively. This means that w_1 is not adjacent to z_2 and w_2 is not adjacent to z_1. But then $G_k[\{w_1, w_2, z_1, z_2, v_k, x\}]$ is a tent, yielding a contradiction.

Now suppose w_1 and w_2 are adjacent (see the right side of Fig. 10). Let w_3 be the left neighbor of w_1 on cycle C, and let w_4 be the right neighbor of w_2 on C. Since C is an induced cycle, the vertices w_3, w_1, w_2, w_4 form an induced path. Both w_1 and w_2 are adjacent to z_1, and neither w_3 nor w_4 is adjacent to z_1 by the definition of w_1 and w_2. Since v_k is not adjacent to any of the vertices w_1, w_2, w_3, w_4, graph $G_k[\{z_1, w_1, w_2, w_3, w_4, v_k\}]$ is a net. This contradiction shows that Case 3a cannot occur in model $\mathcal{I}^*$.

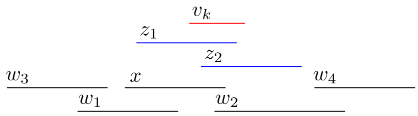


Fig. 11 Illustration for Case 4a

Case 3b: $x^s < z_2^s < z_1^e < x^e$, $x \in N_{G_k}(v_k)$ Since $x \in N_{G_k}(v_k)$ and intervals I_x and I_{v_k} intersect, x is not an obstruction vertex of type 1. However, since I_{v_k} is a subinterval of I_x, x is an obstruction vertex of type 2. As we showed in Case 1b, x cannot have two neighbors x_1 and x_2 such that $x_1^s < x^s < x_1^e < v_k^s$ and $v_k^e < x_2^s < x^e < x_2^e$, since then $G_k[\{x_1, x, x_2, v_k\}]$ would be a claw. As a consequence, either no end point of any interval in model $\mathcal{I}^*$ can be to the right of x^s and to the left of v_k^s, or no start point of any interval in $\mathcal{I}^*$ can be to the right of v_k^e and to the left of x^e.

Let us assume that no end point appears to the right of x^s and to the left of v_k^s; the other case can be dealt with analogously. Just like in Case 1b, we define a set $Y := \{y \in V(G_k) \mid x^s \le y^s < v_k^s\}$. Note that Y is non-empty, as $x \in Y$. We can use the exact same arguments as in Case 1b to show that none of the vertices in Y is an obstruction vertex of type 2, and that every vertex in Y is an obstruction vertex of type 1 due to the fact that, for every $y \in Y$, I_{v_k} is a subinterval of I_y in model $\mathcal{I}^*$.

We modify model $\mathcal{I}^*$ in the exact same way as in Case 1b: we move the start point v_k^s of interval I_{v_k} to the immediate left of x^s, making sure that no start or end point of any interval in $\mathcal{I}^*$ lies to the right of v_k^s and to the left of x^s. As we saw in Case 1b, extending I_{v_k} this way does not create any new obstruction vertices. Moreover, as the vertices of Y are no longer obstruction vertices in the new model and the set Y is non-empty, the number of obstruction vertices strictly decreased.

Case 4a: $x^s < z_2^s < x^e < z_1^e$, $x \notin N_{G_k}(v_k)$ Recall that the intervals representing the vertices of C cover the circle of model $\mathcal{I}^*$. Let w_1 and w_2 be vertices of C such that w_1 is the leftmost neighbor of x contained in C and w_2 is the rightmost neighbor of z_2 contained in C. Furthermore, let w_3 be the left neighbor of w_1 in C, and let w_4 be the right neighbor of w_2 in C (see Fig. 11).

By the definition of z_1 and z_2, vertex v_k is not adjacent to any of the vertices w_1, w_2, w_3, w_4. Vertex w_2 is adjacent to x, since otherwise $G_k[\{w_2, z_2, v_k, x\}]$ is a claw. Vertex w_3 is adjacent to w_1, but not to any vertex in the set $\{x, v_k, z_1, z_2, w_2, w_4\}$, as w_1 the leftmost neighbor of x and w_3 is to the left of w_1. By symmetrical arguments, w_4 is adjacent to w_2, but not to any vertex in $\{x, v_k, z_1, z_2, w_1, w_4\}$. Vertices w_3 and w_4 are not adjacent, since then the six intervals I_{w_3}, I_{w_1}, I_{z_1}, I_{z_2}, I_{w_2}, and I_{w_4} would cover the circle of the proper circular arc model $\mathcal{I}_{k-1}$, contradicting Lemma 2.

Since the graph $G_k[\{v_k, z_2, x, w_2, w_4, w_1\}]$ is not a net, at least one of the edges $\{w_1, w_2\}, \{w_1, z_2\}$ is present in G_k; all other potential edges have already been excluded. Suppose $\{w_1, w_2\} \notin E$, which means that $\{w_1, z_2\} \in E$. In this case, $G[\{w_1, z_2, w_2, v_k\}]$ is a claw, yielding a contradiction. Hence we must have $\{w_1, w_2\} \in E$. Moreover, we must have $\{w_1, z_2\} \in E$, since otherwise $G_k[\{w_1, w_2, w_4, z_2\}]$ would be a claw. Now we obtain a contradiction, as $G[\{v_k, z_2, w_3, w_1, w_4, w_2\}]$ is a net. We conclude that Case 4a cannot occur in model $\mathcal{I}^*$.

Case 4b: $x^s < z_2^s < x^e < z_1^e$, $x \in N_{G_k}(v_k)$ Recall that z_1 is defined to be the leftmost neighbor of v_k in model $\mathcal{I}^*$. Since x is adjacent to v_k and x is to the left of z_1, we immediately obtain a contradiction, implying that Case 4b cannot occur.

Case 5: $z_1^e < x^s < z_2^s < x^e$ This case is symmetric to Case 2.

Case 6a: $z_1^e < x^s < x^e < z_2^s$, $x \notin N_{G_k}(v_k)$ In this case, the vertices z_1 and z_2 are not adjacent to each other, and neither z_1 nor z_2 is adjacent to x. By Lemma 3, there exist two consecutive vertices w_t, w_{t+1} on cycle C such that $w_t, w_{t+1} \in N_{G_k}(v_k)$ and $N_{G_k}(v_k) \subseteq N_{G_k}(w_t) \cup N_{G_k}(w_{t+1})$. Note that, apart from interval I_{v_k}, there is no interval in model $\mathcal{I}^*$ whose start point is to the left of z_1^e and whose end point is to the right of z_2^s, since otherwise I_x would be a subinterval of such an interval in $\mathcal{I}_{k-1}$, contradicting the assumption that $\mathcal{I}_{k-1}$ is a proper circular arc model. In particular, this means that there is an induced path z_1, w_i, w_{i+1}, z_2 in $N_{G_k}(v_k)$ such that w_i, w_{i+1} are vertices of C; it is possible that $z_1 = w_t$ or $z_2 = w_{t+1}$. Interval I_x is not a subinterval of I_{w_i} or $I_{w_{i+1}}$ in the proper circular arc model $\mathcal{I}_{k-1}$, so x is adjacent to both w_i and w_{i+1}. Now we have a contradiction, since $G_k[\{z_1, w_i, w_{i+1}, z_2, x, v_k\}]$ is a tent. Hence Case 6a does not occur in model $\mathcal{I}^*$.

Case 6b: $z_1^e < x^s < x^e < z_2^s$, $x \in N_{G_k}(v_k)$ Just like in Case 6a, z_1 and z_2 are not adjacent to each other or to vertex x. Since we now consider the case where $x \in N_{G_k}(v_k)$, the graph $G_k[\{z_1, x, z_2, v_k\}]$ is a claw. This contradiction implies that Case 6b cannot occur.

Case 7: $z_2^s < x^s < z_1^e < x^e$ This case is symmetric to Case 4.

Case 8: $z_2^s < x^s < x^e < z_1^e$ This case cannot occur, since I_x is a subinterval of both I_{z_1} and I_{z_2}, contradicting the assumption that $\mathcal{I}_{k-1}$ is a proper circular arc model of G_{k-1}.

For each of the eight cases above, we either proved that they cannot occur in model $\mathcal{I}^*$, or we showed how model $\mathcal{I}^*$ can be manipulated in such a way that the number of obstruction vertices strictly decreases. Since we obtain a proper circular arc model $\mathcal{I}_k$ of G_k once we have successfully decreased the number of obstruction vertices to 0, this completes the proof of Theorem 1. □

4 Algorithmic Implications

In this section, we present an $\mathcal{O}(6^k kn^6)$ time algorithm for Proper Interval Vertex Deletion. Our FPT algorithm is based on the structural characterization of almost proper interval graphs given in Theorem 1, together with a linear-time algorithm for solving Proper Interval Vertex Deletion on almost proper interval graphs. We also present a 6-approximation algorithm for the optimization variant of Proper Interval Vertex Deletion on general graphs.

Recall that Wegner [25] showed that the class of proper interval graphs is exactly the class of {claw, net, tent, hole}-free graphs, and that an almost proper interval graph is defined to be a {claw, net, tent, C_4, C_5, C_6}-free graph. Hence, a graph G

is a proper interval graph if and only if G is an almost proper interval graph that is hole-free. Solving the PROPER INTERVAL VERTEX DELETION problem on almost proper interval graphs therefore boils down to finding a smallest subset of vertices in an almost proper interval graph whose deletion destroys all the holes in the graph. This motivates the following definition.

Let $G = (V, E)$ be a connected almost proper interval graph. A *hole cut* of G is a vertex set $X \subseteq V$ such that $G - X$ is a proper interval graph. A hole cut X of G is *minimal* if X is empty or if, for every proper subset $X' \subset X$, the graph $G - X'$ contains a hole. A hole cut of G is *minimum* if G does not have a hole cut whose size is strictly smaller than the size of X.

Due to Theorem 1, graph G is a proper circular arc graph. Let $\mathcal{I}$ be a proper circular arc model of G. For any vertex $v \in V$, let R_v denote the set of neighbors of v that are to the right of v, i.e., $R_v := \{w \in V \mid w^s < v^e < w^e\}$. Clearly, for every vertex $v \in V$, the set R_v is a hole cut; after all, in the proper circular arc model $\mathcal{I}'$ of the graph $G - R_v$, vertex v does not have any neighbor that is to the right of v. This implies that we can "cut" the circle in model $\mathcal{I}'$ to the immediate right of point v^e in order to obtain a proper interval model of $G - R_v$. The following lemma shows that every minimal hole cut of G is exactly the set R_v for some $v \in V$.

Lemma 5 *Let G be a connected almost proper interval graph, and let $\mathcal{I}$ be a proper circular arc model of G. For every minimal hole cut X of G, there exists a vertex v in G such that $X = R_v$.*

Proof First suppose G is a proper interval graph. Then the unique minimal hole cut of G is the empty set. Moreover, since G is connected, there is a unique vertex $v \in V(G)$ such that R_v is empty. Hence, $X = R_v$ in this case.

Now suppose G is not a proper interval graph. Let X be a minimal hole cut of G. Suppose, for contradiction, that $R_v \nsubseteq X$ for each vertex $v \in V(G)$. Then every vertex in $V(G) \setminus X$ has a neighbor to the right, where "to the right" is defined with respect to the proper circular arc model $\mathcal{I}$ of G. As a consequence, there exists a set of intervals in $\mathcal{I}$ representing vertices in $V(G) \setminus X$ such that the union of these covers the circle. Let $\mathcal{Y}$ be an inclusion-minimal such set. Since G is a connected almost proper interval graph that is not a proper interval graph, G contains a hole C of length at least 7. Moreover, since G is a proper circular arc graph by Theorem 1, we know by Lemma 1 that for every subset $\mathcal{X} \subseteq \mathcal{I}$ of intervals that cover the circle of model $\mathcal{I}$, we have $|\mathcal{X}| \geq 4$. This implies that $|\mathcal{Y}| \geq 4$, which means that the set Y of corresponding vertices in G induces a hole in G. Since $Y \cap X = \emptyset$, the vertices in Y also induce a hole in $G - X$, contradicting the fact that X is a hole cut of G. □

Lemma 5 implies that a connected almost proper interval graph contains at most n minimal hole cuts. The next lemma shows how we can exploit this fact in order to solve PROPER INTERVAL VERTEX DELETION in linear time on connected almost proper interval graphs.

Lemma 6 *The* PROPER INTERVAL VERTEX DELETION *problem can be solved in $\mathcal{O}(n + m)$ time on almost proper interval graphs.*

Proof Let (G, k) be an instance of PROPER INTERVAL VERTEX DELETION, where G is an almost proper interval graph. Let us first assume that G is connected; at the end of this proof, we consider the case where G is not connected. Note that (G, k) is a yes-instance if and only if G has a hole cut of size at most k. Hence, in order to prove Lemma 6 for connected almost proper interval graphs, it suffices to show that we can find a minimum hole cut of G in $\mathcal{O}(n+m)$ time.

We first check if G is a proper interval graph. It is well-known that this can be done in $\mathcal{O}(n+m)$ time, for example using the recognition algorithm for proper interval graphs by Deng et al. [8]. If G is a proper interval graph, then the empty set is the unique minimum hole cut of G.

Suppose G is not a proper interval graph. By Lemma 5, for every minimal hole cut X of G, there exists a vertex $v \in V(G)$ such that $X = R_v$. Since a minimum hole cut is also minimal, we only need to show how we can find, in linear time, a set R_v in G of smallest size. In order to find such a set, we use a FIFO queue Q, where vertices of G will be added to the back of the queue and removed from the front of the queue. We will make two "round trips" along the circle of model $\mathcal{I}$, starting at point x^e for an arbitrary vertex x of G, and traversing the circle in the clockwise direction, i.e., from left to right.

The first round trip is used to add and remove vertices to and from Q in such a way that at the moment x^e has been reached for the second time, Q contains exactly the vertices of R_x. This can be done as follows. After adding x to the queue, we start traversing the circle from left to right, starting at point x^e. As soon as a point y^s is reached, vertex y is added to the back of the queue, and vertex y is removed from the front of queue as soon as point y^e is reached. Note that the fact that $\mathcal{I}$ is a proper circular arc model guarantees that vertex y is at the front of the queue at the moment y^e is reached. Let us consider which vertices are in the queue at the moment x^e has been reached for the second time, i.e., at the moment x has been removed from the queue for the first time. For every vertex $z \in V(G) \setminus R_x$, we encountered both z^s and z^e, which means that z is not in Q. On the other hand, for every $z \in R_x$, we only encountered point z^s, and not z^e. This means that Q contains exactly the vertices of R_x after x^e has been reached for the second time.

In the second round trip, we again add a vertex y to the queue when y^s is reached, and remove y from the queue when as soon as point y^e is reached. Every time a point y^e is reached, Q contains exactly those vertices whose intervals contain the point y^e, i.e., Q contains exactly the vertices of the set R_y. Hence, by keeping track of the size of Q during the second round trip, and returning the elements of Q at the moment it has smallest size, we can find a set R_v such that $|R_v| \leq |R_w|$ for every $w \in V(G)$. As we argued before, such a set R_v is a minimum hole cut of G due to Lemma 5.

It remains to argue how to proceed when the input graph G is not connected. Let $G_1, \ldots, G_r$ be the connected components of G, and let n_i and m_i denote the number of vertices and edges in G_i, respectively, for $1 \leq i \leq r$. For each connected component G_i, we can use the above procedure to find, in $\mathcal{O}(n_i + m_i)$ time, a minimum hole cut of G_i. Let x_i be the size of a minimum hole cut in G_i, for $1 \leq i \leq r$, and let $x := \sum_{i=1}^{r} x_i$. Since $\sum_{i=1}^{r} n_i = n$ and $\sum_{i=1}^{r} m_i = m$, the value of x can be computed in $\mathcal{O}(n+m)$ time. It is obvious that G has a hole cut of size at most k if and only if $x \leq k$. □

We are now ready to provide an efficient FPT algorithm for the PROPER INTERVAL VERTEX DELETION problem on general graphs.

Theorem 2 *The* PROPER INTERVAL VERTEX DELETION *problem can be solved in* $\mathcal{O}(6^k kn^6)$ *time.*

Proof Let (G,k) be an instance of PROPER INTERVAL VERTEX DELETION, and let us refer to the integer k as the *budget* of the instance. Any set $X \subseteq V(G)$ with $|X| \leq k$ such that $G - X$ is a proper interval graph is called a *solution* for (G,k). Recall that the forbidden induced subgraphs of proper interval graphs are the claw, the net, the tent, and all the holes [25] (see also Fig. 1). In order to transform G into a proper interval graph, we need to delete vertices from G such that the obtained graph does not contain any of the aforementioned forbidden induced subgraphs.

The first phase of the algorithm is a bounded search tree procedure that transforms the instance (G,k) into $\mathcal{O}(6^k)$ sub-instances, such that the graph in each sub-instance is an almost proper interval graph, and the budget is at least 0. This is done as follows. Starting with the instance (G,k), the algorithm checks whether G contains a subset of vertices U such that U induces a claw, a net, a tent, a C_4, a C_5, or a C_6 in G. Since each of these graphs has at most six vertices, this check can trivially be performed in $\mathcal{O}(n^6)$ time. If no such set U is found, then G is an almost proper interval graph, and the algorithm proceeds to the next phase. If such a set U is found, the algorithm branches on the at most six possible ways of deleting a single vertex u from U, creating at most six sub-instances (G',k'), where $G' := G - u$ for some vertex $u \in U$, and $k' := k - 1$. Since any solution for (G,k) must contain at least one vertex of each forbidden induced subgraph in G, we have that (G,k) is a yes-instance if and only if at least one of the sub-instances (G',k') is a yes-instance of PROPER INTERVAL VERTEX DELETION.

As long as there is a sub-instance in which the graph is not an almost proper interval graph and the budget is at least 1, another round of branching is performed; every sub-instance in which the graph is already an almost proper interval graph is left untouched. Each time the algorithm branches, the budget is decreased by exactly 1. Since the initial budget was k, the first phase is completed after at most k rounds of branching. The above procedure naturally defines a search tree, where each of the leaves corresponds to a sub-instance. At the end of the first phase, the algorithm discards every sub-instance in which the graph is not $\{\text{claw}, \text{net}, \text{tent}, C_4, C_5, C_6\}$-free and the budget is 0. If all sub-instances are discarded, then G contains more than k vertex-disjoint forbidden induced subgraphs, which means that (G,k) is a no-instance. Otherwise, the algorithm proceeds to the second phase, which is described below. Since the algorithm branches at most k times, the total number of sub-instances that is created in the first phase of the algorithm is $\mathcal{O}(6^k)$. The total time used in this phase is $\mathcal{O}(6^k kn^6)$.

In the second phase, the algorithm considers each sub-instance (G'',k'') where G'' is an almost proper interval graph and $k'' \geq 0$. Since G'' is an almost proper interval graph, we can use the algorithm of Lemma 6 to decide in $\mathcal{O}(n+m)$ time whether (G'',k'') is a yes-instance. As argued before, it is clear that (G,k) is a yes-instance if and only if at least one of the sub-instances (G'', k'') is a yes-instance. This completes the proof of Theorem 2. □

For the remainder of this subsection, we consider PROPER INTERVAL VERTEX DELETION to be an optimization problem rather than a decision problem. In other words, we define PROPER INTERVAL VERTEX DELETION to be the problem that takes as input a graph G, and the task is to find a vertex subset $X \subseteq V(G)$ of minimum size such that $G - X$ is a proper interval graph. We now show how the FPT algorithm of Theorem 2 can be turned into a 6-approximation algorithm for PROPER INTERVAL VERTEX DELETION.

Theorem 3 *There is a* 6*-approximation algorithm for the* PROPER INTERVAL VERTEX DELETION *problem running in time* $\mathcal{O}(n^7)$.

Proof Let $G = (V, E)$ be a graph. We describe an $\mathcal{O}(n^7)$ time procedure that finds a set $X \subseteq V$ such that $G - X$ is a proper interval graph, and such that $|X| \leq 6|Y|$ for any set $Y \subseteq V$ such that $G - Y$ is a proper interval graph. Initially, the set X is empty. As long as there exists a vertex set $U \subseteq V$ such that $G[U]$ is a claw, a net, a tent, a C_4, a C_5, or a C_6, we delete all the vertices of U from G, and add all the vertices of U to X. If no such vertex set U exists, then the remaining graph G' is an almost proper interval graph by definition. Using the algorithm of Lemma 5, we can find a minimum hole cut X' of G' in $\mathcal{O}(n + m)$ time. Finally, we add all the vertices of X' to X. The approximation factor 6 follows from the observation that each set $U \neq X'$ whose vertices are added to X contains at most 6 vertices, and that any hole cut Y of G must contain at least one vertex from each such set U.

It remains to analyze the running time. Since each of the sets U that is found in the first phase of the algorithm contains at most 6 vertices, each such set U can be found in $\mathcal{O}(n^6)$ time. As the algorithm finds $\mathcal{O}(n)$ sets U in total, the first phase of the algorithm takes $\mathcal{O}(n^7)$ time. The second phase of the algorithm, i.e., finding the hole cut X' of the almost proper interval graph G', takes $\mathcal{O}(n + m)$ time. This yields an overall running time of $\mathcal{O}(n^7)$. □

5 Conclusion

We proved that every {claw, net, tent, C_4, C_5, C_6}-free graph is the disjoint union of proper circular arc graphs. Using this structural result, we obtained an $\mathcal{O}(6^k kn^6)$ time algorithm for PROPER INTERVAL VERTEX DELETION, as well as a polynomial-time 6-approximation algorithm for the optimization variant of the problem.

Van Bevern et al. [23] proved that the PROPER INTERVAL VERTEX DELETION problem is NP-hard on graphs that are {claw, net, tent}-free. We showed that the problem can be solved in linear time on almost proper interval graphs, i.e., on graphs that are {claw, net, tent, C_4, C_5, C_6}-free. It would be interesting to know if the problem remains NP-hard when restricted to {claw, net, tent, C_4}-free graphs, or if it becomes polynomial-time solvable on this graph class.

To conclude this paper, we mention two related open problems. Is the problem of deciding whether at most k vertices can be deleted from a given graph in order to get an interval graph or a proper circular arc graph fixed-parameter tractable when parameterized by k? Settling the parameterized complexity of this problem for interval graphs seems to be the more interesting of the two.

Acknowledgements We would like to thank Fedor V. Fomin for comments and suggestions on the manuscript and two anonymous referees for excellent reports, which helped us to significantly improve the presentation of the paper. This work has been supported by the Norwegian Research Council.

References

1. Brandstädt, A., Dragan, F.F.: On linear and circular structure of (claw, net)-free graphs. Discrete Appl. Math. **129**(2–3), 285–303 (2003)
2. Brandstädt, A., Le, V.B., Spinrad, J.P.: Graph Classes: A Survey. Society for Industrial and Applied Mathematics, Philadelphia (1999)
3. Cai, L.: Fixed-parameter tractability of graph modification problems for hereditary properties. Inf. Process. Lett. **58**(4), 171–176 (1996)
4. Cao, Y., Chen, J., Liu, Y.: On feedback vertex set new measure and new structures. In: Proceedings SWAT 2010. Lecture Notes in Computer Science, vol. 6139, pp. 93–104. Springer, Berlin (2010)
5. Chen, J., Liu, Y., Lu, S., O'Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. J. ACM **55**(5), 21:1–21:19 (2008)
6. Chen, J., Kanj, I.A., Xia, G.: Improved upper bounds for vertex cover. Theor. Comput. Sci. **411**(40–42), 3736–3756 (2010)
7. Courcelle, B.: Graph rewriting: an algebraic and logic approach. In: Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics (B), pp. 193–242 (1990)
8. Deng, X., Hell, P., Huang, J.: Linear-time representation algorithms for proper circular-arc graphs and proper interval graphs. SIAM J. Comput. **25**(2), 390–403 (1996)
9. Gavril, F.: Algorithms for minimum coloring, maximum clique, minimum covering by cliques, and maximum independent set of a chordal graph. SIAM J. Comput. **1**(2), 180–187 (1972)
10. Golumbic, M.C.: Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York (1980)
11. Grötschel, M., Lovász, L., Schrijver, A.: Polynomial algorithms for perfect graphs. In: Berge, C., Chvátal, V. (eds.) Topics on Perfect Graphs. Annals of Discrete Mathematics, vol. 21, pp. 325–356 (1984)
12. Heggernes, P., van 't Hof, P., Jansen, B.M.P., Kratsch, S., Villanger, Y.V.: Parameterized complexity of vertex deletion into perfect graph classes. In: Proceedings FCT 2011. Lecture Notes in Computer Science, vol. 6914, pp. 240–251 (2011)
13. Kaplan, H., Shamir, R., Tarjan, R.E.: Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. SIAM J. Comput. **28**(5), 1906–1922 (1999)
14. Kawarabayashi, K., Reed, B.A.: Computing crossing number in linear time. In: Proceedings STOC 2007, pp. 382–390. ACM, New York (2007)
15. Lekkerkerker, C., Boland, J.: Representation of a finite graph by a set of intervals on the real line. Fundam. Math. **51**, 45–64 (1962)
16. Lewis, J.M., Yannakakis, M.: The node-deletion problem for hereditary properties is NP-complete. J. Comput. Syst. Sci. **20**(2), 219–230 (1980)
17. Lokshtanov, D.: Wheel-free deletion is W[2]-hard. In: Proceedings IWPEC 2008. Lecture Notes in Computer Science, vol. 5018, pp. 141–147. Springer, Berlin (2008)
18. Marx, D.: Chordal deletion is fixed-parameter tractable. Algorithmica **57**(4), 747–768 (2010)
19. Marx, D., Schlotter, I.: Obtaining a planar graph by vertex deletion. Algorithmica **62**(3–4), 807–822 (2012)
20. Roberts, F.S.: Indifference graphs. In: Proof Techniques in Graph Theory, pp. 139–146. Academic Press, San Diego (1969)
21. Seymour, P.D., Thomas, R.: Call routing and the ratcatcher. Combinatorica **14**(2), 217–241 (1994)
22. Tucker, A.: Structure theorems for some circular-arc graphs. Discrete Math. **7**, 167–195 (1974)
23. van Bevern, R., Komusiewicz, C., Moser, H., Niedermeier, R.: Measuring indifference: unit interval vertex deletion. In: Proceedings WG 2010. Lecture Notes in Computer Science, vol. 6410, pp. 232–243. Springer, Berlin (2010)

24. Villanger, Y.V., Heggernes, P., Paul, C., Telle, J.A.: Interval completion is fixed parameter tractable. SIAM J. Comput. **38**(5), 2007–2020 (2009)
25. Wegner, G.: Eigenschaften der Nerven homologisch-einfacher Familien im R^n. Ph.D. thesis, University of Göttingen (1967)
26. Yannakakis, M.: Edge-deletion problems. SIAM J. Comput. **10**(2), 297–309 (1981)
27. Yannakakis, M.: Computing minimum fill-in is NP-complete. SIAM J. Algebr. Discrete Methods **2**(1), 77–79 (1981)