Finding an Induced Subtree in an Intersection Graph is often hard*

Hidefumi Hiraishi[1], Dejun Mao[2], and Patrick Schnider[3]

1 **Graduate School of Information Science and Technology, The University of Tokyo**
`hiraishi1729@is.s.u-tokyo.ac.jp`

2 **Graduate School of Information Science and Technology, The University of Tokyo**
`maodejun001@is.s.u-tokyo.ac.jp`

3 **Department of Computer Science, ETH Zürich**
`patrick.schnider@inf.ethz.ch`

Abstract

We prove that the induced subtree isomorphism problem is NP-complete for penny graphs and chordal graphs as text graphs. As a step in the proofs, we reprove that the problem is NP-complete if the text graph is planar. For many other graph classes NP-completeness follows, as they contain one of the above three classes as a subclass, e.g., segment intersection graphs and coin graphs (contain planar graphs) or unit disk graphs (contain penny graphs).

1 Introduction

The Subtree isomorphism problem (STI) takes as input a graph G, called the *text graph*, and a tree T, called the *pattern graph*. The task is to determine whether G contains a copy of T, that is, a subgraph that is isomorphic to T. Clearly, this problem is in NP. It follows from the NP-completeness of Hamiltonian path that STI is NP-complete for all graph classes for which Hamiltonian path is hard. This includes many graph classes that we will talk about in this work, such as planar graphs, unit disk graphs and chordal graphs [2]. In some of the literature, such as [7, 9], STI is actually restricted to the case where G is a tree. In this case, STI allows a polynomial algorithm.

In contrast to STI, the *Induced Subtree isomorphism* (ISTI) asks whether a text graph G contains an *induced* copy of T. While this modification might seem rather minor at first glance, it can actually change the problem quite significantly. For example, on interval graphs, STI is NP-complete, whereas ISTI is tractable [4]. On the other hand, NP-completeness can still follow from the NP-completeness of Hamiltonian path, but only after subdividing each edge with an additional vertex (see e.g. [6]). The question then reduces to finding a path of length $2n-1$ in the new graph, which can in turn be reduced to Hamiltonian path. However, this reduction only works for graph classes that are closed under subdivision of edges. This includes planar graphs, but not penny graphs or chordal graphs.

In this abstract, we will show that ISTI is still hard for the two latter classes. For chordal graphs, this answers a question by Heggernes, van't Hof and Milanič [4]. We will also give an alternative proof that ISTI is hard for planar graphs. While our proof is arguably more complicated than the argument above, it is the basis for the other reductions. For many other classes of intersection graphs, NP-completeness of ISTI follows from our results.

* This work was initiated when the third author was visiting the University of Tokyo.

36th European Workshop on Computational Geometry, Würzburg, Germany, March 16–18, 2020.
This is an extended abstract of a presentation given at EuroCG'20. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

In general, an intersection graph is a graph whose vertices are subsets of some ground set and two vertices share an edge if and only if the corresponding sets intersect. The sets can be geometric objects such as straight-line segments in the plane, giving rise to *segment intersection graphs*, or disks in the plane, giving rise to *disk graphs*. For disk graphs, we can also enforce that any two disks have disjoint interiors (*coin graphs*), that all disks have the same radius (*unit disk graphs*), or both of these conditions (*penny graphs*). On the other hand, the sets can also be of a combinatorial nature. For example, the vertices could be subtrees of a given tree and two vertices share an edge if the corresponding subtrees share at least one vertex. These subtree intersection graphs are called *chordal graphs*. Another definition for chordal graphs is that every cycle of length at least 4 needs to have a chord, i.e., an edge that is not part of the cycle but connects two vertices of the cycle. These two definitions were shown to be equivalent by Gavril [3].

2 Planar graphs

We will use a reduction from a variant of 3-SAT, called Clause-Linked Planar 3-SAT. Given a CNF formula ϕ with clause set C and variable set V, the *incidence graph* $G_\phi = (C \cup V, E)$ is the graph that contains an edge between a variable and a clause if and only if the variable or its negation appear as a literal in the clause. We say that ϕ is *planar* if G_ϕ is a planar graph. The problem Planar 3-SAT is 3-SAT restricted to planar formulas. Planar 3-SAT is NP-complete [5]. We can enforce even more conditions without making the problem tractable: we say that a planar 3-CNF formula ϕ is *clause-linked* if there exists a path P (without additional vertices) connecting the clauses in $G(\phi)$ such that $G(\phi) \cup P$ is still a planar graph. Clause-Linked Planar 3-SAT, which is 3-SAT restricted to clause-linked planar formulas, is still NP-complete, see for example [8].

▶ **Theorem 2.1.** Induced Subtree Isomorphism *is NP-complete even when restricted to planar graphs.*

We will only describe the construction of the gadgets, the correctness is rather straight-forward.

Proof. It is straight-forward to see that ISTI is in NP, so the rest of the proof will be dedicated to prove NP-hardness. We will do this by reduction from Clause-Linked Planar 3-SAT. For any clause-linked planar 3-SAT formula ϕ we construct a planar graph H with the property that there is an assignment satisfying ϕ if and only if H contains an induced copy of a certain tree.

Let ϕ be a clause-linked planar 3-SAT formula and let $G(\phi)$ be its associated graph and P the path through its clauses. Consider a plane drawing of $G(\phi) \cup P$. We will mimic the formula ϕ by constructing subgraphs, called *gadgets*, that serve as variables, negations and clauses, and concatenating them according to the drawing of the graph $G(\phi)$. We start with the construction of the variable and negation gadgets.

Let v be a variable with k occurrences $(v, C_1), \ldots, (v, C_k)$ in ϕ, where the order of the occurrences is according to the rotational order of the respective edges in the drawing of $G(\phi) \cup P$. For an illustration of the construction, see Figure 1. We first construct a path $W = (w_1, w_2, \ldots, w_{2k})$ of length $2k$. We then add k vertices $x_1, \ldots, x_k$ and connect each x_i to w_{2i-1} and w_{2i}. Consider the star on 5 vertices (4 leafs). We call this star the *variable signature*. For each x_i, construct a copy of the variable signature and connect one of its leafs to x_i. Finally, extend a path $p_i = (w_i, p_{i,1}, \ldots, p_{i,l_1}, \ldots, p_{i,l})$ of length $l = l_1 + l_2$ from each w_i, where l_1 and l_2 are some large enough numbers (that are still polynomial in the number of variables), which we will define later. We call the vertices $p_{2i-1,l}$ and $p_{2i,l}$ the *endpoints* of (v, C_i). For every pair of such paths p_{2i-1}, p_{2i}, add two vertices a_i, b_i. If (v, C_i) is a positive occurrence, connect both a_i and b_i to p_{2i-1,l_1}, otherwise connect them to p_{2i,l_1}. We call such two vertices an *edge signature*.

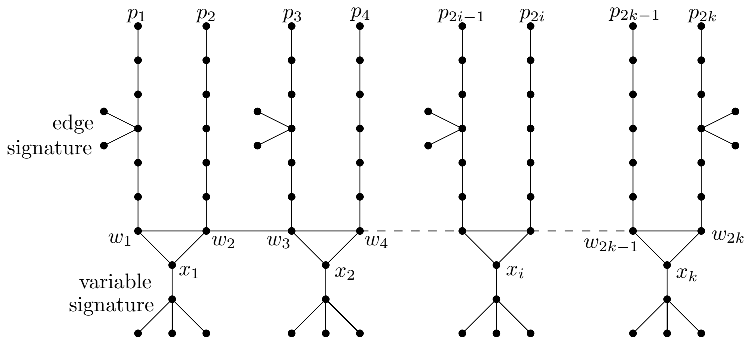

Figure 1 A variable gadget. The variable is negated in the clause C_k.

The idea behind this construction is that we will choose for each variable all the variable signatures and either the paths $p_1, p_3, \ldots, p_{2k-1}$ or the paths $p_2, p_4, \ldots, p_{2k}$ in the induced subtree. Also choosing the edge signature for a path will then mean that the variable is set to true. Note that every variable gadget admits a plane drawing in a disk of some radius such that only the paths p_i leave the disk and they do so according to the rotational order of the respective edges in the drawing of $G(\phi) \cup P$.

We will now construct the clause gadgets. For an illustration see Figure 2. Let $C_1, \ldots, C_m$ be the clauses in the order in which they appear along P. For every clause C_i, draw a vertex c_i and connect them with a path $c_1, \ldots, c_m$. For some clause C_i, let v_1, v_2 and v_3 be the variables occurring in C_i. Connect c_i to the endpoints of (v_1, C_i), (v_2, C_i) and (v_3, C_i). Again, this gadget admits a plane drawing that agrees with the rotational order around each C_i in the drawing of $G(\phi) \cup P$.

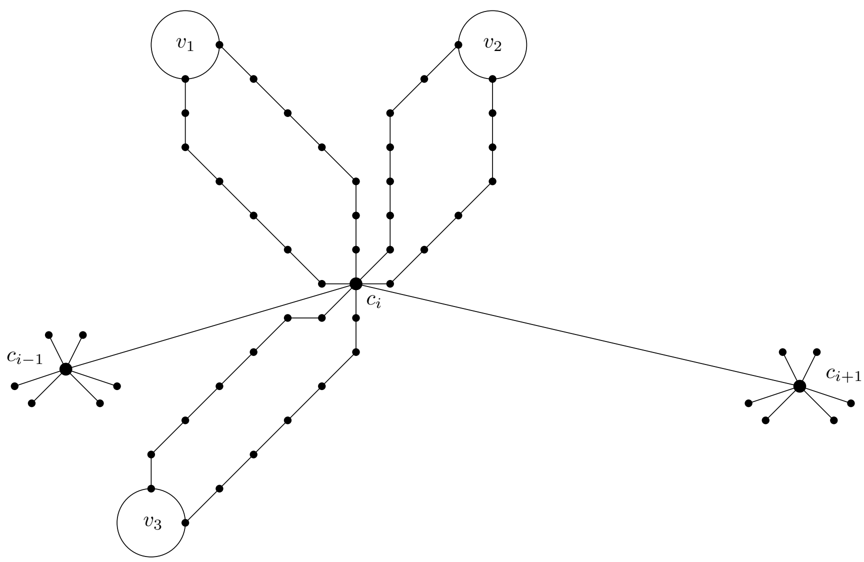

Figure 2 A clause gadget.

The union of all the gadgets will be the text graph H. As every gadget admits a planar drawing that agrees with the original rotational order, H admits a plane drawing. See Figure 3 for an example of the whole construction. We will now define our pattern graph T. Start with a path M of length m (the number of clauses). From each vertex of M, extend three paths of length l and connect each of them to a variable signature and exactly one of them to an edge signature.

It now follows from the construction that H contains an induced copy of T if and only if ϕ is satisfiable.

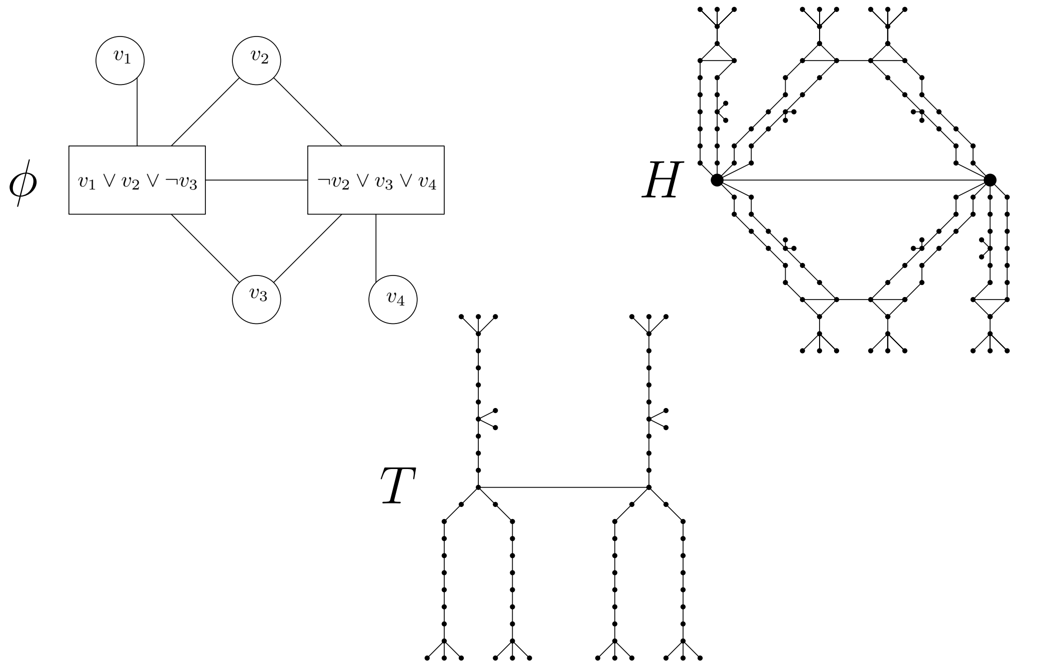

Figure 3 A drawing of a planar formula ϕ with the corresponding text graph H and pattern graph T.

◀

▶ **Corollary 2.2.** Induced Subtree Isomorphism *is NP-complete even when restricted to coin graphs or segment intersection graphs.*

Proof. The Koebe-Andreev-Thurston circle packing theorem states that every planar graph is a coin graph (and vice versa). Further, every planar graph is a segment intersection graph [1]. ◀

3 Penny graphs

In this section we will modify our reduction to fit penny graphs. It is again straightforward to check that ISTI is in NP, so we will only show NP-hardness. Again, we will only describe the construction of the gadgets, as all other arguments are analogous to the case of planar graphs.

▶ **Theorem 3.1.** Induced Subtree Isomorphism *is NP-complete even when restricted to penny graphs.*

Proof. We first show that we can draw unit disks with disjoint interiors such that their induced embedded penny graph coincides with the drawing of a variable gadget as constructed above. For an illustration of the construction see Figure 4. Place the centers of the disks the path W on a circle with very large radius, i.e., the disks will only cover a very small part of the circle. The disks corresponding to the x_i's and the variable signatures can then be placed inside the circle without any intersections. Further, the paths p_i can be placed outside the circle, and choosing l large enough, we can make enough space for the edge signatures. As for the clause gadgets, note that in our original construction, each vertex c_i is the center of an induced star on 9 vertices (8 leafs). However, the largest induced star induced star in a penny graph can have 6 vertices (5 leafs). So, instead of connecting the endpoints of p_{2i-1} and p_{2i} directly to c_i, we will connect them to a new vertex d and then connect this vertex to c_i by a path of some length l_3. Similarly, we will also connect c_i to c_{i+1} by a path of length l_4. Here, l_3 and l_4 are chosen large enough to have enough room the place the corresponding unit disks without unwanted intersections. Defining the pattern graph H accordingly, it is straight-forward to check that all the above arguments still go through.

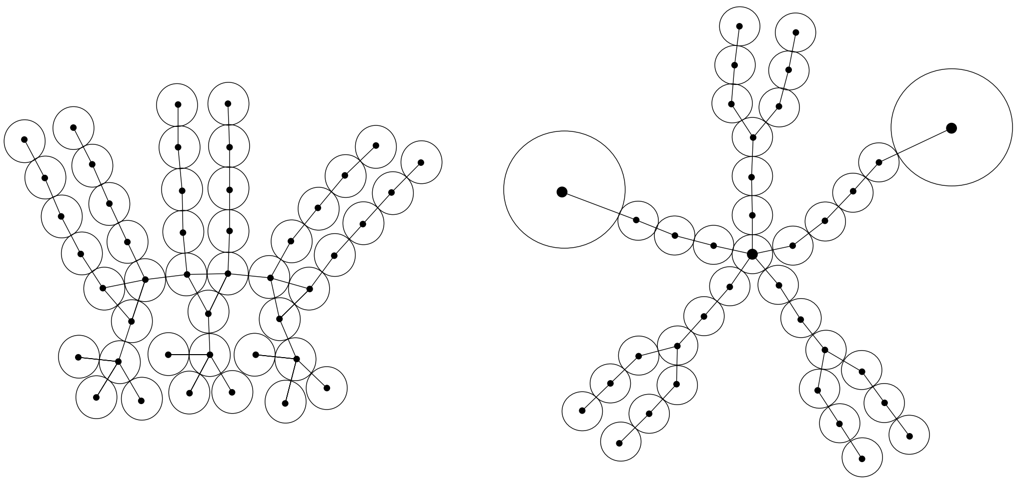

Figure 4 A variable gadget (left) and a clause gadget (right) for penny graphs.

◀

Clearly, every penny graph is a unit disk graph, so we immediately get the following corollary:

▶ **Corollary 3.2.** Induced Subtree Isomorphism *is NP-complete even when restricted to unit disk graphs.*

4 Chordal graphs

In this section we will modify our reduction to fit chordal graphs. It is again straightforward to check that ISTI is in NP, so we will only show NP-hardness. We will again only change the gadgets slightly, but we will include a large number of additional edges to make the graph chordal. As above, we will only describe the modifications in the construction.

▶ **Theorem 4.1.** Induced Subtree Isomorphism *is NP-complete even when restricted to chordal graphs.*

Proof. We make the two following modifications to the variable gadgets for planar graphs: first, we place an additional vertex y_i on the edge between w_i and w_{i+1}. So, the path W is now $w_1, y_1, w_2, y_2, \ldots, w_{2k-1}, y_{2k-1}, w_{2k}$. Secondly, instead of placing an edge signature at one place on a path p_i, we place one at every vertex. We further add an additional vertex for every vertex on some p_i, connecting it to only this vertex. We call these vertices *edge leafs*. The clause gadgets we extend by constructing a set of k paths of length 3 for each clause and connecting each of them to the respective c_i. We call these k 3-paths *clause signatures*.

Now, we will add chords to all cycles in our current graph. There are two types of cycles: (i) cycles defined by p_{2i-1} and p_{2i} for some variable and (ii) cycles inherited from cycles in $G(\phi) \cup P$. For (i), we add a complete bipartite graph between the vertices of p_{2i-1} and p_{2i}. Further, we connect all vertices of p_{2i-1} and p_{2i} to y_{2i-1}. As every such cycle must contain vertices from both p_{2i-1} and p_{2i} and possibly y_i, any such cycle will now have a chord. Similarly we add a complete bipartite graph between the vertices of p_{2i} and p_{2i+1} and connect all vertices of p_{2i} and p_{2i+1} to y_{2i}. This does not destroy any existing cycles, but it also does not introduce any chordless cycles. These edges will be helpful for the correctness proof, as they ensure that not both p_{2i} and p_{2i+1} can be chosen.

For (ii), recall that every edge in $G(\phi)$ corresponds to some pair of paths p_{2i-1} and p_{2i} for some variable. Further recall that all edges of P correspond to single edges in our constructed graph. Thus, every cycle in $G(\phi) \cup P$ that uses t edges of $G(\phi)$ induces 2^t cycles in our constructed graph. Further, each of these cycles in the constructed graphs will go through some vertices y_j and paths p_i. For every p_i and every y_j, we connect all vertices on p_i to y_j. Again, this puts a chord in every cycle.

Denote by H the chordal graph that we get from the above construction. We will now define our pattern graph T, which will be very similar to the pattern graph in the reduction for planar graphs. Start with a path M of length m (the number of clauses). From each vertex of M, connect it to a clause signature and three paths of length l. Add an edge leaf to each vertex on these paths. Finally, connect each of the paths to a variable signature and for exactly one path per clause, connect each vertex on the path to an edge signature.

Similar to above it can now be shown that H contains an induced copy of T if and only if ϕ is satisfiable.

We claim that H contains an induced copy of T if and only if ϕ is satisfiable. Finding a copy of T given a satisfying assignment of ϕ is analogous to the proof for planar graphs. For the other direction, we will again show that every induced copy of T corresponds to a satisfying assignment of ϕ. For this, we first note that T contains $3m$ disjoint paths of length l with edge leafs, each connected to a variable signature. On the other hand, H contains $6m$ paths of length l with edge leafs, always two of which (p_{2i-1} and p_{2i} for every variable) are connected to the same variable signature. In particular, every induced copy of T must contain either p_{2i-1} or p_{2i} and it cannot contain both. Further, due to the complete bipartite graph between p_{2i} and p_{2i+1}, T cannot contain both of them. Thus, T either contains $p_1, p_3, \ldots, p_{2k-1}$ or $p_2, p_4, \ldots, p_{2k}$. Again, we set the corresponding variable to TRUE in the first case, and to FALSE in the second case. Note that each the clause signatures ensure that c_i is the endpoint of three such paths and necessarily in T. By construction, each of these paths is connected to an edge signature if and only if the corresponding literal on the connected clause is positive. In particular, each c_i being incident to a path connected to an edge signature implies that under this assignment, each clause contains at least one positive literal, i.e., ϕ is satisfiable, which concludes the proof.

◀

References

1 Jérémie Chalopin and Daniel Gonçalves. Every planar graph is the intersection graph of segments in the plane. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 631–638. ACM, 2009.

2 H. N. de Ridder et al. Information system on graph classes and their inclusions. `www.graphclasses.org`, 2016.

3 Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47 – 56, 1974. URL: `http://www.sciencedirect.com/science/article/pii/009589567490094X`, `doi:https://doi.org/10.1016/0095-8956(74)90094-X`.

4 Pinar Heggernes, Pim van't Hof, and Martin Milanič. Induced subtrees in interval graphs. In *International Workshop on Combinatorial Algorithms*, pages 230–243. Springer, 2013.

5 D. Lichtenstein. Planar formulae and their uses. *SIAM Journal on Computing*, 11(2):329–343, 1982. `doi:10.1137/0211025`.

6 Jiří Matoušek and Robin Thomas. On the complexity of finding iso-and other morphisms for partial k-trees. *Discrete Mathematics*, 108(1-3):343–364, 1992.

7 David W. Matula. Subtree isomorphism in o(n5/2). In B. Alspach, P. Hell, and D.J. Miller, editors, *Algorithmic Aspects of Combinatorics*, volume 2 of *Annals of Discrete Mathematics*, pages 91 – 106. Elsevier, 1978. URL: `http://www.sciencedirect.com/science/article/pii/S0167506008703248`, `doi:https://doi.org/10.1016/S0167-5060(08)70324-8`.

8 A. Pilz. Planar 3-SAT with a clause/variable cycle. *CoRR*, abs/1710.07476, 2017. URL: `http://arxiv.org/abs/1710.07476`, `arXiv:1710.07476`.

9 Ron Shamir and Dekel Tsur. Faster subtree isomorphism. *Journal of Algorithms*, 33(2):267 – 280, 1999. URL: `http://www.sciencedirect.com/science/article/pii/S0196677499910441`, `doi:https://doi.org/10.1006/jagm.1999.1044`.