An Efficient Branching Algorithm for Interval Completion

Yixin Cao*

Abstract

We study the interval completion problem, which asks for the insertion of a set of at most \( k \) edges to make a graph of \( n \) vertices into an interval graph. We focus on chordal graphs with no small obstructions, where every remaining obstruction is known to have a shallow property. From such a shallow obstruction we single out a subset 6 or 7 vertices, called the frame, and 5 missed edges in the subgraph induced by the frame. We show that if none of these edges is inserted, then the frame cannot be altered at all, and the whole obstruction is also fixed, by and large, in the sense that their related positions in an interval representation of the objective interval graph have a specific pattern. We propose a simple bounded search process, which effectively transforms a given graph to a graph with the structural property that all obstructions are shallow and have fixed frames. Then we fill in polynomial time all obstructions that have been previously left in indecision. These efforts together deliver a simple parameterized algorithm of time \( 6^k \cdot n^{O(1)} \) for the problem, significantly improving the only known parameterized algorithm of time \( k^{2k} \cdot n^{O(1)} \).

1 Introduction

A graph is an interval graph if its vertices can be assigned to the intervals of the real line such that there is an edge between two vertices if and only if their corresponding intervals intersect. Interval graphs are the natural models for DNA chains in biology [2] and many other applications, among which the most cited ones include jobs scheduling in industrial engineering [11] and seriation in archeology [16]. Motivated by pure contemplation of combinatorics and practical problems of biology respectively, Hajós [13] and Benzer [2] independently initiated the study of interval graphs. An interval graph \( G \) is called an interval supergraph of \( G \) if they have the same vertex set and every edge of \( G \) also appears in \( G \). The minimum interval completion problem asks for the minimum size of interval supergraphs of a given graph; or equivalently, the minimum number of edges whose insertion transforms a graph into an interval graph. Originally formulated in sparse matrix computations [21], this problem later found application in physical mapping of DNA [15]. A similar and related problem is chordal completion, which is also widely known as minimum fill-in. A graph is chordal if it contains no hole, and the chordal completion problem asks for the minimum size of chordal supergraphs of a given graph.

These problems are, understandably, NP-hard [10, 24]. Therefore, early work of Kaplan et al. [15] and Cai [4] focused on their parameterized tractability. Recall that a problem, parameterized by \( k \), is fixed-parameter tractable (FPT) if it admits an algorithm with runtime \( f(k) \cdot n^{O(1)} \), where \( f \) is a computable function depending only on \( k \) [7]. Cai [4] observed that if a hereditary graph class \( \mathcal{H} \) can be characterized by a finite number of forbidden (induced) subgraphs, then the fixed-parameter tractability of the completion problem follows from a basic bounded search tree algorithm. Many important graph classes, however, have minimal obstructions of arbitrary large size; in particular, holes of any length are forbidden in both interval and chordal graphs. Even so, chordal completion can still be solved by a bounded search tree algorithm by observing that a large hole readily implies a negative answer to the problem [15].

An interval graph is known to be chordal and not contain a structure called “asteroidal triple” (or AT for short), i.e., three vertices of which each pair is connected by a path avoiding neighbors of the third one [17]. Therefore, to solve interval completion, one has to destroy not only all holes, but all ATs as well. Using bounded search to fill holes and small obstructions is now a pedestrian task [15, 4, 23], which focuses us on large witnesses for ATs in a chordal graph. Such a witness is known to have the shallow property. For a shallow witness \( X \), there is a set of \( O(|X|) \) edges such that the insertion of any of them will suffice to break \( X \). This hints its disposal has to be harder than the holes; a similar dichotomy has been observed in breaking holes by deletion versus by completion [18].

Observing the fixed-parameter tractability of chordal completion, Kaplan et al. [15] asked if the apparently harder interval completion problem is FPT as well. This question was, after a dozen years, resolved

*Institute for Computer Science and Control, Hungarian Academy of Sciences. Email: yixin@sztaki.hu. Supported by the European Research Council (ERC) grant "PARAMTIGHT: Parameterized complexity and the search for tight complexity results," reference 280152.
by Villanger et al. [23], who designed a $k^{2k} \cdot n^{O(1)}$ time algorithm. It remains the only know FPT algorithm for the problem. The main purpose of this paper is to propose a simple and improved FPT algorithm.

**Theorem 1.1.** There is a $6^k \cdot n^{O(1)}$ time algorithm for deciding whether or not there is a set of at most $k$ edges whose insertion makes an $n$-vertex graph $G$ an interval graph.

**Our techniques.** The main technical observation behind our disposal of a shallow witness is its frame and shallow terminal. A shallow witness contains a unique AT; they are called terminals of this shallow witness. Its frame is defined to be the union of the terminals as well as their neighbors; it consists of 6 or 7 vertices. All other vertices of the shallow witness are inner vertices of the longest defining path. (See Figure 1). The terminal neither in nor adjacent to this path is the shallow terminal. We show there is a set of 5 special edges in the frame such that if an interval supergraph $\hat{G}$ contains none of them, then it contains no other edge in the frame, —such a frame is already finalized. Then an edge between the shallow terminal and a vertex out of the frame has to be inserted. This observation suggests that we branch on either inserting one of the 5 edges to its frame, or assuming the frame will appear in $\hat{G}$ as is. In the last case, we put aside the shallow terminal and work on the remaining part.

Here comes the two crucial combinatorial results that justify this partition. All shallow terminals in a chordal graph without small obstructions are well clustered, i.e., either similar or disconnected; and the similar ones can be treated in the same way. Formally, we show those shallow terminals form a set of disjoint modules (a set of vertices with the same neighbors out of it); and more importantly, such a module can never be broken: edges between it and other vertices must be inserted in an all-or-none manner. They together permit us to have a clear cut on any shallow witness whose frame is fixed.

We then turn to next shallow witness and repeat this process. After it is exhausted, we are left with a partition of two disjoint interval subgraphs. A polynomial-time procedure will suffice to merge them and finish the job. We also observe that for small obstructions a 6-way branching will suffice, down from the trivial one that takes up to 12 ways. These studies enable us to achieve the desired time complexity.

We would like to call special attention to the preservation of modules in interval completion, which should not be confused with the similar fact for the deletion problem, where it comes as a trivial consequence of hereditary property. On one hand, this property definitely benefits the further study of this problem; indeed, if we assume the existence of a small solution, the graph necessarily has many modules. On the other hand, such a property can be shown to hold in completion problems to other graph classes and might be helpful.

**Related work.** This time complexity asymptotically matches that of the algorithm for INTERVAL DELETION [5], which is inherently harder than INTERVAL COMPLETION. Compared to the deletion problem, the completion of holes is well understood, both combinatorially [4] and computationally [15, 8, 9]. To fill a large hole we need a large number of edges, while comparatively, the removal of a single vertex will suffice to break an arbitrarily large hole. So holes pose themselves as a much more significant trouble to the deletion problem. This fact, unfortunately, leads some authors to believe an approach for INTERVAL DELETION can be trivially adapted for INTERVAL COMPLETION. This is nevertheless not the case, and the completion problem has its own peculiar difficulties we have to surmount.

An easy fact is, by removing any vertex from a minimal forbidden subgraph, we break this subgraph once and for all. Noting that interval graphs are hereditary, an interval deletion set can thus be viewed as a hitting set for all minimal forbidden subgraphs. On the other hand, an edge inserted to fix an erstwhile forbidden subgraph might introduce a new one. Such side effects arise great bitterness, and require extreme care on each step. One might then be tempted to consider a set $E^+$ of edges that hits every minimal forbidden subgraph (both ends in this subgraph) and each unit, i.e., a subset of $E^+$, is free of side effects. A second thought tells us that nevertheless this additional side-effect-free condition is only a placebo, and adds nothing to what counts as a proof. Observe that “neither $G + E_1$ nor $G + E_2$ contains a new forbidden subgraph” is not a sufficient condition for “$G + (E_1 \cup E_2)$ contains no new forbidden subgraph.” Indeed, if this sort of argument might work, it has to be something like “every to-be-inserted-edge-unit is side-effect-free to every intermediate graph.” Or equivalently, there is an ordered partition $(E_1, \ldots, E_\ell)$ of $E^+$ such that for any $1 \leq i < \ell$, the insertion of $E_{i+1}$ will not introduce a new forbidden subgraph to $G + (E_1 \cup \cdots \cup E_i)$ as a side effect. This argument has to be extremely complicated, if doable at all: a single edge will break everything, even if every previous edge serves its purpose faithfully and successfully.

Interval graphs and chordal graphs are not the only graph classes that receive attention in this respect. Other graph classes include perfect graphs and proper interval graphs. A graph is perfect if neither it or its

---

$^1$Some author did claim a result based on a falsified assumption that a sequence of “safe” edge sets fixing all forbidden subgraphs of $G$ makes an interval completion set, where a set $E_X$ of edges is called “safe” if $G + E_X$ contains no new forbidden subgraph.
complement contains an odd hole. An interval graph is a proper interval graph if it has a representation with no interval containing another one. These four classes have a proper containment relations, proper interval graphs are a subclass of interval graphs, and all graph classes are perfect. The known results on these graphs classes are summarized in Table 1. Note that a W[2]-hard problem is unlikely to be FPT [7].

This paper is organized as follows. Section 2 sets the definitions and recalls some known facts. Section 3 depicts minimal ways to fix minimal forbidden graphs. Section 4 characterizes the external and internal behavior of modules in a minimum interval supergraph. Section 5 investigates shallow terminals in chordal graphs with no small obstructions. Section 6 presents our bounded search tree algorithm, and proves Theorem 11. Section 7 closes this paper with some technical remarks.

2 Preliminaries

Graphs discussed in this paper shall always be undirected and simple. The vertex set and edge set of a graph G are denoted by V(G) and E(G) respectively. The size of graph G is defined to be |E(G)|, i.e., the number of edges in it. We say G' is a supergraph of G if V(G) = V(G') and E(G) ⊆ E(G'). Given a set E' of missed edges in V(G) \ E(G), the supergraph G + E' is defined to be (V(G), E(G) ∪ E').

We say that a pair of vertices u and v is adjacent (to each other) if they are connected by an edge, denoted by u ∼ v; otherwise nonadjacent and denoted by u /∼ v. Two vertex sets X and Y are completely connected if x ∼ y for every pair of x ∈ X and y ∈ Y. We denote by N_G(v) the set of neighbors of v, i.e., vertices adjacent to v, and N_G[U] = N_G[v] ∪ U. Let U ⊆ V(G) be a subset of vertices. Neighbors of U are defined analogously: N_G[U] = ∪_{v∈U} N_G[v] and N_G[U] = N_G[U] \ U. The subscript will be omitted if it is clear from context. A clique in a graph where every pair of vertices is adjacent. A vertex is simplicial if its neighbors induce a clique. The subgraph induced by U is denoted by G[U], and G − U is used as a shorthand for G[V(G) \ U]. We say a connected induced subgraph G[U] is a connected component of G if N(U) = ∅.

A sequence of distinct vertices v₀v₁...v_l such that v_i ∼ v_{i+1} for each 0 ≤ i < l is called a path, or a v₀-v_l path when the two end vertices are of special interest; the length is defined to be l. If l > 1 and v₀ ∼ v_l, then the sequence v₀v₁...v_{l-1}v_l is called a cycle of length l + 1. As an abuse of notation, we use u ∈ P (resp., u ∈ C) to denote that the vertex u appears in the path P (resp., cycle C), i.e., we also consider a path (resp., cycle) as the set of elements in the sequence. A hole is an induced cycle of length at least 4.

An interval representation of an interval graph G is given by I = {I_v|v ∈ V(G)} Each vertex v corresponds to an closed interval I_v with endpoints left(v) and right(v); i.e., I_v = [left(v), right(v)]. Likewise, for a subset X of vertices, we define left(X) = min_{x∈X} left(x) and right(X) = max_{x∈X} right(x). Observe that if a subset X of vertices induces a connected subgraph, then the union of {I_v|v ∈ X} also forms an interval. Assume without loss of generality, no intervals for distinct vertices share a same endpoint; as a result, an interval representation of graph G defines precisely 2|V(G)| distinct endpoints. For any point p, we can find a small positive value ϵ such that in [p − ϵ, p + ϵ] the only possible endpoint of an interval for some vertex is p. We define K_p = {v|p ∈ I_v}, which clearly induces a clique.

A set X of vertices is an x-y separator if x, y /∈ X and there is no x-y path in G − X. The following lemma relates an interval representation of an interval graph and its separators.

Lemma 2.1. Let u and v be a pair of nonadjacent vertices in a connected interval graph G, and I be an interval representation of G. A set X of vertices is a u-v separator if and only if K_p ⊆ X for some point p such that I_u and I_v lie on different sides of p.

We say that an interval graph G is an interval supergraph of G if G ⊆ G. An interval supergraph G is minimum if there is no strictly smaller interval supergraph of G. No generality will be lost by assuming G is connected. This problem is hence defined as follows:

**Interval completion** (G, k): Given a connected graph G and a nonnegative integer k, find an interval supergraph of G of size no more than ||G|| + k, or report no such a graph exists.
Moreover, using our generalized notations, these edges are exactly the same. See Lemma 3.4 and its corollary.

### 3 Forbidden induced subgraphs

A forbidden induced subgraph refers to a non-interval graph, and it is minimal if every proper induced subgraph of it is an interval graph. Three vertices form an asteroidal triple (AT) if each pair of them is connected by a path that avoids the neighbors of the third one. We use asteroidal witness (AW) to refer to a minimal forbidden induced subgraph that is not a hole. It should be easy to check that an AW contains precisely one AT, called terminals of this AW; and its vertex set is the union of these three defining paths for this triple. By definition, the terminals are the only simplicial vertices of this AW and they are nonadjacent to each other. Lekkerkerker and Boland [17] observed that a graph is an interval graph if and only if it is chordal and contains no AW, and more importantly, proved the following characterization.

**Theorem 3.1** ([17]). A minimal forbidden induced subgraph is either a hole or an AW depicted in Figure 1.

![Asteroidal witnesses in a chordal graph (terminals are marked as squares).](image)

Some remarks are in order. First, it is easy to verify that a hole of 6 or more vertices witnesses an AT, e.g., any three nonadjacent vertices within it, but following convention, we only refer to it as a hole, and reserve the term AW for graphs listed in Figure 1. Second, for the purpose of the current paper, we single out AT- and †-AWs with $d \leq 3$, and denote them by $d$-nets, and respectively $d$-tents; they, together with long claws and whipping tops, are called small AWs. The others, i.e., †- and ‡-AWs with $d > 3$, are called long AW.

The frame of a long AW is defined to be the union of the terminals and their neighbors. By definition, at least one vertex of the long AW is not in its frame; all of them belong to the longest defining path. The ends $(l, r)$ and inner vertices ($B = \{b_1, \ldots, b_d\}$) of this path are called base terminals and base (vertices) respectively. The other terminal $s$ is the shallow terminal, whose neighbor(s) $c$ or $(c_1, c_2)$ are the center(s).

To avoid repetition of the essentially same argument for †-AWs and ‡-AWs, we use a generalized notation for both †- and ‡-AWs. In particular, both $c_1$ and $c_2$ refer to the only center $c$ when it is a †-AW. As long as we do not use the adjacency of $c_1$ and $l$, $c_2$ and $r$, or $c_1$ and $c_2$ in any of the arguments, this unified (abused) notation will not introduce inconsistencies.\(^2\) For the sake of notational convenience, we will use $h$ and $t$ to refer to the base vertices $b_1$ and $b_d$, respectively; the frame is then denoted by $(s : c_1,c_2 : l,h; t,r)$.

**Lemma 3.2.** In an AW a vertex is simplicial if and only if it is a terminal.

In time $O(n^3)$, we can find a minimal forbidden induced subgraph or asserts its nonexistence as follows. For a hole, we guess three consecutive vertices $\{h_1, h_2, h_3\}$, and then search for the shortest $h_1$-$h_3$ path in $G - \{N[h_2] \setminus \{h_1, h_3\}\}$. For an AW, we guess three independent vertices $\{t_1, t_2, t_3\}$, and for $i = 1, 2, 3$, search for the shortest path between the other two in $G - N[t_i]$. Since the AW found as such is the minimum

\(^2\)Albeit the frames in †- and ‡-AWs have different number of vertices and different number of edges, they both have 10 missed edges. Moreover, using our generalized notations, these edges are exactly the same. See Lemma 3.4 and its corollary.
We may prove the second assertion first. We show

\[ \hat{G} \]

Moreover, if \( G \) contains none of \( \{lc_2, c_1 r, ht, sh, st\} \), then the frame induces the same subgraph in \( G \) and \( \hat{G} \).

**Proof.** We may prove the second assertion first. We show \( \hat{G} \) cannot contain any of the other 5 missed edges \( \{sl, sr, lr, lt, hr\} \) in the frame. The insertion of edge \( sl \) will introduce a 4-hole \( (schl) \), which requires the insertion of at least one of \( sh \) and \( lc_2 \), which are not allowed. A symmetric argument apply to the edge \( sr \). The insertion of edge \( lt \) will introduce a 4-hole \( (lhc_2t) \), which requires the insertion of at least one of \( ht \) and \( lc_2 \), which are not allowed. A symmetric argument apply to the edge \( hr \). Now we are left with only \( lr \), whose insertion will introduce a 5-hole \( (chtltc) \) or 4-hole \( (lc_1c_2r) \) depending on the type of the AW. Every minimal set of edges that fill this hole need edges that have been excluded.

To make the terminals cease to form an AT, we have to insert an edge to connect one terminal and the defining path connecting other terminals. This makes a list of at most \( d + 7 \) edges, among which only \( \{sl, sr, lr, lt, hr\} \) are not included in (1). All of them are in the frame, and hence cannot be inserted without also inserting one of (1). This proves the first assertion and completes the proof.

The exclusion of edges \( \{lc_2, c_1 r, ht, sh, st\} \) will perpetuate the structure of the frame, which is hence called an *unchangeable frame*. In most case, the frame is the only structure in a long AW that concerns us.

The purpose of Lemma 3.4 is surely not trying to decrease the directions we need to branch on a long AW, from \( d + 7 \) or \( d + 5 \) to \( d + 3 \). Instead, we are after the structural information that turns out to be crucial. Of special significance is the edge \( ht \); though its insertion does not break this AT. This is formulated in the following corollary and visualized in Figure 2 Noting that as a consequence of \( h \neq t \), intervals \( l_h \) and \( l_t \) are disjoint.

![Figure 2: Interval rep. of an unchangeable frame.](image)

**Corollary 3.5.** Let \( (s : c_1, c_2 : l, h; t, r) \) be the frame of a long AW in graph \( G \). If an interval supergraph \( \hat{G} \) of \( G \) contains none of \( \{lc_2, c_1 r, ht, sh, st\} \), then in any interval representation of \( \hat{G} \), the interval for \( s \) is between intervals for \( h \) and \( t \).

A similar and straightforward check furnishes for each small AW a set of at most 6 edges, reduced from e.g., 12 candidate edges for a long-claw. The proof is deferred to the appendix for lack of space.

**Lemma 3.6.** For each small AW in a graph \( G \), there is a set of at most 6 edges, depicted as dashed edges in Figure 3 such that any interval supergraph of \( G \) contains at least one of them.
One should compare the edges used for 3-nets and 3-tents with those for long AWs in Lemma 3.4. It is worth noting that the threshold for the base length of a long AW, \( d \geq 4 \), is carefully chosen. A partial motivation is the following lemma. See also the proof of Lemma 5.1 (Section 3.2) for more motivations and Section 7 for a discussion.

**Lemma 3.7.** Let \( X = \{s : c_1, c_2 : l, h ; t, r\} \) be the frame of a long AW \( W \) in graph \( G \), and \( x \) be a neighbor of both \( h \) and \( t \). If \( x \neq s \), then there is a small AW in \( G[X \cup \{x\}] \), and any interval supergraph of \( G \) contains at least one of \( \{l_{c_2}, c_1 r, s h, s t, h t\} \) or \( s r \).

**Proof.** From the adjacencies between \( x \) and \( \{h, t, s\} \) it can be inferred \( x \notin W \). There is an \( l - r \) path \( (l h x t r) \) that avoids \( N[s] \), which replaces \( (l B r) \) in \( W \) to define a small AW. (See Figure 9, where the case \( N(x) \cap \{l, r\} = l \) is symmetric to the case \( N(x) \cap \{l, r\} = r \).) The second assertion ensues. \( \square \)

## 4 Modules

A subset \( M \) of vertices forms a module of \( G \) if all vertices in \( M \) have the same neighbors outside of \( M \). In other words, for any pair of vertices \( u, v \in M \), a vertex \( x \notin M \) is either adjacent to both or neither of \( u \) and \( v \). A brief inspection reveals that none of graphs in Figure 1 has a module \( M \) satisfying \( 1 < |M| < |V(G)| \), and this is true also for holes of length greater than 4. Recall that a minimal forbidden induced subgraph contains at least 4 vertices.

**Lemma 4.1.** Let \( M \) be a module of graph \( G \). If a subset \( X \) of vertices induces a minimal forbidden subgraph, then either \( X \subseteq M \), or \( |M \cap X| \leq 2 \), where equality only holds if \( X \) induces a 4-hole.

Naturally, one will surmise that a minimum interval supergraph \( \hat{G} \) of \( G \) will preserve modules of \( G \). Nonetheless, this does not hold true in general. Here we manage to show a slightly weaker version; by a connected module we mean a module that induces a connected subgraph. The way we prove this crucial theorem is by working on an interval representation \( J \) of the interval supergraph \( \hat{G} \); if \( M \) is not a module of \( \hat{G} \), then we modify \( \{l_v \mid v \in M\} \) to make a new interval representation \( J' \) that gives a strictly smaller interval supergraph \( \hat{G}' \) of \( G \). In one case we will use the project operation defined as follows. Given a set \( X \) of sub-intervals of \( [p, q] \) and another interval \( [p', q'] \), we project \( x \in X \) to \( x' \) by applying the mapping

\[
t \to \frac{q' - p'}{q - p}(t - p) + p'
\]

to both endpoints of \( x \). It is easy to verify that \( x' \) is a sub-interval of \( [p', q'] \), and more importantly, these two sets of intervals represent the same interval graphs.

**Theorem 4.2.** A connected module \( M \) of graph \( G \) remains a module in any minimum interval supergraph of \( G \).

**Proof.** Without loss of generality, we may assume \( |M| > 1 \) and \( N_G(M) \neq \emptyset \). Let \( \hat{G} = \langle V, \hat{E} \rangle \) be a minimum interval supergraph of \( G \) and \( J = \{I_v \mid v \in V\} \) be an interval representation of \( \hat{G} \). We define \( p = \min_{v \in M} \text{right}(v) \) and \( q = \max_{v \in M} \text{left}(v) \); in particular \( \text{right}(x) = p \) and \( \text{left}(y) = q \). Denote by \( N = \bigcap_{v \in M} N_{\hat{G}}(v) \) the set of common neighbors of \( M \) in \( \hat{G} \); by definition, \( N_G(M) \subseteq N \subseteq N_{\hat{G}}(M) \). The theorem can be formulated as \( N_{\hat{G}}(M) = N \). Suppose to its contrary, there exists a vertex \( z \in N_{\hat{G}}(M) \setminus N \), then we modify \( J \) into another interval set \( J' = \{I'_v \mid v \in V\} \). We argue that the interval graph \( \hat{G}' \) extracted from \( J' \) is a supergraph of \( G \) and
Figure 4, then there exists a vertex (not necessarily left, right) which implies \(|\mathcal{G}| \geq \sum_{x} 1\). For each \(u \in N\), we set \(I_u = I_u^\prime\). In the graph \(\tilde{G}^\prime\) represented by \(J^\prime\), the subgraph induced by \(M\) is a clique; the subgraph induced by \(V \setminus M\) is the same as \(\tilde{G} - M\); and \(M\) is completely connected to \(V\). This verifies \(G \subseteq \tilde{G}^\prime\). On the other hand, for any vertex \(v \in M\), from \(I_u^\prime \subseteq I_u^\prime\), it can be inferred \(N_{\tilde{G}}(v) \subseteq N_{\tilde{G}^\prime}(v)\); it follows that \(\tilde{G}^\prime \subseteq \tilde{G}\). By assumption, \(I_u^\prime(=I_u)\) is either left to \(q - \varepsilon\) or right to \(p + \varepsilon\); hence \(z\) in \(\tilde{G}^\prime\) and \(\tilde{G}^\prime \neq \tilde{G}\). Putting them together we prove the case \(p > q\).

Case 2. \(p < q\). Then \(x \neq y\). We have \(I_v \cap [p, q] \neq \emptyset\) for every \(v \in M\), and \([p, q] \subset I_u\) for every \(u \in N\). (See Figure 4). We construct \(J^\prime\) as follows. Let \(\ell\) be a point in \([p, q]\) such that \(K_\ell \setminus M\) has the minimum cardinality; without loss of generality, we may assume \(\ell\) is different from any endpoint of intervals in \(J\). For each \(v \in M\), we set \(I_v^\prime\) by projecting \(I_v^\prime\) from \([p, q]\) to \([\ell - \varepsilon, \ell + \varepsilon]\). For each \(u \in V \setminus M\), we set \(I_u^\prime = I_u\). Let \(G^\prime\) be represented by \(J^\prime\) and \(N^\prime = N_{\tilde{G}^\prime}(M)\). Observe that no interval in \(J^\prime\) has an endpoint in \([\ell - \varepsilon, \ell + \varepsilon]\); for each \(u \in V \setminus M\), the interval \(I_u^\prime(=I_u)\) contains \(\ell\) if and only if \([\ell - \varepsilon, \ell + \varepsilon] \subseteq I_u^\prime\). In other words, \(N^\prime = K_\ell \setminus M\). For each \(u \in N_{\tilde{G}^\prime}(M)\), the interval \(I_u^\prime(=I_u)\) contains \([p, q]\) which contains \([\ell - \varepsilon, \ell + \varepsilon]\) in turn; hence \(N_{\tilde{G}^\prime}(M) \subseteq N^\prime\). The subgraphs induced by \(M\) and \(V \setminus M\) are the same as \(G[M]\) and \(\tilde{G} - M\) respectively. It follows that \(G \subseteq \tilde{G}^\prime\).

It remains to show \(|\tilde{G}^\prime| < |\tilde{G}|\), which is equivalent to \(|E(\tilde{G})^\prime \cap (M \times V \setminus M)| < |E(\tilde{G}) \cap (M \times V \setminus M)|\), where \(|E(\tilde{G})^\prime \cap (M \times V \setminus M)| = |M\times N^\prime|\). By the selection of \(\ell\), for each point \(v \in [p, q]\), we have \(|K_\ell \setminus M| \geq |N^\prime|\). In other words, each vertex of \(M\) has at least \(|N^\prime|\) neighbors in \(\tilde{G}\), and thus \(|E(\tilde{G}) \cap (M \times V \setminus M)| \geq |M|\times |N^\prime|\), where equality only holds if \(|N_{\tilde{G}}(v)\setminus M| = |N^\prime|\) for every vertex \(v \in M\). We argue that this inequality has to be strict, which implies \(|\tilde{G}^\prime| < |\tilde{G}|\). As \(z \notin (N \cup M)\), it follows that \([p, q] \not\subset I_z^\prime(=I_z)\) (see the thick/red edges in Figure 4). If \(p < \text{right}(z) < q\) (see \(z_1\) in Figure 4), then \(|K_{\text{right}(z)} \setminus M| > |K_{\text{right}(z)} \setminus M| \geq |K_\ell \setminus M| = |N^\prime|\). As \(\tilde{G}[M]\) is connected, there exists a vertex \(v \in M\) such that \(\text{right}(z) \in I_v\), i.e., \(|N_{\tilde{G}[M]}(v)\setminus M| > |N^\prime|\). A symmetric argument applies if \(p < \text{left}(z) < q\). Hence we may assume there exists no vertex \(u \in V \setminus M\) such that \(I_u^\prime(=I_u)\) has an endpoint in \([p, q]\). As a consequence, \(K_{p-\varepsilon} \setminus M = K_p \setminus M = N^\prime\). If \(\text{right}(z) < p\) (see \(z_2\) in Figure 4), then there exists a vertex \(v \in M\) adjacent to both \(N^\prime\) and \(z\); e.g., the one with \(\text{left}(v) = \text{left}(M)\) (not necessarily \(x\)). A symmetric argument applies if \(\text{left}(z) > q\). This completes the proof.

Theorem 4.2 will ensure preservation of any connected module in perpetuity. With the help of Lemma 2.2 it can be further strengthened to:

**Corollary 4.3.** Let \(\tilde{G}\) be a minimum interval supergraph of graph \(G\). A connected module \(M\) of any graph \(\tilde{G}\) satisfying \(G \subseteq \tilde{G}^\prime \subseteq \tilde{G}\) is a module of \(\tilde{G}\).

The following theorem characterizes internal structures of modules, and can be viewed as a complement to Theorem 4.2 which characterizes the external behavior of modules in a minimum interval supergraph.

**Theorem 4.4.** Let \(\tilde{G}\) be a minimum interval supergraph of \(G\) and \(M\) be a connected module of \(\tilde{G}\). If \(\tilde{G}[M]\) is not a clique, then for any minimum interval supergraph \(G_M\) of \(G[M]\), replacing \(E(\tilde{G}[M])\) by \(E(G_M)\) in \(\tilde{G}\) gives a minimum interval supergraph \(\tilde{G}^\prime\) of \(G\); and in particular, \(\tilde{G}[M]\) is a minimum interval supergraph of \(G[M]\).

**Proof.** Let \(J = \{|v| \in V(\tilde{G})\}\) be an interval representation of \(\tilde{G}\). We define \(p = \min_{v \in M} \text{right}(v)\) and \(q = \max_{v \in M} \text{left}(v)\); in particular \(\text{right}(x) = p\) and \(\text{left}(y) = q\). As \(\tilde{G}[M]\) is not a clique, \(x \neq y\) and \(p < q\).
It follows that \( [p, q] \subseteq I_u \) for each \( u \in N_G(M) \) and \( I_u \) is disjoint from \( [\text{left}(M), \text{right}(M)] \) for \( u \notin N_G(M) \).

It is easy to verify that \( \hat{G} \) corresponds to the following interval representation obtained by modifying \( J \): we build an interval representation for \( \hat{G}_M \) and project it to \([\text{left}(M), \text{right}(M)]\) to replace \( I_v \) \( v \in M \).

Observing that \( \hat{G}[M] \) is an interval supergraph of \( G[M] \), if follows that \( |\hat{G}_M| \leq |\hat{G}[M]| \) and \( |\hat{G}'| \leq |\hat{G}| \). Since \( \hat{G} \) is minimum, both inequalities have to be equalities; this completes the proof.

According to Corollary 4.3 any connected module \( M \) of an intermediate graph between \( G \) and \( \hat{G} \) is a connected module of \( \hat{G} \), to which Theorem 4.4 applies. As a result, the subgraph induced by \( M \) in \( G \) is either a clique, or a minimum interval supergraph of \( G[M] \). In either case we have

**Corollary 4.5.** Let \( M \) be a connected module of graph \( G \). For any minimum interval supergraph \( \hat{G}_M \) of \( G[M] \), there exists a minimum interval supergraph \( \hat{G} \) of \( G \) such that \( \hat{G}_M \subseteq \hat{G}[M] \).

## 5 Shallow terminals in reduced graphs

We say a graph is **reduced** if it contains no hole or small AW\(^3\). As indicated by Lemma 3.4 the shallow terminal shall be of special interest during the disposal of a long AW. The following characterizations of shallow terminals were first proved on a class of less restricted graphs that excludes small AWs and small holes in [5]. Reduced graphs, excluding small AWs and all holes, are a trivial subset of them, whereby Lemma 5.1 and 5.2 apply to reduced graphs as well. For completeness, their proofs are repeated in Appendix.

**Lemma 5.1.** Let \( W \) be an AW with shallow terminal \( s \) and base \( B \) in a reduced graph, and \( x \) is adjacent to \( s \).

1. Then \( x \) is also adjacent to the center(s) of \( W \) (different from \( x \)).
2. Classifying \( x \) with respect to its adjacency to \( B \), we have the following categories:
   - (full) \( x \) is adjacent to every base vertex.
     - Then \( x \) is also adjacent to every vertex in \( N(s) \setminus \{x\} \).
   - (partial) \( x \) is adjacent to some, but not all base vertices.
     - Then there is an AW whose shallow terminal is \( s \), one center is \( x \), and base is a proper sub-path of \( B \).
   - (none) \( x \) is adjacent to no base vertex.
     - Then \( x \) is adjacent to neither base terminals, and thus replacing the shallow terminal of \( W \) by \( x \) makes another AW.

**Lemma 5.2.** Let \( W \) be an AW with shallow terminal \( s \) and base \( B \) in a reduced graph \( G \). Let \( C = N(s) \cap N(B) \) and \( M \) be the connected component of \( G - C \) containing \( s \). Then \( C \) induces a clique and is completely connected to \( M \).

Let \( \text{ST}(G) \) denote the set of shallow terminals of a reduced graph \( G \). The lemmas above indicate a nice structure for \( \text{ST}(G) \). Observe that for a module \( M \) in a chordal graph, at least one of \( M \) and \( N(M) \) induces a clique. We say \( M \) is **simplicial module** if it is a connected module and \( N(M) \) induce a clique. We remark that this name is suggested by Lemma 5.2 and the fact that any \( s \in M \) is simplicial in \( G - \{M \setminus \{s\}\} \).

**Lemma 5.3.** Let \( M \) be a simplicial module of a reduced graph \( G \). If an AW \( W \) contains a vertex \( x \in M \) and \( W \nsubseteq M \), then \( x \) is a terminal of \( W \). Moreover, if \( G \) is connected, then \( G - M \) is connected.

**Proof.** By Lemma 4.1, \( x \) is the only vertex in \( M \setminus W \), thus simplicial; the first assertion follows from Lemma 5.2. As \( N(M) \) induces a clique, if a pair of vertices \( u, v \notin M \) is connected by a path intersecting \( M \), then there is a \( u \rightarrow v \) path avoiding \( M \); the second assertion follows.

**Theorem 5.4.** In the subgraph induced by \( \text{ST}(G) \), each connected component makes a simplicial module of \( G \).

**Proof.** Given any AW, we can use Lemma 5.2 to construct the pair of sets \( M \) and \( C \); using definition we can verify that \( M \) makes a simplicial module of \( G \). If \( C \cap \text{ST}(G) = \emptyset \) then \( M \) is a connected component of the subgraph induced by \( \text{ST}(G) \). Hence we assume otherwise, and let \( W \) be another AW with shallow terminal \( s \in C \cap \text{ST}(G) \). As \( s \) is adjacent to every other vertex of \( C \cup M \), the base \( B \) of \( W \) is disjoint from \( C \cup M \); i.e., \( M \) is adjacent to \( s \) but not \( B \). The new set \( M' \) obtained by applying Lemma 5.2 on \( W \) contains both \( M \) and \( s \); it is also a simplicial module of \( G \). This process can be repeated for a finite number of steps, until a connected component of the subgraph induced by \( \text{ST}(G) \), also a simplicial module, is found.

---

\(^3\)On this ostensibly counterintuitive notion a remark is worthwhile. We reduce graphs by inserting edges; a reduced graph is thus a supergraph of the original graph. We use “reduced” in the sense that its structure is simpler, and its size is closer to \( \hat{G} \) than \( G \).
A reduced graph $G$ contains no hole or small AW; $\text{ST}(G)$ intersects every long AW, and thus $G - \text{ST}(G)$ is an interval graph. On the other hand, applying Lemma 5.5 repetitively on connected components of $\text{ST}(G)$ gives:

**Corollary 5.5.** If a reduced graph $G$ is connected, then $G - \text{ST}(G)$ is a connected interval graph.

To apply the results of this section we need to first find $\text{ST}(G)$. We check for each triple of vertices whether they form an AT or not, and identify an AW for them if yes. The AW is necessarily a long AW and contains a shallow terminal. Clearly it takes polynomial time to check all triples. The following lemma assures us that all shallow terminals can be found as such.

**Lemma 5.6.** In a reduced graph, all AWs with the same set of terminals have the same shallow terminal.

**Proof.** Let $(s : c_1, c_2 : l, h; t, r)$ be the frame of an AW $W$. We consider the distance between $l$ and $r$ in $G - N[s]$, which cannot be 1 by definition. As a shallow terminal is in distance either 2 or 3 to a base terminal, if the distance between $l$ and $r$ in $G - N[s]$ is strictly larger than 3, then this assertion must hold true. Suppose it is 2 and $x \not\in N[s]$ is a common neighbor of $l$ and $r$; clearly $x \not\in W$. As there cannot be a hole by assumption, $x$ must be adjacent to both $h$ and $t$. Noting $x \not\in s$, this contradicts Lemma 5.7. Suppose now it is 3 and $(lxyr)r$ is a shortest $l-r$ path in $G - N[s]$; then $l \not\in y$ and $x \not\in r$. It might happen that $x$ or $y$ is in $W$; in particular, $x = h$ or $y = t$, but not both. If $x, y \not\in W$, then $x, y$ make a cycle with the path $(lhc_1c_2tr)$. From the nonexistence of holes and the known adjacencies, it can be inferred that $x \sim h$ and $y \sim t$; and at least one of $x \sim t$ and $y \sim h$ holds true. If $x = h$ or $y = t$, then the other vertex is adjacent to both $h$ and $t$. Therefore, we always ends with a vertex in $N(h) \cap N(t) \setminus N(s)$, contradicting Lemma 5.7. This completes this proof.

It should be noted that this does not rule out the possibility of the shallow terminal of an AW being a base terminal of another AW; if this happens, these AWs necessarily have at least one different terminal. Indeed, for any AW not fully contained in $\text{ST}(G)$ in a reduced graph $G$, we can conclude from Theorem 5.4 and Lemma 5.5 that $\bullet$ its shallow terminal is in $\text{ST}(G)$; $\bullet$ its base terminals might or might not be in $\text{ST}(G)$; and $\bullet$ all other vertices are disjoint from $\text{ST}(G)$.

Finally, our branching shall be conducted on a “locally minimal” AW which can be found as follows.

**Lemma 5.7.** For any $s \in \text{ST}(G)$ in a reduced graph $G$, there is an AW whose shallow terminals is $s$ and whose base is completely connected to $N(s) \setminus \text{ST}(G)$. Moreover, such an AW can be found in polynomial time.

**Proof.** We start from any AW $W$ with shallow terminal $s$. We use Lemma 5.1 to categorize vertices in $N[s] \setminus \text{ST}(G)$ with respect to $W$. None of them cannot be in category “none,” as otherwise such a vertex is a shallow terminal and has to be in $\text{ST}(G)$. If every vertex in $N(s) \setminus \text{ST}(G)$ is in category “full,” then $W$ is already what we need and we are done. Hence let us assume $x$ is in category “partial,” then we have another AW with shallow terminal $s$ and a strictly shorter base. Applying this argument repeatedly will eventually procure an AW with shallow terminal $s$ such that every vertex in $N(s) \setminus \text{ST}(G)$ is in category “full.” It is easy to verify that this procedure can be implemented in polynomial time; this completes this proof.

## 6 The algorithm

Now we are ready to present the main algorithm and prove Theorem 1.1. Our basic strategy is a sandwich approach, which either inserts edges to $G$, or excludes some other edges by setting them as “avoidable.” As such we narrow the search space from both sides, until the objective graph is obtained.

The execution of the algorithm, an intermixed application of several branching rules, can be described as a search tree of which every node contains a pair $(G, A)$, where $A$ denotes the set of “avoidable” edges. On a non-leaf node of this tree, we make mutually exclusive decisions, each generating a different child node. Let $(G, A)$ and $(G', A')$ be contained in a parent node and, respectively, a child node in the search tree; it holds $G \subseteq G'$ and $A \subseteq A'$. We say an interval supergraph $G'$ of $G$ is feasible for $(G, A)$ if $E(G')$ is disjoint from $A$, even its size is larger than $|G| + k$. Any graph feasible for $(G', A')$ is also feasible for $(G, A)$.

We say an input instance $(G, k)$ of INTERVAL COMPLETION is a “YES” instance if the size of a minimum interval supergraph of $G$ is at most $|G| + k$; a “NO” instance otherwise. We define $n_{\text{mis}}$ to be the size of minimum interval supergraphs of the input graph for a “YES” instance; and $|G| + k$ for a “NO” instance. By definition, it always holds $n_{\text{mis}} \leq |G| + k$.

The way we prove the correctness of our algorithm is by showing if a node in the search tree has a feasible supergraph of size upper bounded by $n_{\text{mis}}$, then at least one of its children nodes has a feasible supergraph of the same size. For a “NO” instance, this holds trivially: no matter which edges are chosen to be inserted...
or forbidden in any step, the monotonicity of $G$ and $A$ ensures every path in the search tree faithfully ends with “NO.” Hence we can focus on “YES” instances, where the root node surely satisfies this condition. With inductive reasoning, we conclude that there is a leaf node containing an interval graph $\hat{G}$ of size $\text{mis}$. To such a leaf node there is a unique path from the root, and $\hat{G}$ is feasible for every node in the path.

On the complexity analysis, we focus on the number of leaves of the search tree the algorithm traverses. It is achieved by bounding the number of children nodes of a non-leaf node with respect to the decrease of measure. As we have no intention of optimizing the order of the polynomial factor, for the generation of a child node, we are satisfied if it can be executed in polynomial time.

To facilitate the recursive calls and inductive proofs, we augment the algorithm inputs and strengthen inductive hypothesis as follows. In addition to the graph $G$ and parameter $k$, our algorithm takes as inputs:

- $U$: a set of shallow terminals;
- $X$: a set of unchangeable frames; and
- $A$: a set of “avoidable” edges.

They are related as follows. Let $M$ be a connected component of $G[U]$. Within $X$ there is an unchangeable frame $(s : c_1, c_2 : l, h; t, r)$, denoted by $X(M)$, such that $s$ is the only vertex in $X(M) \cap M$, and $\{lc_2, c_1, r, ht, xh, xt|x \in M\} \subseteq A$ (Lemma 3.4). We will use $h(M)$, $t(M)$, and $s(M)$ to refer to the vertices $h, t,$ and $s$ respectively in $X(M)$. The set $A$ is disjoint from $E(G)$, and both of them increase only; and $|G| + k$ remains the same throughout. The original instance $(G, k)$ is supplemented with empty $U$, $X$, and $A$; it thus makes $(G, k, \emptyset, \emptyset, \emptyset)$. We define the measure as $m = k - |U|$.

Aside from the aforementioned condition on size, we delineate 6 other conditions that hold throughout. All the 7 invariants are summarized in Figure 5. It should be easy to verify the base case, $(G, k, \emptyset, \emptyset, \emptyset)$, satisfies all the conditions.

\begin{center}
\textbf{C1.} There is a minimum interval supergraph of $G$ that has size $\text{mis}$ and avoids $A$.

Let $M$ be a connected component of $G[U]$.\n
\textbf{C2.} $M$ is a simplicial module of $G$ and induces an interval graph. There is a frame $X(M)$ in $X$.

\textbf{C3.} An edge is inserted between $M$ and a vertex in $V(G) \setminus U$ only when $M$ is removed from $U$.

Let $(s : c_1, c_2 : l, h; t, r) = X(M)$.

\textbf{C4.} $\{lc_2, c_1, r, ht, xh, xt|x \in M\} \subseteq A$. ($X(M)$ is unchangeable.)

\textbf{C5.} In $X(M)$, $s$ is the only vertex in $M$, and $\{c_1, c_2, h, t\}$ is disjoint from $U$.

\textbf{C6.} Every vertex in $N(M) \setminus U$ is adjacent to both $h$ and $t$.

\textbf{C7.} There is an $h$-$t$ path in $G - N(M)$.
\end{center}

Figure 5: Invariants during our algorithm

The remainder of this section is devoted to presenting the algorithm and proving the following lemma.

**Lemma 6.1.** On input $(G, k, U, X, A)$ that satisfies all conditions C1-7, the algorithm runs in $6^{k-|U|} \cdot n^{O(1)}$. Moreover, at the exit of this algorithm, $(\perp)$: The algorithm returns a minimum interval supergraph of $G$, and $U = \emptyset$.

An immediate implication of C1-7 is the following termination condition, which enables us to prune many subtrees.

**Claim 1.** If $k < |U|$ holds in a node of the search tree, then any feasible solution to $(G, A)$ has size strictly larger than $|G| + k$.

**Proof.** By Lemma 3.4 and C7 for every $s \in U$, we need to insert at least one edge between $s$ and $V(G) \setminus U$.

The algorithm consists of two phases. Phase I partitions the graph into two disjoint interval subgraphs, while Phase II merges them. Phase I iteratively executes two procedures, until the required condition is achieved. Phase II runs one single procedure and only once. We now describe each procedure, analyze its runtime, and verify the inductive hypothesis. (See Figure 7 for an outline of the algorithm)
6.1 Phase I

The aim of this phase is to partition the graph into two disjoint interval subgraphs $G[U]$ and $G - U$. As $U$ always induces an interval graph throughout (C2), the focus shall be laid on $G - U$. Procedure 1 reduces $G - U$ by breaking all its holes and small AWs. Procedure 2 takes care of long AWs; since it works only on reduced graphs, but the disposal of one long AW might introduce holes and/or small AWs, between the disposal of two long AWs we need to rerun procedure 1.

Procedure 1. Reducing $G - U$. This procedure repeatedly finds a small AW or a hole, and uses Lemmas 3.3 or 3.6 respectively, to fill it. On a hole $H$, we branch on inserting one of the at most $4^{|H| - 3}$ minimal sets specified in Lemma 3.3 in each branch, the measure decreases by $|H| - 3$. On a small AW, we branch on inserting one of the 6 edges specified in Lemma 3.6 in each branch, the measure decreases by 1.

Case 1. If the pair $M_1$ and $x$ above is found, then we end this procedure. Otherwise we proceed as follows. Let $M = M' \cup \bigcup_{i=1}^{p} M_i$. Since for each $1 \leq i \leq p$, the set $M_i$ is a module of $G'$ (C2), $M_i \sim M'$ means every vertex in $M_i$ is adjacent to $M'$. In other words, $\bigcup_{i=1}^{p} M_i = N(M') \cap U = M \cap U$. The following proposition characterizes $M$.

Claim 2. The set $M$ is a simplicial module of $G$. For each $1 \leq i \leq p$, it contains a long AW whose frame is $X(M_i)$.

Proof. According to Theorem 5.4, $M'$ is a simplicial module of $G - U$. If $M'$ is not adjacent to $U$, then $M = M'$. Using definition we can also verify that $M$ is also a simplicial module of $G$, and the statement holds vacuously. Hence we may assume otherwise.

There is a frame $X(M_1)$ in $X$ (C2); let it be $(s : c_1, c_2 : l, h; t, r)$ where $s \in M_1$. By assumption, $M_1$ is adjacent to some vertex $v \in M'$, which is then adjacent to both $h$ and $t$ (C5). Noting that $h, t \not\in U$ (C5), we must have $h, t \in N_{G - U}(M')$. As $M'$ is a simplicial module of $G - U$, a vertex in $N_{G - U}(M')$ is adjacent to every other vertex in $N_{G - U}(M')$. From the nonadjacency of $h$ and $t$, we can conclude $h, t \in M'$, and they are completely connected to $N_{G - U}(M')$. Then $N_{G - U}(M')$ is completely connected to $s$, and also $M_1$ as $M_1$ is a module of $G$.

On the other direction, $N(M_1)$ induces a clique (C2). Consider any neighbor $v$ of $M_1$ in $V(G) \setminus M'$. It is adjacent to $M'$ and not contained in $U$; as $M'$ is a module of $G - U$, it follows that $v$ is completely connected to $M'$.

Arguments above also apply to $M_i$ for $2 \leq i \leq p$. As a result, if a vertex $x \in V(G) \setminus M$ is adjacent to $M'$, then it is adjacent to every vertex in $N(M') \cap U$, and vice versa. This verifies $M$ is a module of $G$. By definition, $M$ is connected, and $N(M) = N_{G - U}(M')$ as it is disjoint from $U$; therefore, $M$ is a simplicial module of $G$.

We have already shown $h$ and $t$ are in $M'$. We now consider other vertices of $X(M_1)$, i.e., $(c_1, c_2, l, r)$. As $l$ and $c_2$ are nonadjacent and are both adjacent to some vertex of $M'$, which is a subset of $M$, they have to be in $M$. A symmetric argument will imply $c_1, r \in M$. Thus, $M$ contains every vertex of $X(M_1)$. Finally let $P$ be the shortest $h$-$t$ path in $G - N(M_1)$ (C7). As $G - U$ is chordal, every inner vertex in $P$ is adjacent to both $c_1$ and $c_2$; hence also in $M'$. This completes the proof.

If the module $M$ does not induce an interval graph, then we make a recursive call to fill it. Specifically, we invoke our algorithm with $(G[M], k, U \cup M, \{X(M_1) \mid 1 \leq i \leq p\}, A \cap M^2)$. The following claim captures the validity of this invocation:
Claim 3. The tuple $(G|M), k, U \cap M, \{X(M_i)\}_{1 \leq i \leq p}, A \cap M^2)$ satisfies C1-7, where $\min$s is set to be the size of minimum interval supergraphs of $G[M]$.

Proof. If $A \cap M^2 = \emptyset$, then C1 holds vacuously. Otherwise, let $\hat{G}$ be a minimum interval supergraph of $G$ that avoids $A$ (C1). The subgraph $\hat{G}[M]$ is then not complete; according to Theorem 4.4, $\hat{G}[M]$ is a minimum interval supergraph of $G[M]$ and avoids $A \cap M^2$. C2-7 follow from Claim 2.

With an inductive reasoning, we may assume this invocation returns a minimum interval supergraph of $G[M]$; let it be $\hat{G}_M$. According to Corollary 3.5 and Lemma 2.1, every vertex in $M$ is in some minimal interval supergraphs of $G$. To show it makes an interval graph, it suffices to build an interval representation as follows. Without loss of generality, we may assume this invocation returns a minimum interval supergraph of $G[M]$.

In this juncture, the configuration becomes $(G', k' = k - k_M, U' = U \setminus M, X, A)$, where $G'$ differs from $G$ only in edges $M^2$, i.e., $G'[M] = \hat{G}_M$. It is easy to verify that C1-7 remain true.

By definition and Claim 2, $M$ is a set of shallow terminals of $G - (U \setminus M)$, and it remains a set of shallow terminals of $G' - U'$. We can take any vertex $s \in M$, and use Lemma 5.7 to find a long $AW$ with frame $(s : c_1, c_2 : l, h; t, r)$. We branch into 6 directions as follows:

- insert one of the 3 edges $(lc_2, c_1r, ht)$ and decrease $k$ by 1;
- insert either $h \times M$ or $t \times M$ and decrease $k$ by $|M|$ or
- add $M$ into $U$, $X(M) = \{s : c_1, c_2 : l, h; t, r\}$ into $X$, and $(lc_2, c_1r, ht, xt|x \in M \to A)$.

In each of the 6 directions, the measure decreased by at least 1: either $k$ decreases, or $|U|$ increases.

Claim 4. In at least one branch C1-7 remain true.

Proof. Let $\hat{G}$ be a minimum interval supergraph of $G$ that avoids $A$ (C1). If $\hat{G}$ contains one of the edges $(lc_2, c_1r, ht)$, or one of the sets $h \times M$ and $t \times M$, then C1 remains true at this branch. The edge(s) are inserted in $G - U$ only, and hence C2-7 remain true.

Hence we may assume $\hat{G}$ contains none of the specified edges, which implies C1. In this direction, $M$ is newly inserted to $U$ and they are nonadjacent. By Claim 2, C2-7 are satisfied on $M$. While no other vertices in $U$ are impacted; hence C2-7 remain true on them.

At the end of this procedure, if $G - U$ is already an interval graph, then we are done with Phase I and turn to Procedure 3 directly; otherwise we come back to Procedure 1.

6.2 Phase II

We are now at the second phase, where, a priori, both $G[U]$ and $G - U$ are interval graphs.

Procedure 3. Merging $U$ to $G - U$. We construct an interval representation $J$ for $G - U$. For each connected component $M$ of $G[U]$, both $h(M)$ and $t(M)$ are in $V(G) \setminus U (C3)$ and are nonadjacent. Assume without loss of generality, $I_{h(M)}$ goes left to $I_{t(M)}$: we define $p_M = \text{right}(h(M))$ and $q = \text{left}(t(M))$. Let $\ell$ be the point in $[p, q]$ that minimizes $|K_{\ell}|$ among them satisfying that $K_{\ell} \times M$ is disjoint from $A$. We insert edges to completely connect $M$ and $K_{\ell} \setminus N(M)$, and remove $M$ from $U$. We stop at $U = \emptyset$. If the total number of edges inserted is larger than $k$, then we return "NO"; otherwise the interval graph obtained.

Claim 5. The graph $\hat{G}$ obtained as above is a minimum interval supergraph of $G$ that avoids $A$.

Proof. To show it makes an interval graph, it suffices to build an interval representation as follows. Without loss of generality, we may assume $c$’s selected for different connected components are different and avoid any endpoint of $J$. We build an interval representation for $G[M]$, and project it to $[\ell - c, \ell + c]$. It is easy to verify this interval representation corresponds to $\hat{G}$.

To show it is minimum, we show the edges inserted to each connected component $M$ of $G[U]$ is minimum. According to Corollary 3.5 and Lemma 2.1, every vertex in $M$ is in some minimal $h(M) \times t(M)$ separator $S$ in any interval supergraph of $G$. This separator has to be a clique, and contain at least a minimal $h$-$t$ separator in the subgraph $G - U$. Therefore, we need to find some $h$-$t$ separator $S'$ in the subgraph $G - U$, and completely connect it to $M$ (C2). By the selection of $\ell$, we need to insert at least $M \times K_{\ell} \setminus N(M)$ edges to $M$. This verifies that $\hat{G}$ is minimum and completes the proof.
This procedure runs in polynomial time. For each connected component $M$ in $G[U]$, at least $|M|$ edges in $M \times V(G) \setminus U$ are inserted (C7). Therefore, the measure is non-increasing.

(C1) is ensured by Claim 5. Since $U = \emptyset$ after this step, C2-7 hold vacuously. Exit condition $\perp$ is also satisfied.

7 Concluding remarks

Theorem 5.4 only holds in graphs free of 3-nets and 3-tents (see Lemma 3.7 and especially Figure 9). Hence in the reduction step, we do away with them and make sure $d > 3$ in the remaining graph. Interestingly, they are also the largest $\dagger$- and $\ddagger$-AWs, respectively, that admit a 6-way branching (see Lemma 5.6 and especially Figure 3). This indicates we might have reached the limit of basic bounded search. And to further lower the exponential factor in the time complexity, new observations and approach are required. We leave it open for the existence of a sub-exponential algorithm and more rivetingly, a polynomial kernel.

We present the algorithm with the bare essentials of modules. One may insert one more preprocessing step to our algorithm. That is, we may compute a modular decomposition for the graph, and then on insertion of an edge between a module and others, we fill in an all-or-none manner. It might speed up the algorithm on graphs with many nontrivial modules. We leave this for later work on algorithmic engineering. As shown in Figure 6, there are interval supergraphs that do break disconnected modules, so the connected condition in Theorem 4.2 is essential. We remark that an alternative way is to replace “for any” by “there exists,” as in the following statement.

Lemma 7.1. For any module $M$ of graph $G$, there exists a minimum interval supergraph $\hat{G}$ of $G$ such that $M$ is a module of $\hat{G}$.

In the intermediate step of our algorithm, we have some edges forbidden. One should not confuse this with the INTERVAL SANDWICH problem [12, 11] (see also [8]). The latter generalizes INTERVAL SUPERGRAPH by imposing an arbitrary set $F$ of edges that are not allowed to be inserted. The crucial difference is that a minimum solution to an instance $(G, F)$ of INTERVAL SANDWICH is not necessarily a minimum supergraph of $G$. This explains why we use “avoidable” instead “forbidden” for our algorithm. The new challenge is surely that modules are not necessarily preserved, and our algorithm will cease to work. A natural question is, can we adapt our algorithm to work on INTERVAL SANDWICH?

Acknowledgment. I am grateful to Sylvain Guillemot for his careful reading of an early version of this manuscript and helpful comments.

References

[1] Amotz Bar-Noy, Reuven Bar-Yehuda, Ari Freund, Joseph Naor, and Baruch Schieber. A unified approach to approximating resource allocation and scheduling. Journal of the ACM, 48(5):1069–1090, 2001.
[2] Seymour Benzer. On the topology of the genetic fine structure. Proceedings of the National Academy of Sciences, 45(11):1607–1620, 1959.
[3] Stéphane Bessy and Anthony Perez. Polynomial kernels for proper interval completion and related problems. In Owe et al. [20], pages 229–239.
[4] Leizhen Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. Information Processing Letters, 58(4):171–176, 1996.
[5] Yixin Cao and Dániel Marx. Interval deletion is fixed-parameter tractable. http://arxiv.org/pdf/1211.5933, 2012.
[6] Yixin Cao and Dániel Marx. A combinatorial algorithm for chordal deletion. Manuscript, 2013.
[7] Rodney G. Downey and Michael R. Fellows. Parameterized Complexity. Springer, 1999.

[8] Fedor V. Fomin and Yngve Villanger. Subexponential parameterized algorithm for minimum fill-in. In Yuval Rabani, editor, SODA, pages 1737–1746. SIAM, 2012. Full version available at http://arxiv.org/pdf/1104.2230v1.pdf.

[9] Fedor V. Fomin and Yngve Villanger. Searching for better fill-in. In Natacha Portier and Thomas Wilke, editors, STACS, volume 20 of LIPIcs, pages 8–19. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.

[10] Michael R. Garey and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman, San Francisco, 1979.

[11] Martin Charles Golumbic, Haim Kaplan, and Ron Shamir. Graph sandwich problems. Journal of Algorithms, 19(3):449–473, 1995.

[12] Martin Charles Golumbic and Ron Shamir. Complexity and algorithms for reasoning about time: A graph-theoretic approach. Journal of the ACM, 40(5):1108–1133, 1993.

[13] György Hajós. (problem 65) über eine art von graphen. Internationale Mathematische Nachrichten, 11, 1957.

[14] Pinar Heggernes, Pim van’t Hof, Bart M. P. Jansen, Stefan Kratsch, and Yngve Villanger. Parameterized complexity of vertex deletion into perfect graph classes. In Owe et al. [20], pages 240–251.

[15] Haim Kaplan, Ron Shamir, and Robert E. Tarjan. Tractability of parameterized completion problems on chordal, strongly chordal, and proper interval graphs. SIAM Journal on Computing, 28(5):1906–1922, 1999. Preliminary version appeared in FOCS 1994.

[16] David George Kendall. Incidence matrices, interval graphs and seriation in archaeology. Pacific Journal of Mathematics, 28:565–570, 1969.

[17] C. G. Lekkerkerker and J. Ch. Boland. Representation of a finite graph by a set of intervals on the real line. Fundamenta Mathematicae. Polska Akademia Nauk, 51:45–64, 1962.

[18] Dániel Marx. Chordal deletion is fixed-parameter tractable. Algorithmica, 57(4):747–768, 2010.

[19] Assaf Natanzon, Ron Shamir, and Roded Sharan. A polynomial approximation algorithm for the minimum fill-in problem. SIAM Journal on Computing, 30(4):1067–1079, 2000. Preliminary version appeared in STOC 1998.

[20] Olaf Owe, Martin Steffen, and Jan Arne Telle, editors. Fundamentals of Computation Theory - 18th International Symposium, FCT 2011, Oslo, Norway, August 22-25, 2011. Proceedings, volume 6914 of LNCS. Springer, 2011.

[21] Donald J. Rose. A graph-theoretic study of the numerical solution of sparse positive definite systems of linear equations. In Ronald C. Reed, editor, Graph Theory and Computing, pages 183–217. Academic Press, New York, 1972.

[22] Pim van’t Hof and Yngve Villanger. Proper interval vertex deletion. Algorithmica, 65(4):845–867, 2013.

[23] Yngve Villanger, Pinar Heggernes, Christophe Paul, and Jan Arne Telle. Interval completion is fixed parameter tractable. SIAM Journal on Computing, 38(5):2007–2020, 2009. Preliminary version appeared in STOC 2007 under title “Interval completion with few edges”.

[24] Mihalis Yannakakis. Computing the minimum fill-in is NP-complete. SIAM Journal on Algebraic and Discrete Methods, 2(1):77–79, 1981.
A Outline of the main algorithm

Algorithm Interval-Completion(G, k, U, X, A)
input: graph G, integer k, set U of shallow terminals, set X of frames, and set A of forbidden edges.
output: a minimum interval supergraph of G of size \(|G| + k|A|\) and avoiding A; or “NO.”

PHASE I. \(\ll\) return “NO” when \(k < |U|\) or some step has to use “avoidable” edge(s).
while \(G - U\) is not an interval graph do
1. use Lem. 9.3 and 9.5 to reduce \(G - U\);
2. if \(G - U\) is an interval graph then goto Phase II;
3. pick a connected component \(M'\) in \(\text{ST}(G - U)\);
   \(|M_1, \ldots, M_p| = N(M') \cap U\);
4. for \(i = 1\) to \(p\) do
   if there is a common neighbor \(x\) of \(h(M_i)\) and \(t(M_i)\) nonadjacent to \(s(M_i)\) then
   \(x = k - |M_i|\); \(U = U \setminus M_i\); goto step 1;
5. \(M = M' \cup \bigcup_{i=1}^{p} M_i;\)
6. \(\hat{G}_M = \text{Interval-Completion}(G|M), k, U \cap M, \{X(M_i)|1 \leq i \leq p\}, M \cap M^2;\)
7. if \(\hat{G}_M\) is “NO” then return “NO”;
else \(k = k - (|\hat{G}_M| - |G|M|); \)
\(\hat{G}_M; U = U \setminus M;\)
8. use Lem. 8.7 to pick an AW with frame \((s : c_1, c_2 : l, h; t, r)\) from \(G - U\), where \(s \in M;\)
do branching
   \(\ll\) only case 1 is used when \(k < |M|\);
   case 1: insert one edge of \((c_{2}, c_{1}, t, h, r);\)
   \(k = k - 1;\)
   case 2: insert edge set \((h \times M \lor t \times M;\)
   \(k = k - |M|;\)
   case 3: \(U = U \cup M;\)
   \(X(M) = \{s : c_1, c_2 : l, h; t, r\};\)
   update A;
PHASE II. \(\ll\) now both \(G[U]\) and \(G - U\) are interval graphs.
build interval representations for \(G - U;\)
for each connected component \(M\) in \(U\) do
a. \(p = \text{right}(h(M))\) and \(q = \text{left}(t(M));\)
for \(\ell \not\in [p, q]\) such that no edge in \(K_{\ell} \times M\) is forbidden and \(K_{\ell}\) has the minimum size;
c. connect \(M\) and \(K_{\ell} \setminus N(M);\)
d. decrease \(k\) accordingly; remove \(M\) from \(U;\)
if \(k < 0\) then return “NO”; else return \(G.\)

Figure 7: Outline of algorithm for INTERVAL COMPLETION

B Omitted proofs

B.1 Proof of Lemma 3.6

The numbers of edges that are eligible to break AIs witnessed by small AWs in Figure 3 are 12, 8, 9, 10, 6, 7, and 8 respectively. Observe that small AWs always reveal symmetry property.

Figure 8: A minimum interval supergraph must contain a dashed edges.

Proof of Lemma 3.6 An edge must be inserted between one terminal and the defining path connecting other terminals. We redraw the graphs in Figure 8 and label the vertices for the easiness of references.

Long claw. The insertion of edge \(t_1v_2\) will introduce a 4-hole \((t_1v_1c_2v_2t_1)\). To break this hole we will need at least one of \(t_1c\) and \(v_1v_2\), which are both included. Symmetrical arguments apply to all of \((t_1v_3, t_2v_2, t_2v_3, t_3v_1, t_3v_2)\). The insertion of \(t_3t_2\) will introduce a 5-hole \((t_1v_1c_2v_3t_1)\). All the five edges required to break this hole is either included or previously discussed. Symmetric arguments apply to \(t_3v_2\) and \(t_3v_3\).

Whipping top. The insertion of edge \(t_1t_2\) will introduce a 4-hole \((t_1v_1t_2t_1)\). To break this hole we will need at least one of \(t_1t_2\) and \(t_2c\), which are both included. A symmetric argument applies to \(t_1t_3\). The insertion of edge \(t_2v_3\) will introduce a 4-hole \((t_2v_2c_3v_3t_2)\). To break this hole we will need at least one of \(t_2c\) and \(v_2v_3\), which are both included. A symmetric argument applies to \(t_2v_3\). The insertion of \(t_2t_3\) will introduce a 5-hole \((t_1v_1c_2v_3t_2)\). To break this hole we will need at least one of \(t_2c\), \(t_3c\), and \(v_2v_3\), which are all included.

2-Net. The insertion of edge \(t_1t_2\) will introduce a 4-hole \((t_1v_1v_2t_2t_1)\). To break this hole we will need at least one of \(t_1v_2\) and \(t_2v_1\), which are both included. Symmetric arguments apply to the other two edges \(t_2t_3\).
and \( t_3 t_1 \) that are not included.

1-Tent. The insertion of edge \( t_1t_2 \) will introduce a 4-hole \((t_1t_2v_1v_2t_1)\). To break this hole we will need at least one of \( t_1v_1 \) and \( t_2v_2 \), which are both included. Symmetric arguments apply to the other two edges \( t_2t_3 \) and \( t_3t_1 \) that are not included.

2-Tent. The insertion of edge \( t_1t_2 \) or \( t_1t_3 \) has the same affect for tents. The insertion of edge \( t_2t_3 \) will introduce a 4-hole \((t_2v_0v_1t_3t_2)\). To break this hole we will need at least one of \( t_2v_1 \) and \( t_3v_0 \). The insertion of the former makes \((t_1t_2t_3,v_1,v_2,v_3)\) a tent. As shown above, we need at least one of \((t_1v_1,t_2v_2,t_3v_3)\), which are all included. A symmetric argument applies to \( t_3v_0 \).

Arguments for 3-nets and 3-tents are word-for-word copy of that for long AWs in Lemma 3.4.

\[ \square \]

### B.2 Proof of Lemma 5.1

| \( q = p + 1 \) | \( q = p + 2 \) | \( q > p + 2 \) |
|---------------|---------------|---------------|
| \( p = 0 \)  | \( (xsb_1b) \) | \( (l,x,c,s,b_2,b_1) \) |
| \( p = 1 \)  | \( (l,b_1,x,s,c,b_2) \) | \( (l,b_1,s,b_3,b_2) \) |
| \( p > 1 \)  | \( (b_{p-2},b_{p-1},b_p,s,x,b_{p+2},b_{p+1}) \) | \( (b_{p-1},b_p,x,q,b_{q-1}) \) |
| \( q = p + 1 \) | \( (xsb_1b) \) | \( (l,x,c,s,b_2,b_1) \) |
| \( p = 1 \)  | \( (l,b_1,x,s,c,b_2) \) | \( (l,b_1,s,b_3,b_2) \) |
| \( p > 1 \)  | \( (b_{p-2},b_{p-1},b_p,s,x,b_{p+2},b_{p+1}) \) | \( (b_{p-1},b_p,x,q,b_{q-1}) \) |

\* : The vertex \( x \) is in category “none.”

\** : The vertex \( x \) would be in category “full” if \( q = d + 1 \).

\*** : A 4-hole \((xb_p b_{p+1} b_{p+2} x)\) would be introduced if \( x ~ b_{p+2} \).

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|}
\hline
\( q = p + 1 \) & \( q = p + 2 \) & \( q > p + 2 \) \\
\hline
\( p = 0 \) & \( (xsb_1b) \) & \( (l,x,c,s,b_2,b_1) \) \\
\( p = 1 \) & \( (l,b_1,x,s,c,b_2) \) & \( (l,b_1,s,b_3,b_2) \) \\
\( p > 1 \) & \( (b_{p-2},b_{p-1},b_p,s,x,b_{p+2},b_{p+1}) \) & \( (b_{p-1},b_p,x,q,b_{q-1}) \) \\
\hline
\end{tabular}
\caption{Structures used in the proof of Lemma 5.1 (category “partial”)}
\end{table}

Proof of Lemma 5.1 Suppose to the contrary of statement (1), and without loss of generality, \( x \neq c_2 \). If \( x ~ b_1 \) for some \( 1 \leq i \leq d \) then there is a 4-hole \((xsb_1b)\). Hence we may assume \( x \sim B \). There is \( \bullet \) a 5-hole \((xsb_1b)s\) or \((xsb_1b)\) if \( W \) is a \( 1 \)-AW, and \( x \sim 1 \) or \( x \sim r \), respectively; \( \bullet \) a 5-hole \((xsb_1b)\) or 4-hole \((xsb_2r)\) if \( W \) is a \( 1 \)-AW, and \( x \sim 1 \) or \( x \sim r \), respectively; \( \bullet \) a long-claw \((x,s,c,b_1,b_d,r)\) if \( W \) is a \( 1 \)-AW and \( x \sim 1 \) or \( r \); \( \bullet \) a net \((x,s,c_1,r,c_2)\) if \( W \) is a \( 1 \)-AW and \( x \neq c_1, l, r \); or \( \bullet \) a whipping top \((r,c_2,s,x,c_1,b_1)\) centered at \( c_2 \). Neither of these cases is possible, and thus statement (1) is proved.

For statement (2), let us handle category “none” first. Note that \( x \) nonadjacent to \( B \), cannot be a center of \( W \). If \( x \sim 1 \), then there is a 4-hole \((xsb_1b)\). A symmetrical argument will rule out \( x \sim r \). Now that \( x \) is adjacent to the center(s) but not base terminals nor base vertices of \( W \), then \((x,c_1,c_2 : l,B)\) makes another AW.

Assume now that \( x \) is in category “full.” Suppose the contrary and \( x \sim v \) for some \( v \in N(s) \setminus \{x\} \). We have already proved in statement (1) that \( v \) and \( x \) are adjacent to the center(s) of \( W \) (different from them). In particular, if one of \( v \) and \( x \) is a center, then they are adjacent. Therefore, we can assume that \( v \) and \( x \) are not centers. If \( v \sim b_1 \) for some \( 1 \leq i \leq d \), then there is a 4-hole \((xsb_1b)\). Otherwise, \( v \sim B \), and it is in category “none.” Let \( W' \) be the AW obtained by replacing \( s \) in \( W \) by \( v \); then by Lemma 3.7 \( x \sim v \) will imply the existence of small AW, which is impossible.

Finally, assume that \( x \) is in category “partial,” that is, \( x \sim B \), but \( x \sim b_1 \) for some \( 1 \leq i \leq d \). In this case, we construct the claimed AW as follows. As the case \( x \sim l \) but \( x \sim r \) is symmetric to \( x \sim l \) but \( x \sim r \); on the other hand, \( x \) is adjacent to both \( l \) and \( r \) will put it to category “full.” Hence in the following we may assume that \( x \sim r \). Let \( p \) be the smallest index such that \( x \sim b_p \), and \( q \) be the smallest index such that \( p < q \leq d + 1 \) and \( x \sim b_q \) (if \( q \) exists by assumptions). See Table 2 for the structures for \( 1 \)-AW and \( 1 \)-AW respectively. As the graph is reduced and contains no small forbidden induced subgraph, it is immediate from Table 2 that the case \( q > p + 2 \) holds; otherwise there always exists a small forbidden induced subgraph. This completes the categorization of vertices in \( N(s) \setminus T \).

\[ \square \]
B.3 Proof of Lemma 5.2

In this proof we will use \( W = (s : c_1, c_2 : l, B, r) \) to denote an AW.

Proof of Lemma 5.2 Let \( x \) and \( y \) be any pair of vertices such that \( x \in C \) and \( y \in M \). Since \( G[M] \) is connected by definition, we can find a shortest path \( P = (v_0 \ldots v_p) \) between \( v_0 = s \) and \( v_p = y \) in \( G[M] \). We claim that \( P \not\models B \). Suppose the contrary and let \( q \) be the smallest index satisfying \( v_q \models B \); note that \( q \geq 1 \). This means that every \( v_i \) with \( i < q \) is in category “none” of Lemma 5.1(2). Therefore, applying Lemma 5.1(1,2) on \( v_i \) and AW \((v_{i-1} : c_1, c_2 : l, B, r)\) inductively for \( i = 1, \ldots, q-1 \), we conclude that there is an AW \( W_i = (v_i : c_1, c_2 : l, B, r) \) for each \( i < q \). One more application of Lemma 5.1(1) shows that \( v_q \) is adjacent to the center(s) of \( W_{q-1} \) as well. If \( v_q \) is adjacent to all vertices of \( B \), i.e., in the category “full” with respect to every \( W_i \), then Lemma 5.1(2) on \( v_q \) and \( W_{q-1} \) implies that \( v_q \) is adjacent to \( v_{q-2} \in N(W_{q-1}) \), contradicting the assumption that \( P \) is shortest. Otherwise (the category “partial”), according to Lemma 5.1(2), there is another AW \( W' = (v_{q-1} : c'_1, c'_2 : l', B', r') \), where \( B' \subseteq B \), and \( v_q \in \{c'_1, c'_2\} \). Now an application of Lemma 5.1(1) on \( v_q \) and \( W' \) shows that \( v_q \) is adjacent to \( v_{q-2} \in N(W_{q-1}) \), again a contradiction. From these contradictions we can conclude \( P \not\models B \). Applying Lemma 5.1 inductively on \( v_{i+1} \) and \( W_i = (v_i : c_1, c_2 : l, B, r) \), we get an AW with the same centers for every \( 0 \leq i \leq p \).

As \( x \) is adjacent to both \( s \) and \( B \), it cannot be in category “none” with respect to \( W \). We now separate the discussion based on whether \( x \) is in the category “full” or “partial.” Suppose first that \( x \) is in the category “full”; as \( x \in N(s) \), Lemma 5.1(1) implies that \( x \models c_1, c_2 \). Then applying Lemma 5.1(2) inductively, where \( i = 1, \ldots, p \), on vertex \( x \) and \( W_{i-1} \) we get that \( x \models v_i \) for every \( i \leq p \); in particular, \( x \models v_p (= y) \). Suppose now that \( x \) is in in category “partial.” Then by Lemma 5.1(2), there is an AW \( W'_0 = (v_0 : c'_1, c'_2 : l', B', r') \), where \( B' \subseteq B \), and \( x \in \{c'_1, c'_2\} \). As \( P \not\models B \), we have that \( v_i \not\models B' \) for any \( 0 \leq i \leq p \), i.e., \( v_i \) is in category “none” with respect to \( W'_0 \). Therefore, by an inductive application of Lemma 5.1(2) on the vertex \( x \) and AW \( W'_{i-1} = (v_{i-1} : c'_1, c'_2 : l', B', r') \) for \( i = 1, \ldots, p \), we conclude that there is an AW \( W'_p = (v_p : c'_1, c'_2 : l', B', r') \), from which \( x \models y \) follows immediately.

Now we show the second assertion. For any pair of vertices \( x \) and \( y \) in \( C \), we apply Lemma 5.1 on \( x \) and \( W \); by definition, \( x \models B \) and thus cannot be in category “none.” If \( x \) is in category “full” with respect to \( W \), then Lemma 5.1(2) implies that \( x \) is adjacent to \( y \in N(s) \). Otherwise, if \( x \) is in category “partial” with respect to \( W \), then Lemma 5.1(2) implies that there is an AW \( W' = (s : c'_1, c'_2 : l', B', r') \) where \( B' \subseteq B \) and \( x \in \{c'_1, c'_2\} \). Therefore, by Lemma 5.1(1) on the vertex \( y \in N(s) \) and \( W' \), we get that \( y \models c'_1, c'_2 \) and hence \( x \models y \). 

C Omitted figures

Figure 9: Adjacency between a common neighbor \( x \) of \( B \) and \( s \). [Lem. 3.7]