Induced Subgraph Isomorphism on interval and proper interval graphs*

Pinar Heggernes† Daniel Meister‡ Yngve Villanger†

Abstract

The Induced Subgraph Isomorphism problem on two input graphs G and H is to decide whether G has an induced subgraph isomorphic to H. Already for the restricted case where H is a complete graph the problem is NP-complete, as it is then equivalent to the Clique problem. In a recent paper [10] Marx and Schlotter show that Induced Subgraph Isomorphism is NP-complete when G and H are restricted to be interval graphs. They also show that the problem is $W[1]$-hard with this restriction when parametrised by the number of vertices in H. In this paper we show that when G is an interval graph and H is a connected proper interval graph, the problem is solvable in polynomial time. As a more general result, we show that when G is an interval graph and H is an arbitrary proper interval graph, the problem is fixed parameter tractable when parametrised by the number of connected components of H. To complement our results, we prove that the problem remains NP-complete when G and H are both proper interval graphs and H is disconnected.

1 Introduction

Given two graphs G and H, where G has more vertices than H, the Induced Subgraph Isomorphism (ISI) problem is to decide whether G has an induced subgraph isomorphic to H. Equivalently, the question is whether we can delete vertices from G to obtain a graph isomorphic to H. ISI is a generalisation of several well known NP-complete problems like Clique, Independent Set, Longest induced path, and Graph Isomorphism, and it is thus NP-complete, as well as $W[1]$-hard when parametrised by the number of vertices in H.

As the problem is applicable in a variety of important practical areas [4], it has been studied with respect to polynomial-time solvability and fixed parameter tractability on restricted input graphs. ISI is solvable in polynomial time when G and H are both trees [11] or 2-connected outerplanar graphs [8], but it remains NP-complete when G is a tree and H is a forest [6], when G and H are both cographs [3], or when G is a cubic planar graph and H is a path [6]. When parametrised by the number of vertices in H, the problem is known to be fixed parameter tractable when G and H are planar [4] or have maximum degree bounded by a constant [2]. In a very recent paper by Marx and Schlotter, ISI is studied on interval graphs. When both G and H are interval graphs, the authors show that the problem is NP-complete and $W[1]$-hard when parametrised by the number of vertices in H [10].

*This work is supported by the Deutsche Forschungsgemeinschaft and by the Research Council of Norway.
†Department of Informatics, University of Bergen, Norway. Emails: pinar.heggernes@ii.uib.no, yngve.villanger@ii.uib.no
‡Theoretical Computer Science, RWTH Aachen University, Germany. Email: meister@cs.rwth-aachen.de

Here, we show that ISI is solvable in polynomial time when G is an interval graph and H is a connected proper interval graph. In fact, we give a more general result: when G is an interval graph and H is an arbitrary proper interval graph, ISI is fixed parameter tractable when parametrised by the number of connected components of H (and consequently also when parametrised by the number of vertices in H). To indicate that these results are the best that we can hope for, we show that ISI remains NP-complete when G and H are both proper interval graphs and H is disconnected. Note that such a result does not follow from any previous NP-completeness result, as the problems mentioned above (Clique, Independent Set, Longest Induced Path, etc) are all solvable in polynomial time on proper interval graphs.

To achieve our polynomial-time algorithm, we give an intermediate algorithm for solving the following problem in polynomial time: Given two connected interval graphs G and H with a partial ordering of the vertices of each graph, is there an isomorphism between H and an induced subgraph of G that respects the given partial orderings? This enables us to solve the problem when H is a connected proper interval graph and no ordering for G is given, since proper interval graphs have very specific such orderings. Our main result is obtained by showing that if H is isomorphic to an induced subgraph G' of G then the relative ordering of the vertices of G' is of a restricted type "fitting" the ordering of H, in any interval ordering for G.

Interval graphs form one of the best studied graph classes as they are used for modeling problems in many different application areas, like biology (in particular DNA computations and phylogeny), archaeology, and scheduling [7]. Proper interval graphs form an important and natural subclass of interval graphs, and they are also known as unit interval graphs or indifference graphs. Many NP-hard graph problems become solvable in polynomial time on interval graphs and even more on proper interval graphs. An example related to our problem is Graph Isomorphism which can be solved in linear time on interval graphs [1]. From this point of view, the mentioned results of Marx and Schlotter and our hardness result are surprising.

Due to space restrictions, the proofs of Lemma 3.1 and Theorem 3.5 are omitted, the proof of Lemma 3.3 is moved to the appendix, and all other proofs are shortened.

2 Definitions and notation

We consider simple finite undirected graphs. For a graph $G = (V, E)$, $V = V(G)$ is the *vertex set* of G and $E = E(G)$ is the *edge set* of G. For every edge $uv \in E$, vertices u and v are *adjacent* or *neighbours*. The *neighbourhood* of a vertex u in G is $N_G(u) =_{\mathrm{def}} \{v \mid uv \in E\}$, and the *closed neighbourhood* of u is $N_G[u] =_{\mathrm{def}} N_G(u) \cup \{u\}$. A set $X \subseteq V$ is called *clique* of G if the vertices in X are pairwise adjacent. A *maximal* clique is a clique that is not a proper subset of any other clique. For $U \subseteq V$, the *subgraph of G induced by U* is denoted by $G[U]$ and it is the graph with vertex set U and edge set equal to the set of edges $uv \in E$ with $u, v \in U$. For every $U \subseteq V$, $G' = G[U]$ is an *induced subgraph* of G. By $G \setminus X$ for $X \subseteq V$, we denote the graph $G[V \setminus X]$.

For two graphs G and H, G is *isomorphic* to H if there is a bijective mapping φ from $V(G)$ to $V(H)$ such that for every vertex pair u, v of G, $uv \in E(G)$ if and only if $\varphi(u)\varphi(v) \in E(H)$. Mapping φ is called an *isomorphism* from G to H. If G has an induced subgraph G' such that G' is isomorphic to H then we say that G has an induced subgraph isomorphic to H or, equivalently, H is isomorphic to an induced subgraph of G. Let us formally define the problem we are working on.

Induced Subgraph Isomorphism (ISI)
Input: Two graphs G and H.
Question: Does G have an induced subgraph that is isomorphic to H?

For a graph G, vertices u, v of G and an integer $k \geq 0$, a *u, v-path of length k* is a sequence $(u_0, \ldots, u_k)$ of $k+1$ distinct vertices of G such that $u_i u_{i+1} \in E(G)$ for $0 \leq i < k$ and $u_0 = u$ and $u_k = v$. A path $(u_0, \ldots, u_k)$ is *chordless* if $u_i u_j \notin E(G)$ for $0 \leq i < i+1 < j \leq k$. A graph G is *connected* if there is a u, v-path in G for every vertex pair u, v of G. A *connected component* of G is a maximal connected induced subgraph of G. The *distance* between two vertices u and v in G is the smallest integer k such that G has a u, v-path of length k.

A graph is an *interval graph* if intervals of the real line can be assigned to its vertices such that two vertices are adjacent if and only if their assigned intervals overlap. A *clique path* of a graph G is an ordering $\langle A_1, \ldots, A_k \rangle$ of the maximal cliques of G that satisfies the following for every vertex x of G: if $1 \leq p < q < r \leq k$ and $x \in A_p \cap A_r$ then $x \in A_q$. A graph is an interval graph if and only if it has a clique path [5]. A clique path can be constructed in linear time [5]. Note that an interval graph can have many different clique paths. An *proper interval graph* is an interval graph whose vertices can be assigned intervals such that no interval is properly contained in any other interval. A *claw* is a graph that is isomorphic to $K_{1,3}$. A graph is *claw-free* if it does not have a claw as an induced subgraph. Proper interval graphs are exactly the claw-free interval graphs [13].

A *vertex ordering* for a graph G is a linear ordering $\sigma = \langle u_1, \ldots, u_n \rangle$ of the vertices of G. For two vertices u_i, u_j of G in σ, we write $u_i \preccurlyeq_\sigma u_j$ if $i \leq j$. If additionally $i \neq j$ then we write $u_i \prec_\sigma u_j$. A vertex ordering σ for $G = (V, E)$ is called *interval ordering* if for every vertex triple u, v, w of G, $u \prec_\sigma v \prec_\sigma w$ and $uw \in E$ imply $vw \in E$. A graph is an interval graph if and only if it admits an interval ordering [12]. A vertex ordering σ for G is called *proper interval ordering* if for every vertex triple u, v, w of G, $u \prec_\sigma v \prec_\sigma w$ and $uw \in E$ imply $uv, vw \in E$. A graph is a proper interval graph if and only if it admits a proper interval ordering [9]. Interval orderings and proper interval orderings can be computed in linear time, if they exist.

3 Polynomial-time solvable cases of Induced Subgraph Isomorphism on interval graphs

We show that when G is an interval graph and H is a connected proper interval graph, ISI is solvable in polynomial time. From our intermediate results to reach this algorithm, it will follow that ISI is fixed-parameter tractable, parametrised by the number of connected components of H, when G is an interval graph and H is an arbitrary proper interval graph.

To obtain this result, we start by giving an intermediate result which is interesting on its own. In the first subsection we study the following problem: given two interval graphs G and H with clique paths $\langle A_1, \ldots, A_k \rangle$ and $\langle B_1, \ldots, B_l \rangle$ for G and H, respectively, decide whether there is an isomorphism from H to an induced subgraph of G that preserves the order of the maximal cliques given by the clique paths. We show that this problem is solvable in polynomial time.

3.1 Induced Subgraph Isomorphism on ordered interval graphs

We start by showing that any isomorphism between an interval graph and an induced subgraph of another interval graph must map maximal cliques of the two graphs to each other.

Lemma 3.1. *Let G and H be interval graphs with clique paths $\langle A_1,\ldots,A_k\rangle$ and $\langle B_1,\ldots,B_l\rangle$, respectively. If there is an isomorphism φ from H to an induced subgraph of G then there is a mapping $\psi:\{1,\ldots,l\}\to\{1,\ldots,k\}$ such that $\varphi(B_l)\subseteq A_{\psi(l)}$ and $\varphi(B_i\setminus B_{i+1})\subseteq A_{\psi(i)}\setminus A_{\psi(i+1)}$ for every $1\le i<l$ and $\psi(i)\neq\psi(j)$ for every $1\le i<j\le l$.*

This subsection considers isomorphisms that require $\psi(1)<\cdots<\psi(l)$ for function ψ of Lemma 3.1. We formalise this notion in the following way. Let G and H be graphs and let σ and τ be vertex orderings for respectively G and H. We say that H is (σ,τ)*-isomorphic* to an induced subgraph G' of G if there exists an isomorphism φ from H to G' such that $\varphi(u)\prec_\sigma\varphi(v)$ for every vertex pair u,v of H with $u\prec_\tau v$.

An interval graph $G=(V,E)$ may have many interval orderings. An interval ordering σ for G is a *preference interval ordering* if additionally the following condition is satisfied for every vertex triple u,v,w of G: if $u\prec_\sigma v\prec_\sigma w$ and $uw\in E$ and $uv\notin E$ then there is $x\in V$ such that $w\prec_\sigma x$ and $wx\in E$ and $vx\notin E$. (Informally, a preference interval ordering is a right endpoint ordering for an interval model where ties are broken by a left endpoint ordering.) Every interval graph has a preference interval ordering, and such an ordering can be computed in linear time. An interval graph can have many preference interval orderings. We need to relate preference interval orderings to clique paths and to (arbitrary) interval orderings. Let G be an interval graph with clique path $\langle A_1,\ldots,A_k\rangle$. A *preference interval ordering τ for G related to* $\langle A_1,\ldots,A_k\rangle$ satisfies for every vertex pair u,v of G and every $1\le i<k$ that $u\in A_i\setminus A_{i+1}$ and $v\in A_{i+1}\cup\cdots\cup A_k$ implies $u\prec_\tau v$. Note that such an ordering always exists. Let σ be an interval ordering for G. A *preference interval ordering τ for* (G,σ) satisfies for every vertex pair u,v of G that $uv\notin E$ and $u\prec_\sigma v$ implies $u\prec_\tau v$. It is important to see that also such an ordering always exists and it is also obtainable from a given clique path.

The first algorithm that we consider is called LocalOrderingInducedSubgraph, LOIS for short, and presented in Figure 1. This algorithm solves a restricted version of ISI on interval graphs, namely it requires the same number of maximal cliques for the two input graphs and additionally checks for ordered isomorphisms only.

Lemma 3.2. *Let G and H be interval graphs with the same number of maximal cliques. Let σ and τ be preference interval orderings for respectively G and H. Algorithm* LOIS *on this input computes a (σ,τ)-isomorphism from H to an induced subgraph of G, if it exists.*

Proof. Let $\sigma=\langle x_1,\ldots,x_n\rangle$ and $\tau=\langle y_1,\ldots,y_r\rangle$. We consider rounds of the **while** loop of the main procedure. Denote by a_i^e the value of a_i of the algorithm at the end of the eth round of the main **while** loop. It is not difficult to see that $a_1^e<\cdots<a_r^e$ for every $e\ge 0$. Assume that Algorithm LOIS does not reject, and let round f be the last round of the main **while** loop. Then, H is (σ,τ)-isomorphic to $G[\{x_{a_1^f},\ldots,x_{a_r^f}\}]$, so that $\varphi:V(H)\to V(G),y_i\mapsto x_{a_i^f}$ is a desired isomorphism. We show that Algorithm LOIS accepts if a desired isomorphism exists. Let $\langle A_1,\ldots,A_k\rangle$ and $\langle B_1,\ldots,B_k\rangle$ be the clique paths for respectively G and H that correspond to respectively σ and τ. Let $X_k=_{\rm def}A_k$ and $X_i=_{\rm def}A_i\setminus A_{i+1}$ for every $1\le i<k$.

Assume that there are integers $c_1,\ldots,c_r$ satisfying $1\le c_1<\cdots<c_r\le n$ such that H is (σ,τ)-isomorphic to $G[\{x_{c_1},\ldots,x_{c_r}\}]$. It follows from the definition that $c_i\le a_i^0$ for all $1\le i\le r$. We consider round $e+1$ and assume the claim be true for round e. There is a vertex pair y_p,y_q of H such that $y_py_q\in E(H)$ and $x_{a_p^e}x_{a_q^e}\notin E(G)$, or vice versa. We assume $p<q$. Due to the algorithm, q is largest possible such that such a vertex pair exists. We distinguish between the two possible cases. Let $y_py_q\notin E(H)$ and $x_{a_p^e}x_{a_q^e}\in E(G)$. Then, Subroutine push

Algorithm LocalOrderingInducedSubgraph (LOIS)
Input Graphs G and H with vertex orderings $\sigma = \langle x_1, \ldots, x_n\rangle$ and $\tau = \langle y_1, \ldots, y_r\rangle$, respectively.
Output An isomorphism φ from an induced subgraph G' to H such that for all vertex pairs u, v of G', $u \prec_\sigma v$ if and only if $u \prec_\tau v$, if such an isomorphism exists.

```
begin
    for  i = r  downto  1  do  let a_i = n − r + i  end for;
    while  H is not (σ,τ)-isomorphic to G[{x_{a_1}, ..., x_{a_r}}]  do
        let y_i, y_j be a vertex pair of H where i < j and j is largest possible
            such that y_i y_j ∉ E(H) ⇔ x_{a_i} x_{a_j} ∈ E(G);
        if  y_i y_j ∉ E(H)  then  PUSH(i)  else  PUSH(j)  end if
    end while;
    return a_1, ..., a_r and accept
end.

Subroutine PUSH(b)
begin
    set a_b = a_b − 1;
    while  a_b = a_{b−1}  and  b ≥ 2  do  set b = b − 1;  set a_b = a_b − 1  end while;
    if  a_1 = 0  then  reject  end if
end.
```

Figure 1: Algorithm LOIS.

is called with index p. There is an index t where $1 \leq t \leq p$ such that $a_i^{e+1} = a_i^e$ for all $1 \leq i < t$ and all $p < i \leq r$ and $a_i^{e+1} = a_i^e - 1$ for all $t \leq i \leq p$. Suppose that there is $1 \leq d \leq r$ such that $a_d^{e+1} < c_d$, which would contradict the claim. According to the assumption about $a_1^e, \ldots, a_r^e$, it follows that $t \leq d \leq p$. It holds that $a_i^{e+1} = a_{i-1}^e$ for all $t < i \leq p$. Furthermore, it follows that $c_d = a_d^e = a_d^{e+1} + 1$. The definition of Subroutine PUSH implies that $c_p = a_p^e = a_p^{e+1} + 1$. Thus, $c_q < a_q^{e+1}$. The properties of preference interval orderings show, since $x_{c_p} \prec_\sigma x_{c_q} \prec_\sigma x_{a_q^{e+1}}$, that there is a vertex x of G with $x_{a_q^{e+1}} \prec_\sigma x$ and $x_{c_q}x \notin E(G)$ and $x_{a_q^{e+1}}x \in E(G)$. Due to the choice of q as largest possible, it holds for every $q < j \leq r$ that $y_p y_j \in E(H) \Leftrightarrow x_{c_q} x_{a_j^e} \in E(G) \Leftrightarrow x_{a_q^e} x_{a_j^e} \in E(G)$. Now, the important observation to make is: for i, j such that $x_{c_q} \in X_i$ and $x_{a_q^e} \in X_j$, it holds that $i < j$. This, however, is not possible. As the other case, let $y_p y_q \in E(H)$ and $x_{a_p^e} x_{a_q^e} \notin E(G)$. If there is $1 \leq i \leq r$ such that $x_{a_i^{e+1}} < c_i$ then $c_q = a_q^{e+1} + 1 = a_q^e$. Since $x_{a_p^e} x_{c_q} \notin E(G)$, the properties of interval orderings imply that $z x_{c_q} \notin E(G)$ for every $x_1 \preccurlyeq_\sigma z \preccurlyeq_\sigma x_{a_p^e}$, in particular $x_{c_p} x_{c_q} \notin E(G)$, since $c_p \leq a_p^e$. This yields a contradiction. ∎

We extend the above problem to interval graphs with different numbers of maximal cliques. We apply a variant of Algorithm LOIS as a subroutine. The main difficulty is to determine the correct selection of maximal cliques of G. Our algorithm is called InducedIntervalSubgraph, IIS for short, and it is given in Figure 2. It applies as a subroutine Algorithm LOIS*; this algorithm is described in the next paragraph.

Algorithm LOIS* mainly works as Algorithm LOIS given in Figure 1. As an additional input, there is a lower bound on the value of a_i for each vertex of H. The return value of LOIS* is a vertex or a special symbol instead of the values of $a_1, \ldots, a_r$. During the initialization or the execution of Subroutine PUSH, there may be an a_i that becomes smaller than the corresponding given lower bound. If such a lower bound violation occurs during the initialization step, let m be

Algorithm InducedIntervalSubgraph (IIS)
Input An interval graph G with clique path $\langle A_1, \ldots, A_k \rangle$,
an interval graph H with clique path $\langle B_1, \ldots, B_l \rangle$.
Output **accept** if H is isomorphic to an induced subgraph of G under the extra conditions given in Lemma 3.3.

```
begin
    let τ be preference interval ordering for H related to ⟨B_1, ..., B_l⟩;
    let ψ(0) = 0;  let ψ(1) = 0;  let m = 1;  let B_{l+1} = ∅;
    loop
        INITIALIZE(m);
        let G' = G[A_ψ(1) ∪ ··· ∪ A_ψ(l)];  let σ be preference interval ordering for G' related to ⟨A_ψ(1), ..., A_ψ(l)⟩;
        for  i = 1  to  l  do
            for each x ∈ B_i \ B_{i+1}  do  λ(x) = |A_ψ(1) ∪ ··· ∪ A_ψ(i−1)| + 1  end for
        end for;
        set y = LOIS*(G', H; σ, τ; λ);
        if  y is a vertex of H  then  set m such that y ∈ B_m \ B_{m+1}  else  set m = 0  end if
    while m > 0;
    accept
end.
```

```
Subroutine INITIALIZE(s)
begin
    for  i = 1  to  s  do
        let p be smallest such that p > ψ(i − 1) and p > ψ(s);
        if  p does not exist  then  reject  end if;
        set ψ(i) = p
    end for
end.
```

Figure 2: Algorithm IIS.

the largest integer such that a_m is smaller than its corresponding lower bound. The algorithm stops and returns vertex y_m. Otherwise, the initialization step is executed successfully, and a lower bound violation can occur only during the execution of Subroutine push. Let y_i, y_j with $i < j$ be the (earliest) vertex pair of H for which a problem was encountered. If Subroutine push was called with i as parameter then the algorithm returns y_j, if push was called with j as parameter then the algorithm returns y_i. If no lower bound condition violation ever happens, which means that Algorithm LOIS* accepts, then the algorithm returns a special symbol. Note that LOIS already checks for a lower bound violation, namely it checks whether $a_1 < 1$ at the end of push. So, it is clear that if Algorithm LOIS would reject then Algorithm LOIS* will not return the special symbol.

Lemma 3.3. *Let G and H be interval graphs with $\mathscr{A} = \langle A_1, \ldots, A_k \rangle$ and $\mathscr{B} = \langle B_1, \ldots, B_l \rangle$ clique paths for respectively G and H. Algorithm* InducedIntervalSubgraph *on this input accepts if and only if there are integers $s_1, \ldots, s_l$ satisfying $1 \le s_1 < \cdots < s_l \le k$ such that H is (σ, τ)-isomorphic to an induced subgraph of $G[A_{s_1} \cup \cdots \cup A_{s_l}]$ where σ is a preference interval ordering related to $\langle A_{s_1}, \ldots, A_{s_l} \rangle$ and τ is a preference interval ordering related to $\mathscr{B}$.*

We determine the running time of IIS. The running time is mainly determined by the number of executions of the main loop of IIS and the running time of a single execution of LOIS*. Let graph G have n vertices. The main loop is executed at most n^2 times. Each single loop execution, including the re-initialization, requires $\mathcal{O}(n^2)$ time plus the time for an execution of LOIS*. The running time of this procedure is of order the running time of LOIS. The main **while** loop is executed at most n^2 times. A single loop execution requires $\mathcal{O}(n^2)$ time for checking the isomorphism condition and finding a new vertex pair. This sums up to a total running time of $\mathcal{O}(n^6)$ for IIS.

3.2 Finding induced proper interval subgraphs of interval graphs

In the previous subsection, we gave an algorithm that, given interval graphs G and H, decides whether H is isomorphic to an induced subgraph of G where an additional ordering condition had to be satisfied. This additional ordering condition seems to be necessary to obtain a polynomial-time algorithm when both G and H are interval graphs, as without the ordering condition the problem is NP-complete by the results of Marx and Schlotter [10].

In this section, we show that Induced Subgraph Isomorphism is polynomial-time solvable if G is an interval graph and H is a connected proper interval graph. We will simply apply Algorithm IIS for deciding the question. Part of the input for this algorithm are clique paths. Our decision problem can be solved by trying all possible combinations of clique paths. Interval graphs can have many clique paths, which would result in a worst-case exponential-time algorithm. Connected proper interval graphs, however, have at most two clique paths [13]. For our algorithm, it will be of high importance that the clique path for G can be chosen arbitrarily.

Theorem 3.4. *Given an interval graph G and a connected proper interval graph H, it can be decided in $\mathcal{O}(n^6)$ time whether G has an induced subgraph isomorphic to H.*

Proof. Let G be an interval graph, and let H be a connected proper interval graph. If H is isomorphic to some induced subgraph of G then there is a connected component C such that H is isomorphic to some induced subgraph of C. We therefore apply the below described algorithm to each connected component C of G. Let $\mathscr{A} = \langle A_1, \ldots, A_k \rangle$ and $\mathscr{B} = \langle B_1, \ldots, B_l \rangle$ be clique paths for respectively C and H. Denote by $\mathscr{B}^R$ the reverse of $\mathscr{B}$. Assume that H is isomorphic to an induced subgraph of C via isomorphism φ. Then, the restriction of $\mathscr{A}$ to the vertices that are mapped to by φ correspond to $\mathscr{B}$ or $\mathscr{B}^R$. This correspondence translates into the condition of Lemma 3.3 about the existence of an isomorphism from H to an induced subgraph of C. Therefore, it can be shown that H is isomorphic to an induced subgraph of C if and only if Algorithm IIS accepts on input $(C, H; \mathscr{A}, \mathscr{B})$ or on input $(C, H; \mathscr{A}, \mathscr{B}^R)$. The correctness of this statement strongly depends on the sketched relationship between $\mathscr{A}$ and $\mathscr{B}$. The algorithm for G accepts if IIS accepted for some connected component C; otherwise, it rejects. ■

For complementing the result of Theorem 3.4, we consider the case when input graph H disconnected. It can be shown that any isomorphism from H to an induced subgraph of given graph G maps the vertices of connected components consecutively with respect to any interval ordering for G. With this result and the algorithm of Theorem 3.4, the induced subgraph isomorphism problem can be solved in polynomial time when the order of the connected components of H is fixed with respect to an interval ordering for G. This implies the following result.

Theorem 3.5. *Given an interval graph G on n vertices and a proper interval graph H with r connected components, it can be decided in $\mathcal{O}(r! \cdot rn^6 \log n)$ time whether G has an induced subgraph isomorphic to H.*

Hence, when G is an interval graph and H is a proper interval graph, ISI is fixed-parameter tractable when parametrised by the number of connected components of H.

4 Induced Subgraph Isomorphism is NP-complete on proper interval graphs

In this section, we will show that the algorithms obtained in the previous section can be considered optimal: If the order of the connected components of H is not fixed then the problem becomes NP-complete already when both G and H are proper interval graphs and G is connected. We will obtain the completeness result by a reduction from a variant of the Hamiltonian Path problem.

Theorem 4.1 ([14]). *The* Fixed Hamiltonian Path *problem, given a graph G and a vertex pair u, v of G, to decide whether G has a u, v-path that is Hamiltonian, is NP-complete.*

Let G be a graph and let u, v be a vertex pair of G. We will construct a graph pair (F, H) such that F and H are proper interval graphs and H is isomorphic to an induced subgraph of F if and only if there is a u, v-path in G that is Hamiltonian. Let $u_1, \ldots, u_n$ be the vertices of G. Without loss of generality, we will assume $u_1 = u$ and $u_n = v$. The main idea of the construction is that a u, v-path of G that is Hamiltonian is a sequence of $n - 1$ edges where consecutive pairs are adjacent. Our two graphs will have the following tasks:

- F provides a list of all edges of G and a means for checking whether $n - 1$ selected edges form a sequence of the desired type

- H provides a mechanism for selecting $n - 1$ edges of G.

We begin with the construction of graph F. The graph is composed of subgraphs as shown in Figure 3. The figure shows two graphs, where the upper one is a *graph type*. The graph type has a complete graph on six vertices on its left end and a complete graph on seven vertices on its right end. The two complete graphs are joined by a graph that is a sequence of n triangles, then a chordless path between the vertices c and d and then another sequence of n triangles. The graphs of the depicted type differ from each other just in the length of the path between c and d. For an integer l with $1 - n \leq l \leq n - 1$, let M_l be the graph of the depicted graph type where the path between c and d has length $(8n^3 + 2) + (n + l - 1)(2n + 5)$. By N_l, we denote the induced subgraph of M_l that is obtained by deleting two vertices of minimum degree from each of the two complete graphs. So, N_l has a complete graph on four vertices at its left end, then a sequence of n triangles, a chordless path, another sequence of n triangles and finally a complete graph on five vertices. Now, let i, j be an integer pair where $1 \leq i, j \leq n$ and $i \neq j$. We define $F_{i,j}$ and $H_{i,j}$ as the following graphs:

$$\begin{aligned} F_{i,j} &=_{\text{def}} M_{j-i} \setminus \Big(\{a_1, \ldots, a_n, b_1, \ldots, b_n\} \setminus \{a_i, b_j\} \Big) \\ H_{i,j} &=_{\text{def}} N_{j-i} \setminus \Big(\{a_1, \ldots, a_n, b_1, \ldots, b_n\} \setminus \{a_i, b_j\} \Big). \end{aligned}$$

This means that the two complete graphs of $F_{i,j}$ and $H_{i,j}$ are connected by a long path that contains only two triangles, namely the ones formed with vertex a_i and with vertex b_j.

We consider the second (lower) graph in Figure 3; denote it by Q. We define an induced subgraph of Q for every vertex of G. Let $1 \leq i \leq n$:

$$Q_i =_{\text{def}} Q \setminus \Big(\{a_1, \ldots, a_n, b_1, \ldots, b_n\} \setminus \{a_i, b_i\} \Big).$$

We compound the graphs $Q_1, \ldots, Q_n$ to blocks, where we define three of them:

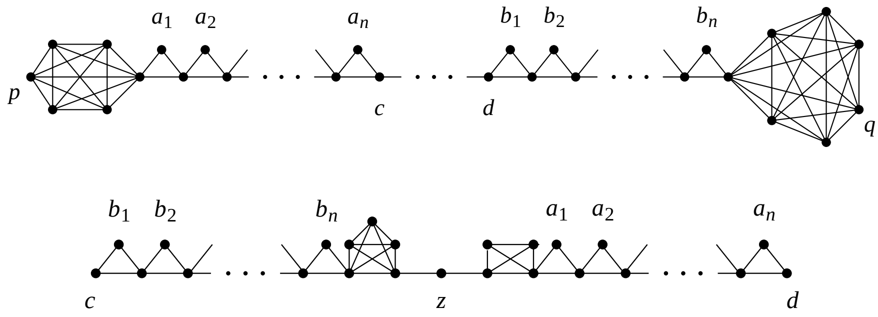

Figure 3: Depicted on the upper part is a graph type, where vertices c and d are connected by a chordless path of arbitrary length. The graph type is used for the construction of graphs F and H. The lower part of the figure depicts a graph, that we call Q in the construction of F.

a) *middle block*
join Q_i and Q_{i+1} by adding the edge between vertex d of Q_i and vertex c of Q_{i+1}, $1 \leq i < n$

b) *start block*
obtained from middle block by deleting all (remaining) vertices $a_2, \ldots, a_n, b_2, \ldots, b_n$ in $Q_2, \ldots, Q_n$

c) *end block*
obtained from middle block by deleting all (remaining) vertices $a_1, \ldots, a_{n-1}, b_1, \ldots, b_{n-1}$ in $Q_1, \ldots, Q_{n-1}$.

Let B_1 and B_n be respectively a start and an end block, and let $B_2, \ldots, B_{n-1}$ be $n-2$ copies of the middle block. We denote by c_l^m, z_l^m, d_l^m the vertices c, z, d, respectively, of Q_m in block B_l. Let $P = P_{8n^3+1}$ be a chordless path on $8n^3+1$ vertices. One vertex of degree 1 of P is called *start vertex*, the other one is called *end vertex*. Obtain F^* from $B_1, \ldots, B_n$ and $n-1$ copies $P_1, \ldots, P_{n-1}$ of P first as the disjoint union of $B_1, \ldots, B_n$ and $P_1, \ldots, P_{n-1}$ and second adding the edge between vertex d_i^n and the start vertex of P_i and the edge between the end vertex of P_i and vertex c_{i+1}^1 for all $1 \leq i < n$. For later arguments, it is important to observe that F^* is constructed from a long chordless path (with vertices c_1^1 and d_n^n as start and end vertex) by adding vertices, that are adjacent to exactly two adjacent vertices on the path.

We are ready for constructing graph F: F is the disjoint union of F^* and $F_{i,j}, F_{j,i}$ for every edge $u_i u_j \in E(G)$. Note that the number of connected components of F is $1 + 2|E(G)|$, since every edge of G is related to two connected components of F.

We continue with the construction of graph H. We define two new graphs, S and T:

s) S has a complete graph on six vertices that is connected to a sequence of n triangles; S is the induced subgraph of the upper graph type depicted in Figure 3 from the complete graph on the left hand side to vertex c

t) T has a complete graph on seven vertices that is connected to a sequence of n triangles; T is the induced subgraph of the upper graph type depicted in Figure 3 from the complete graph on the right hand side to vertex d.

For each $1 \le i \le n$, we define S_i and T_i as follows:

$$S_i =_{\text{def}} S \setminus \Big(\{a_1, \ldots, a_n\} \setminus \{a_i\} \Big) \quad \text{and} \quad T_i =_{\text{def}} T \setminus \Big(\{b_1, \ldots, b_n\} \setminus \{b_i\} \Big) .$$

With theses definitions, H is the disjoint union of $S_1, \ldots, S_{n-1}$ and $T_2, \ldots, T_n$ and $H_{i,j}, H_{j,i}$ for every $u_i u_j \in E(G)$. Note that the number of connected components of H is $2(n-1) + 2|E(G)|$.

Lemma 4.2. *G has a u, v-path that is Hamiltonian if and only if H is isomorphic to an induced subgraph of F.*

Proof. Due to space restrictions, we only show one direction of the claim. Let R be the chordless c_1^1, d_n^n-path in F^*. Assume that H is isomorphic to an induced subgraph of F; let φ be such an isomorphism. Since R contains less than $8n^4$ vertices, there are at most $n-1$ pairs (i,j) such that φ maps vertices of $H_{i,j}$ to vertices of F^*. Let $\mathscr{J}$ be the set of these pairs (i,j). Consider $S_1, \ldots, S_{n-1}, T_2, \ldots, T_n$ of H. Each of these graphs contains a complete graph on six vertices, and the largest clique of F^* has size 5. So, for each $X \in \{S_1, \ldots, S_{n-1}, T_2, \ldots, T_n\}$, there is a pair i, j such that φ maps the vertices of X to vertices of $F_{i,j}$. There is no pair i', j' such that $H_{i',j'}$ is mapped to $F_{i,j}$, since one of the two large cliques of $F_{i,j}$ is already occupied by vertices of X. Since every $F_{i,j}$ contains exactly two large cliques, at most two graphs from $\{S_1, \ldots, S_{n-1}, T_2, \ldots, T_n\}$ can be mapped to the same $F_{i,j}$. By the upper bound on the cardinality of $\mathscr{J}$, there exist $2|E(G)| - (n-1)$ pairs (i', j') such that $H_{i',j'}$ is mapped to $F_{i,j}$. Thus, $n-1$ pairs remain for components $S_1, \ldots, S_{n-1}, T_2, \ldots, T_n$ of H. Since $T_2, \ldots, T_n$ require a complete graph on seven vertices each, the $n-1$ graphs are mapped to pairwise different connected components of F, and therefore, also $S_1, \ldots, S_{n-1}$ are mapped to pairwise different connected components of F.

Denote by $V(R)$ the set of vertices on R. Let $\prec$ be the canonical ordering over $V(R)$ where $c_1^1 \prec d_n^n$. Let $(i,j) \in \mathscr{J}$. Assume that φ does not map q of $H_{i,j}$ to a vertex on R. Since $\varphi(q)$ belongs to a clique of size 5 of F^* and since every clique of F^* has size at most 5, a true twin of q is mapped to a vertex on R. Swapping the mapping for these two vertices results in an isomorphism from H to an induced subgraph of F that maps q to a vertex on R. It is not difficult to see that all vertices on the p, q-path of $H_{i,j}$ except for p must be mapped to a vertex on R by φ. If p is not mapped to a vertex from $V(R)$ but a true twin of p is then we apply the same swapping operation and conclude that we can assume for φ that all vertices of the p, q-path are mapped to vertices from $V(R)$. Otherwise, p is mapped to a vertex from a clique of size 5 in F^*, and by the structure of Q, all neighbours of p must be mapped to vertices of the same clique of size 5. It can be shown that this is not possible. We conclude that $\varphi(p)$ is a neighbour of a z-vertex in a maximal clique of size 4 and $\varphi(q)$ is a neighbour of a z-vertex in a maximal clique of size 5 and that $\varphi(p) \prec \varphi(q)$.

It follows from the above result that no vertex of H is mapped to any vertex x of F^* with $c_1^1 \preccurlyeq x \preccurlyeq z_1^1$ or $z_n^n \preccurlyeq x \preccurlyeq d_n^n$. Let $p_1, \ldots, p_{n-1}$ and $q_1, \ldots, q_{n-1}$ be the p- and q-vertices of the $H_{i,j}$ with $(i,j) \in \mathscr{J}$ where $\varphi(p_1) \prec \cdots \prec \varphi(p_{n-1})$ and $\varphi(q_1) \prec \cdots \prec \varphi(q_{n-1})$. With the above results, it holds that $z_1^1 \prec \varphi(p_1) \prec \varphi(q_1) \prec \varphi(p_2) \prec \cdots \prec \varphi(q_{n-1}) \prec z_n^n$. Remember that there is a vertex z for every $1 \le i \le n-2$ such that $\varphi(q_i) \prec z \prec \varphi(p_{i+1})$. We determine the distance between $\varphi(p_1)$ and $\varphi(q_{n-1})$. For this, we need to consider the pairs in $\mathscr{J}$. Due to the structure of $S_1, \ldots, S_{n-1}, T_2, \ldots, T_n$, it follows that

$$\{i : (i,j) \in \mathscr{J}\} = \{1, \ldots, n-1\} \qquad \text{and} \qquad \{j : (i,j) \in \mathscr{J}\} = \{2, \ldots, n\} .$$

Then, $\sum_{(i,j)\in\mathscr{J}} j-i = n-1$. For every $1 \le i \le n$, the distance between $\varphi(p_i)$ and $\varphi(q_i)$ is equal to the distance between p_i and q_i, which sums up to

$$(n-1)(8n^3+n(2n+5)-1)+(n-1)(2n+5) = 8n^4-6n^3+5n^2-3n-4\,.$$

Since $\varphi(q_i)$ and $\varphi(p_{i+1})$ are at distance at least 2, it follows for the distance between $\varphi(p_1)$ and $\varphi(q_{n-1})$:

$$(8n^4-6n^3+5n^2-3n-4)+2(n-2) = 8n^4-6n^3+5n^2-n-8\,.$$

Since the distances between c_1^1 and $\varphi(p_1)$ and between $\varphi(q_{n-1})$ and d_n^n are equal and at least $n+3$, it follows that the distance between $\varphi(p_1)$ and $\varphi(q_{n-1})$ is at most

$$(8n^4-6n^3+5n^2+n-2)-2(n+3) = 8n^4-6n^3+5n^2-n-8\,,$$

where the first number is the distance between c_1^1 and d_n^n. With the above calculated lower bound on the distance between $\varphi(p_1)$ and $\varphi(q_{n-1})$, we conclude that the two bounds match. Furthermore, the distance between $\varphi(q_i)$ and $\varphi(p_{i+1})$ for every $1 \le i \le n-2$ is exactly 2; in particular, q_i and p_{i+1} are neighbours of a z-vertex. Let (s_i, t_i) be the pair in $\mathscr{J}$ such that p_i, q_i are the vertices of H_{s_i,t_i}. Let $1 \le i \le n-1$ and let z_l^m be the vertex between $\varphi(q_i)$ and $\varphi(p_{i+1})$. It holds that z_l^m belongs to the copy of Q_m in block l. So, due to the definition of Q_m, it contains vertices a_m and b_m and it does not contain any of the vertices from $\{a_1,\ldots,a_n,b_1,\ldots,b_n\}\setminus\{a_m,b_m\}$. It follows from the definition of H_{s_i,t_i} and $H_{s_{i+1},t_{i+1}}$ that H_{s_i,t_i} contains b_m and $H_{s_{i+1},t_{i+1}}$ contains a_m, and this means that $t_i = s_{i+1}$. Since the start block contains only a_1, b_1 in Q_1 and since the end block contains only a_n, b_n in Q_n, it follows that $s_1 = 1$ and $t_{n-1} = n$. Due to the construction of H, G contains the edges $u_1u_{t_1}, u_{s_2}u_{t_2}, \ldots, u_{s_{n-1}}u_n$, which means that G contains a u_1, u_n-path that is Hamiltonian. ∎

Theorem 4.3. Induced Subgraph Isomorphism *is NP-complete when both input graphs are proper interval.*

Proof. NP-completeness of the problem follows from the NP-completeness of Fixed Hamiltonian Path due to Theorem 4.1 and the result of Lemma 4.2. Note that all constructed graphs are indeed proper interval graphs, since they are claw-free interval graphs. Furthermore, the defined graphs can be constructed in polynomial time. ∎

As a final remark, we want to point out that we can make F connected without changing more of the construction of F and H. We would modify F by connecting the connected components of F by chordless paths of length $8n^3+2n(2n+5) = 8n^3+4n^2+10n$. Thus, ISI is NP-complete even when input graph G is connected proper interval.

5 Conclusion

Concluding from our results, we can summarise the knowledge on the tractability of Induced Subgraph Isomorphism when input graph G is an interval graph as follows:

- If H is interval, it is NP-complete and $W[1]$-hard ([10]).
- If H is proper interval, it is NP-complete and fixed parameter tractable (this paper).
- If H is connected proper interval, it is polynomial-time solvable (this paper).

We would like to conclude with a couple of questions:

- What is the computational complexity of ISI when G is a chordal graph and H is a connected proper interval graph?
- For which subclasses $\mathcal{C}$ of proper interval graphs does ISI become polynomial-time solvable when G is interval and H is disconnected and belongs to $\mathcal{C}$?

References

[1] K. S. Booth and G. S. Lueker. A linear time algorithm for deciding interval graph isomorphism. *Journal of the ACM*, 26:183–195, 1979.

[2] L. Cai, S. M. Chan, and S. O. Chan. Random separation: A new method for solving fixed-cardinality optimization problems. *Proceedings IWPEC 2008, Lecture Notes in Computer Science*, 4169:239–250, 2006.

[3] P. Damaschke. Induced subgraph isomorphism for cographs is NP-complete. *Proceedings WG 1991, Lecture Notes in Computer Science*, 484:72–78, 1991.

[4] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.*, 3(3):1–27, 1999.

[5] D. R. Fulkerson and O. A. Gross. Incidence matrices and interval graphs. *Pacific Journal of Mathematics*, 15:835–855, 1965.

[6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*, W. H. Freeman & Co., New York, 1979.

[7] M. C. Golumbic. *Algorithmic Graph Theory and Perfect Graphs.* Annals of Discrete Mathematics Vol 57, Elsevier B.V., Amsterdam, 2004.

[8] A. Lingas. Subgraph isomorphism for biconnected outerplanar graphs in cubic time. *Theor. Comput. Science*, 63(3):295–302, 1989.

[9] P. J. Looges and S. Olariu. Optimal greedy algorithms for indifference graphs. *Computers & Mathematics with Applications*, 25:15–25, 1993.

[10] D. Marx and I. Schlotter. Cleaning Interval Graphs. arXiv:1003.1260v1, 2010.

[11] D. W. Matula. Subtree isomorphism in $O(n^{5/2})$. *Ann. Discrete Math.*, 2:91–106, 1978.

[12] S. Olariu. An optimal greedy heuristic to color interval graphs. *Information Processing Letters*, 37:21–25, 1991.

[13] F. S. Roberts. Indifference Graphs. In: *Proof Techniques in Graph Theory*, Academic Press, New York, 139–146, 1969.

[14] M. Sipser. *Introduction to the Theory of Computation.* International Thomson Publishing, 1996.

Appendix

Proof of Lemma 3.3. Let σ be a preference interval ordering related to $\mathscr{A}$, and let τ be a preference interval ordering related to $\mathscr{B}$. Let G' be an induced subgraph of G. Let σ'' be the restriction of σ to the vertices of G', and let σ' be a preference interval ordering for (G', σ''). Assume that H is (σ', τ)-isomorphic to G'. We apply Lemma 3.1 and obtain that there is a mapping $\psi : \{1, \ldots, l\} \to \{1, \ldots, k\}$ such that $\varphi(B_l) \subseteq A_{\psi(l)}$ and $\varphi(B_i \setminus B_{i+1}) \subseteq A_{\psi(i)} \setminus A_{\psi(i+1)}$ for every $1 \leq i < l$ and $\psi(1) < \cdots < \psi(l)$. The former properties follow from the result of the lemma, the latter property follows from the restriction to (σ', τ)-isomorphisms. Note also that this directly implies the correctness of the definition of the lower bounds λ. The task is to determine ψ. For our algorithm, it will be of importance to note that for any choice of ψ, $G[A_{\psi(1)} \cup \cdots \cup A_{\psi(l)}]$ is an interval graph with clique path $\langle A_{\psi(1)}, \ldots, A_{\psi(l)} \rangle$. In particular, $G[A_{\psi(1)} \cup \cdots \cup A_{\psi(l)}]$ has exactly l maximal cliques. The work of our algorithm can informally be described as follows: determine a set of l maximal cliques of G, check whether H is isomorphic to some induced subgraph, and if not, choose another set of l maximal cliques of G.

For the correctness proof, we consider the two cases. First, let ψ be any increasing mapping from $\{1, \ldots, l\}$ to $\{1, \ldots, k\}$, let $G' = G[A_{\psi(1)} \cup \cdots \cup A_{\psi(l)}]$, let σ' be a preference interval ordering related to $\langle A_{\psi(1)}, \ldots, A_{\psi(l)} \rangle$ and let λ be the lower bound function. If LOIS* on input $(G', H; \sigma', \tau; \lambda)$ returns the special symbol then, due to the definition of the algorithm, no vertex of H violates the lower bound condition during the execution of Algorithm LOIS*, and thus, H is (σ', τ)-isomorphic to some induced subgraph of G' due to Lemma 3.2. It is important to note that Algorithm LOIS* and the lemma are applicable, since H and G' have the same number of maximal cliques. This completes one direction of the correctness proof.

For the other direction of the correctness proof, we have to show that our algorithm finds an isomorphism of the desired form if there exists one. So, let there be integers $c_1, \ldots, c_l$ satisfying $1 \leq c_1 < \cdots < c_l \leq k$ such that H is (σ', τ)-isomorphic to some induced subgraph of $G' = G[A_{c_1} \cup \cdots \cup A_{c_l}]$ where σ' is a preference interval ordering related to $\langle A_{c_1}, \ldots, A_{c_l} \rangle$. Denote by $\psi^e(i)$ the value of $\psi(i)$ at the end of round $e \geq 1$ of the loop of the main procedure. It is not difficult to see that $\psi^e(1) < \cdots < \psi^e(l)$ for every $e \geq 1$. Due to the initialization step, it is clear that $\psi^1(i) \leq c_i$ for every $1 \leq i \leq l$. Now, consider an arbitrary round $e+1$ of the loop of the main procedure, and assume that $\psi^e(i) \leq c_i$ for every $1 \leq i \leq l$. Suppose that there is $1 \leq i \leq l$ such that $c_i < \psi^{e+1}(i)$. Since the algorithm did not terminate after round e of the loop, LOIS* did not return the special symbol, but a vertex y of H. Let m be such that $y \in B_m \setminus B_{m+1}$. From the definition of Subroutine INITIALIZE, it follows that $c_m < \psi^{e+1}(m)$, and therefore, $\psi^e(m) = c_m$. We have to distinguish between the two cases, why y was returned. If LOIS* failed already during the initialization step then $A_{\psi^e(m)} \setminus A_{\psi^e(m+1)}$ does not contain enough vertices to accommodate the vertices from $B_m \setminus B_{m+1}$. This contradicts the arguments of the first paragraph of the proof. So, LOIS* fails during a PUSH operation. Let u, v be the vertex pair of H for which the PUSH operation in LOIS* was called. Remember that $u = y$ or $v = y$. Let φ be as before calling PUSH in LOIS*. Due to the condition of the main **while** loop of LOIS*, it holds that $uv \in E(H)$ if and only if $\varphi(u)\varphi(v) \notin E(G')$. We assume $u \prec_\tau v$. Let b and b' be such that $\varphi(u) \in A_{\psi^e(b)} \setminus A_{\psi^e(b+1)}$ and $\varphi(v) \in A_{\psi^e(b')} \setminus A_{\psi^e(b'+1)}$. We distinguish between the two cases where the index of u or v was the parameter of PUSH. As the first case, assume that $uv \notin E(H)$. Then, PUSH was called with the index of u as parameter, and thus, $y = v$ and m is such that $v \in B_m \setminus B_{m+1}$. This means that $\psi^e(b') = c_m$. It is the fact that $\varphi(u)$ is "rightmost possible" in τ, which also means, it is "rightmost possible" in $A_{\psi^e(b)} \setminus A_{\psi^e(b+1)}$.

Since $\varphi(u)$ and $\varphi(v)$ are adjacent in G, $\varphi(v)$ is adjacent to every vertex in $A_{\psi^e(b)} \cup \cdots \cup A_{c_m}$, in particular, $\varphi(v)$ is adjacent to every vertex in A_{c_b}, which yields a contradiction.

As the second case, assume that $uv \in E(H)$. Then, PUSH was called with the index of v as parameter, and thus, $y = u$ and m is such that $u \in B_m \setminus B_{m+1}$. This means that $\psi^e(b) = c_m$ and $\varphi(v)$ is "leftmost possible" in $A_{\psi^e(b')} \setminus A_{\psi^e(b'+1)}$. And since $\varphi(v)$ is not adjacent to $\varphi(u)$ in G, $\varphi(v)$ is not adjacent to any vertex in $(A_1 \cup \cdots \cup A_{\psi^e(b)}) \setminus A_{\psi^e(b+1)}$. Since $\psi^e(b') \leq c_{b'}$, this yields a contradiction.

Since we have shown contradictions in all cases, it follows that the algorithm indeed finds a correct set of l maximal cliques of G to satisfy the claim. This completes the proof. ∎