








Subgraph Isomorphism on Graph Classes that Exclude a Substructure

Hans L. Bodlaender^{1,2}, Tesshu Hanaka³, Yoshio Okamoto^{4,5},
Yota Otachi⁶, and Tom C. van der Zanden¹

¹ Utrecht University, Utrecht, The Netherlands
{H.L.Bodlaender,T.C.vanderZanden}@uu.nl

² University of Technology Eindhoven, Eindhoven, The Netherlands

³ Chuo University, Bunkyo-ku, Tokyo, Japan
hanaka.91t@g.chuo-u.ac.jp

⁴ The University of Electro-Communications, Chofu, Tokyo, Japan
okamotoy@uec.ac.jp

⁵ RIKEN Center for Advanced Intelligence Project, Tokyo, Japan

⁶ Kumamoto University, Kumamoto 860-8555, Japan
otachi@cs.kumamoto-u.ac.jp

Abstract. We study SUBGRAPH ISOMORPHISM on graph classes defined by a fixed forbidden graph. Although there are several ways for forbidding a graph, we observe that it is reasonable to focus on the minor relation since other well-known relations lead to either trivial or equivalent problems. When the forbidden minor is connected, we present a near dichotomy of the complexity of SUBGRAPH ISOMORPHISM with respect to the forbidden minor, where the only unsettled case is the path of five vertices. We then also consider the general case of possibly disconnected forbidden minors. We show in particular that: the problem is fixed-parameter tractable parameterized by the size of the forbidden minor H when H is a linear forest such that at most one component has four vertices and all other components have three or less vertices; and it is NP-complete if H contains four or more components with at least five vertices each. As a byproduct, we show that SUBGRAPH ISOMORPHISM is fixed-parameter tractable parameterized by vertex integrity. Using similar techniques, we also observe that SUBGRAPH ISOMORPHISM is fixed-parameter tractable parameterized by neighborhood diversity.

Keywords: Subgraph isomorphism · Minor free graphs · Parameterized complexity

Partially supported by NETWORKS (the Networks project, funded by the Netherlands Organization for Scientific Research NWO), the ELC project (the project Exploring the Limits of Computation, funded by MEXT), JSPS/MEXT KAKENHI grant numbers JP24106004, JP18K11168, JP18K11169, JP18H04091, JP18H06469, JP15K00009, JST CREST Grant Number JPMJCR1402, and Kayamori Foundation of Informational Science Advancement. The authors thank Momoko Hayamizu, Kenji Kashiwabara, Hiro-taka Ono, Ryuhei Uehara, and Koichi Yamazaki for helpful discussions. The authors are grateful to the anonymous reviewer of an earlier version of this paper who pointed out a gap in a proof.

1 Introduction

Let Q and G be graphs. A *subgraph isomorphism* η is an injection from $V(Q)$ to $V(G)$ that preserves the adjacency in Q ; that is, if $\{u, v\} \in E(Q)$, then $\{\eta(u), \eta(v)\} \in E(G)$. We say that Q is *subgraph-isomorphic* to G if there is a subgraph isomorphism from Q to G , and write $Q \preceq G$. In this paper, we study the following problem of deciding the existence of a subgraph isomorphism.

SUBGRAPH ISOMORPHISM

Input: Two graphs G (the *host* graph) and Q (the *pattern* graph).

Question: $Q \preceq G$?

The problem SUBGRAPH ISOMORPHISM is one of the most general and fundamental graph problems and generalizes many other graph problems such as GRAPH ISOMORPHISM, CLIQUE, HAMILTONIAN PATH/CYCLE, and BANDWIDTH. Obviously, SUBGRAPH ISOMORPHISM is NP-complete in general. When both host and pattern graphs are restricted to be in a graph class \mathcal{C} , we call the problem SUBGRAPH ISOMORPHISM *on* \mathcal{C} . By slightly modifying known reductions in [7, 14], one can easily show that the problem is hard even for very restricted graph classes. Recall that a linear forest is the disjoint union of paths and a cluster graph is the disjoint union of complete graphs. We can show the following hardness of SUBGRAPH ISOMORPHISM by a simple reduction from 3-PARTITION [14].

Proposition 1.1 (★¹). *SUBGRAPH ISOMORPHISM on linear forests and cluster graphs is NP-complete even if both graphs have the same number of vertices.*

Since most of the well-studied graph classes contain all linear forests or all cluster graphs, it is often hopeless to have a polynomial-time algorithm for an interesting graph class. This is sometimes true even if we further assume that the graphs are connected [19, 21]. On the other hand, it is polynomial-time solvable for trees [27]. This result was first generalized for 2-connected outerplanar graphs [24], and finally for k -connected partial k -trees [15, 26] (where the running time is XP parameterized by k). In [26], a polynomial-time algorithm for partial k -trees of bounded maximum degree is presented as well, which is later generalized to partial k -trees of log-bounded fragmentation [16]. It is also known that for chain graphs, co-chain graphs, and threshold graphs, SUBGRAPH ISOMORPHISM is polynomial-time solvable [19–21]. In the case where only the pattern graph has to be in a restricted graph class that is closed under vertex deletions, a complexity dichotomy with respect to the graph class is known [17].

Because of its unavoidable hardness in the general case, it is often assumed that the pattern graph is small. In such a setting, we can study the parameterized complexity² of SUBGRAPH ISOMORPHISM parameterized by the size of the pattern graph. Unfortunately, the W[1]-completeness of CLIQUE [9] implies

¹ A black star ★ means that the proof is omitted or shortened.

² We assume that the readers are familiar with the concept of parameterized complexity. See e.g. [6] for basic definitions omitted here.

that this parameterization does not help in general. Indeed, the existence of a $2^{o(n \log n)}$ -time algorithm for SUBGRAPH ISOMORPHISM is ruled out assuming the Exponential Time Hypothesis, where n is the total number of vertices [5]. So we need further restrictions on the considered graph classes even in the parameterized setting. For planar graphs, it is known to be fixed-parameter tractable [8, 11]. This result is later generalized to graphs of bounded genus [4]. For several graph parameters, the parameterized complexity of SUBGRAPH ISOMORPHISM parameterized by combinations of them is determined in [25]. In [3], it is shown that when the pattern graph excludes a fixed graph as a minor, the problem is fixed-parameter tractable parameterized by treewidth and the size of the pattern graph. The result in [3] implies also that SUBGRAPH ISOMORPHISM can be solved in subexponential time when the host graph also excludes a fixed graph as a minor.

1.1 Our Results

As mentioned above, the research on SUBGRAPH ISOMORPHISM has been done mostly when the size of the pattern graph is considered as a parameter. However, in this paper, we are going to study the general case where the pattern graph can be as large as the host graph.

We first observe that forbidding a graph as an induced substructure (an induced subgraph, an induced topological minor, or an induced minor) does not help for making SUBGRAPH ISOMORPHISM tractable unless we make the graph class trivial by forbidding either adjacent vertices or nonadjacent vertices. This can be done just by combining some easy observations and known results.

Observation 1.2 (★). *Let \mathcal{C} be the graph class that forbids a fixed graph H as either an induced subgraph, an induced topological minor, or an induced minor. Then, SUBGRAPH ISOMORPHISM on \mathcal{C} is polynomial-time solvable if H has at most two vertices; otherwise, it is NP-complete.*

Our main contribution in this paper is the following pair of results on SUBGRAPH ISOMORPHISM on graph classes forbidding a fixed graph as a substructure. (We prove Theorem 1.3 in Sect. 3 and Theorem 1.4 in Sect. 4.)

Theorem 1.3. *Let \mathcal{C} be the graph class that forbids a fixed connected graph $H \neq P_5$ as either a subgraph, a topological minor, or a minor. Then, SUBGRAPH ISOMORPHISM on \mathcal{C} is polynomial-time solvable if H is a subgraph of P_4 ; otherwise, it is NP-complete.*

Theorem 1.4. *Let \mathcal{C} be the graph class that forbids a fixed (not necessarily connected) graph H as either a subgraph, a topological minor, or a minor. Then, SUBGRAPH ISOMORPHISM on \mathcal{C} is*

- *fixed-parameter tractable parameterized by the order of H if H is a linear forest such that at most one component is of order 4 and all other components are of order at most 3;*

- NP-complete if either H is not a linear forest, H contains a component with six or more vertices, or H contains four components with five vertices.

Note that we have some missing cases. We do not know the complexity of the problem when the forbidden linear forest H contains either

- two or more disjoint P_4 subgraphs but no P_5 subgraph, or
- one, two, or three disjoint P_5 subgraphs but no P_6 subgraph.

2 Preliminaries and Basic Observations

We denote the path of n vertices by P_n , the complete graph of n vertices by K_n , and the star with ℓ leaves by $K_{1,\ell}$. For notational convenience, we allow ℓ to be 0; that is, $K_{1,0} = P_1 = K_1$. The disjoint union of graphs X and Y is denoted by $X \cup Y$ and the disjoint union of k copies of a graph Z is denoted by kZ .

A graph Q is a *minor* of G if Q can be obtained from G by removing vertices, removing edges, and contracting edges, where contracting an edge $\{u, v\}$ means adding a new vertex $w_{u,v}$, making the neighbors of u and v adjacent to $w_{u,v}$, and removing u and v . A graph Q is a *topological minor* of G if Q can be obtained by removing vertices, removing edges, and contracting edges, where contraction of an edge is allowed if one of the endpoints of the edge is of degree 2. A graph Q is a *subgraph* of G if Q can be obtained by removing vertices and edges. If we cannot remove edges but can do the other modifications as before, then we get the induced variants *induced minor*, *induced topological minor*, and *induced subgraph*.

Recall that a graph is a linear forest if it is the disjoint union of paths. In other words, a graph is a linear forest if and only if it does not contain a cycle nor a vertex of degree at least 3. Observe that in all graph containment relations mentioned above, if we do not forbid any linear forest from a graph class, then the class includes all linear forests. Thus, by Proposition 1.1, we have the following lemma.

Lemma 2.1. *If H is not a linear forest, then SUBGRAPH ISOMORPHISM is NP-complete for graphs that do not contain H as a minor, a topological minor, a subgraph, an induced minor, an induced topological minor, or an induced subgraph.*

2.1 Graphs Forbidding a Short Path as a Minor

By the discussion above, we can focus on a graph class forbidding a linear forest as a minor (or equivalently as a topological minor or a subgraph). We here characterize graph classes forbidding a short path as a minor.

Lemma 2.2 (★). *A connected P_3 -minor free graph is isomorphic to K_1 or K_2 .*

Lemma 2.3 (★). *A connected P_4 -minor free graph is isomorphic to either K_3 or $K_{1,s}$ for some $s \geq 0$.*

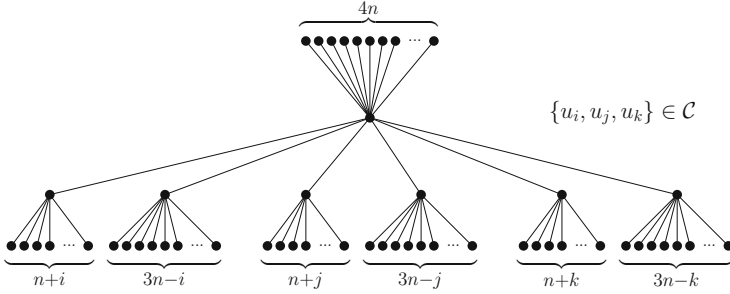


Fig. 1. The tree in G corresponding to $\{u_i, u_j, u_k\} \in \mathcal{C}$.

3 Forbidding a Connected Graph as a Minor

Here we first show that SUBGRAPH ISOMORPHISM on P_k -minor free graphs is linear-time solvable if $k \leq 4$. Note that P_k -minor free graphs include all $P_{k'}$ -minor free graphs if $k' \leq k$.

The following result can be easily obtained from Lemma 2.3.

Lemma 3.1 (★). SUBGRAPH ISOMORPHISM on P_4 -minor free graphs is linear-time solvable.

The following theorem implies that SUBGRAPH ISOMORPHISM on P_k -minor free graphs is NP-complete for every $k \geq 6$.

Theorem 3.2. SUBGRAPH ISOMORPHISM is NP-complete when the host graph is a forest without paths of length 6 and the pattern is a collection of stars.

Proof. The problem clearly is in NP. To show hardness, we reduce from EXACT 3-COVER [14]:

EXACT 3-COVER

Input: Collection \mathcal{C} of subsets of a set U such that each $c \in \mathcal{C}$ has size 3.

Question: Is there a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ such that $\bigcup_{C \in \mathcal{C}'} C = U$ and $|\mathcal{C}'| = |U|/3$?

Suppose we have an instance (\mathcal{C}, U) of EXACT 3-COVER given, where $U = \{u_0, \dots, u_{n-1}\}$. From (\mathcal{C}, U) , we construct the host graph G and the pattern Q .

The host G consists of the disjoint union of $|\mathcal{C}|$ trees as follows (see Fig. 1). For each set $C \in \mathcal{C}$, we take a tree in G as follows. Take a star $K_{1,4n+6}$. For each $u_i \in C$, do the following: take one of the leaves of the star, and add $n+i$ pendant vertices to it. Take another leaf of the star, and add $3n-i$ pendant vertices to it. I.e., if $C = \{u_i, u_j, u_k\}$, then the corresponding tree has seven vertices of degree more than 1: one vertex with degree $4n+6$, which is also adjacent to each of the other six non-leaf vertices; the non-leaf vertices have degree $n+i+1$, $3n-i+1$, $n+j+1$, $3n-j+1$, $n+k+1$, and $3n-k+1$. Call the vertex of degree $4n+6$ the *central* vertex of the component of C .

The pattern graph Q consists of a number of stars (see Fig. 2):

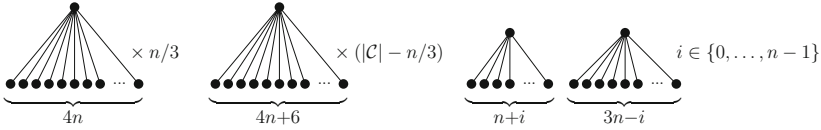


Fig. 2. The pattern graph Q .

- We have $n/3$ stars $K_{1,4n}$.
- We have $|\mathcal{C}| - n/3$ stars $K_{1,4n+6}$.
- For each $i \in \{0, \dots, n-1\}$, we have stars $K_{1,n+i}$ and $K_{1,3n-i}$. Call these the *element stars*.

From (\mathcal{C}, U) , G and Q can be constructed in polynomial time. Now we show that $Q \preceq G$ if and only if (\mathcal{C}, U) is a yes-instance of EXACT 3-COVER. We assume that $n > 6$ in the following.

The if direction: Suppose that the EXACT 3-COVER instance (\mathcal{C}, U) has a solution $\mathcal{C}' \subseteq \mathcal{C}$.

We map each $K_{1,4n+6}$ of Q into a component M of G corresponding to a set $D \notin \mathcal{C}'$. The center of $K_{1,4n+6}$ is mapped to the central vertex of M and all leaves to its neighbors. The other vertices T are isolated and not used.

Embed each $K_{1,4n}$ of Q into a component L of G corresponding to a set $C \in \mathcal{C}'$, mapping the center of $K_{1,4n}$ to the central vertex of L , and the leaves of $K_{1,4n}$ to leaves neighboring the central vertex of L . After we have done so, we left in this component six stars: if $C = \{u_i, u_j, u_k\}$, then the vertices in L that we did not yet use form stars $K_{1,n+i}$, $K_{1,3n-i}$, $K_{1,n+j}$, $K_{1,3n-j}$, $K_{1,n+k}$, $K_{1,3n-k}$. We thus can embed the element stars corresponding to u_i , u_j , and u_k in these stars, and have embedded the entire pattern in the host graph since \mathcal{C}' is a cover of U .

The only if direction: Suppose that $Q \preceq G$. Note that both Q and G have exactly $|\mathcal{C}|$ vertices of degree at least $4n$. Thus it follows that each vertex of degree at least $4n$ in Q must be mapped to a central vertex of a component in G . We can see that one of the following two cases must hold for the components in the host graph G .

Case 1: A star $K_{1,4n+6}$ is embedded in the component. This “uses up” the central vertex and all its neighbors. The only vertices in the component that are not in the image of the star $K_{1,4n+6}$ are leaves with its neighbor being used: these isolated vertices thus cannot be used for embedding any other stars. So all element stars must be embedded in components for which Case 2 holds.

Case 2: A star $K_{1,4n}$ is embedded in the component. At this point, note that the total number of vertices of element stars in Q equals $4n^2 + 2n$: each of the n elements has in total $4n$ leaves and two high degree vertices in its element stars. Also, the total number of vertices not used by the stars $K_{1,4n}$ in the Case 2-components equals $4n^2 + 2n$: we have $n/3$ components of Case 2 in G and each has $16n + 7$ vertices of which $4n + 1$ are used for embedding

the star $K_{1,4n}$. Thus, each vertex in a Case 2-component M must be used for embedding a vertex. This is only possible if we embed in M the element stars of the elements in the set corresponding to M .

So, let \mathcal{C}' be the sets whose component is of Case 2, i.e., where we embedded a $K_{1,4n}$ in its component. This subcollection \mathcal{C}' is a solution for EXACT 3-COVER: for each element u_i , its element stars are embedded in a component that corresponds to a set C that contains u_i , and by the argument above $C \in \mathcal{C}'$. \square

By Lemma 2.1, if a connected graph H is not a path, then SUBGRAPH ISOMORPHISM on H -minor free graphs is NP-complete. Assume that H is a path P_k . If $k \geq 6$, then by Theorem 3.2 the problem is NP-complete. If $k \leq 4$, then by Lemma 3.1 the problem can be solved in polynomial time. This completes the proof of Theorem 1.3.

4 Forbidding a Disconnected Graph as a Minor

In this section, we study the more general cases where the forbidden minor H is not necessarily connected. By Lemma 2.1, we can focus on linear forests H . We already know, by Theorem 3.2, if H contains a component with six or more vertices the problem becomes NP-complete. Thus in the following we consider the case where the components of H have five or less vertices.

Using the results in this section, we can prove Theorem 1.4. Corollary 4.2 implies the positive case of Theorem 1.4. Theorems 3.2 and 4.4 together with Lemma 2.1 imply the negative cases.

4.1 Subgraph Isomorphism on $(P_4 \cup kP_3)$ -Minor Free Graphs

We show that SUBGRAPH ISOMORPHISM on $(P_4 \cup kP_3)$ -minor free graphs is fixed-parameter tractable when parameterized by k . To this end, we present an algorithm that is parameterized by the vertex integrity, which we think is of independent interest. The *vertex integrity* [1] of a graph is the minimum integer k such that there is a vertex set $S \subseteq V$ such that $|S| \leq k$ and the maximum order of the components of $G - S$ is at most $k - |S|$. We call such S a *vi(k) set* of G . Note that the property of having vertex integrity at most k is closed under the subgraph relation.

This subsection is devoted to the proof of the following theorem.

Theorem 4.1. SUBGRAPH ISOMORPHISM on graphs of vertex integrity at most k is fixed-parameter tractable when parameterized by k .

By combining Theorem 4.1, Lemma 3.1, and the fact that kP_3 -minor free graphs have vertex integrity at most $3k - 1$, we can prove the following.

Corollary 4.2 (★). SUBGRAPH ISOMORPHISM on $(P_4 \cup kP_3)$ -minor free graphs is fixed-parameter tractable when parameterized by k .

To prove Theorem 4.1, we start with the following simple fact.

Lemma 4.3 (★). *Let η be a subgraph isomorphism from Q to G . For every $\text{vi}(k)$ set T of G , there exists a minimal $\text{vi}(k)$ set S of Q such that $\eta(S) \subseteq T$.*

Our algorithm assumes that there is a subgraph isomorphism η from Q to G and proceeds as follows:

1. find a $\text{vi}(k)$ set T of G ;
2. guess a minimal $\text{vi}(k)$ set S of Q such that $\eta(S) \subseteq T$;
3. guess the bijection between S and $R := \eta(S)$;
4. guess a subset $F \subseteq E(G - R)$ of the edges “unused” by η such that R is a $\text{vi}(k)$ set of $G - F$;
5. solve the problem of deciding the extendability of the guessed parts as the feasibility problem of an integer linear program with a bounded number of variables.

Proof (Proof of Theorem 4.1). Let G and Q be graphs of vertex integrity at most k . Our task is to find a subgraph isomorphism η from Q to G in FPT time parameterized by k .

We first find a $\text{vi}(k)$ set T of G and then guess a minimal $\text{vi}(k)$ set S of Q such that $\eta(S) \subseteq T$ for some subgraph isomorphism η from Q to G . By Lemma 4.3, such a set S exists if η exists. Finding T can be done in $O(k^{k+1}n)$ time [10], where $n = |V(G)|$. To guess S , it suffices to list all minimal $\text{vi}(k)$ set S of Q . The same algorithm in [10] can be used again: it lists all $O(k^k)$ candidates by branching on $k + 1$ vertices that induce a connected subgraph.

We then guess the subset R of T such that $\eta(S) = R$. We also guess for each $s \in S$, the image $\eta(s) \in R$. That is, we guess an injection from S to T . The number of such injections is $\binom{|T|}{|S|} \cdot |S|! \leq k!$. If there is an edge $\{u, v\} \in E(Q[S])$ such that $\{\eta(u), \eta(v)\} \notin E(G[R])$, then we reject this guess. Otherwise, we try to further extend η .

Observe that R is not necessarily a $\text{vi}(k)$ set of G . In the following, we guess “unnecessary” edges in $G - R$. That is, we guess a subset F of the edges that are not used by η as images of any edges in Q . Furthermore, we select F so that R is a $\text{vi}(k)$ set of $G - F$. Such F exists because η embeds $Q - S$ (and no other things) into $G - R$.

Guessing F : We now show that the number of candidates of F that we need to consider is bounded by some function in k . We partition F into three sets $F_1 = F \cap E(G[T - R])$, $F_2 = F \cap E(V(G) - T, T - R)$, and $F_3 = F \cap E(G - T)$ and then count the numbers of candidates separately.

Guessing F_1 : For F_1 , we just use all $2^{|E(G[T - R])|} < 2^{k^2}$ subsets of $E(G[T - R])$ as candidates. If R is not a $\text{vi}(k)$ set of $G[T] - F_1$, we reject this F_1 .

Guessing F_2 : Since we are finding F such that R is a $\text{vi}(k)$ set of $G - F$, each vertex in $T - R$ has less than k edges to $V(G) - T$ in $G - F$. Thus fewer than k^2 components of $V(G) - T$ have edges to $T - R$ in $G - F$. We guess such components \mathcal{C} .

Observe that each component in $V(G) - T$ is of order at most k and that each vertex of $V(G) - T$ can be partitioned into at most 2^k types with respect

to the adjacency to T . This implies that the components of $V(G) - T$ can be classified into at most 4^{k^2} types (2^{k^2} for the isomorphism type and $(2^k)^k$ for the adjacency to T) in such a way that if two components C_1 and C_2 of $G - T$ are of the same type, then there is an automorphism of G that fixes T and maps C_1 to C_2 . Given this classification of the components in $V(G) - T$, we only need to guess how many components of each type are included in \mathcal{C} . For this guess, we have at most $\binom{4^{k^2} + k^2 - 1}{k^2} < 4^{k^4 + k^2}$ options.

For each guess \mathcal{C} , we guess the edges connecting the components in \mathcal{C} to $T - R$ in $G - F$. Since $|\mathcal{C}| < k^2$ and $|C| \leq k$ for each $C \in \mathcal{C}$, there are at most $k^3 \cdot |T - R| \leq k^4$ candidate edges. We just try all $O(2^{k^4})$ subsets F'_2 of such edges, and set $F_2 = E(V(G) - T, T - R) - F'_2$. In total, we have $O(2^{k^4 + k^2} \cdot 2^{k^4})$ options for F_2 .

Guessing F_3 : Recall that $G - T$ does not contain any component of order more than k . Hence, if $G - R - (F_1 \cup F_2)$ has a component of order more than k , then it consists of some vertices in $T - R$ and some components in \mathcal{C} . Thus, we only need to pick some edges of the components in \mathcal{C} for F_3 to make R a $vi(k)$ set of $G - F$. We use all 2^{k^4} subsets of the edges of the components in \mathcal{C} as a candidate of F_3 .

In total, $F = F_1 \cup F_2 \cup F_3$ has at most $2^{k^2} \cdot 4^{k^4 + k^2} \cdot 2^{k^4} \cdot 2^{k^4}$ candidates, and each candidate can be found in FPT time. We reject this guess F if R is not a $vi(k)$ set of $G - F$. In the following, we assume that F is guessed correctly and denote $G - F$ by G' .

Extending η : Recall that we already know how η maps S to R and that each component in $Q - S$ and $G' - R$ is of order at most k . We now extend η by determining how η maps $Q - S$ to $G' - R$. By renaming vertices, we can assume that $S = \{s_1, \dots, s_q\}$, $R = \{r_1, \dots, r_q\}$, and $\eta(s_i) = r_i$ for $1 \leq i \leq q$.

We say that a vertex u in $Q - S$ matches a vertex v in $G' - R$ if $\{i \mid s_i \in N_Q(u) \cap S\} \subseteq \{i \mid r_i \in N_{G'}(v) \cap R\}$. A set of components $\{C_1, \dots, C_h\}$ of $Q - S$ fits a component D of $G' - R$ if there is an isomorphism ϕ from the disjoint union of C_1, \dots, C_h to D such that for all $u \in \bigcup_i V(C_i)$ and $v \in V(D)$, $\phi(u) = v$ holds only if u matches v . Note that if $h > k$, then $\{C_1, \dots, C_h\}$ can fit no component of $G' - R$.

As we did before for guessing F_2 , we classify the components of $Q - S$ and $G' - R$ into at most 4^{k^2} types. Two components C_1 and C_2 of $Q - S$ (or of $G' - R$) are of the same type if and only if there is an isomorphism ϕ from C_1 to C_2 such that $\phi(v_1) = v_2$ implies that $N_Q(v_1) \cap S = N_Q(v_2) \cap S$ (or $N_{G'}(v_1) \cap R = N_{G'}(v_2) \cap R$, respectively). We denote by $t(C)$ the type of a component C and by $t(\{C_1, \dots, C_h\})$ the multi-set $\{t(C_1), \dots, t(C_h)\}$. Observe that $\{C_1, \dots, C_h\}$ fits D if and only if all sets $\{C'_1, \dots, C'_h\}$ with $t(\{C'_1, \dots, C'_h\}) = t(\{C_1, \dots, C_h\})$ fits D' with $t(D') = t(D)$.

Observe that the guessed part $\eta|_S$ can be extended to a subgraph isomorphism η from Q to G' if and only if there is a partition of the components of $Q - S$ such that each part $\{C_1, \dots, C_h\}$ in the partition can be injectively mapped to a component D of $G' - R$ where $\{C_1, \dots, C_h\}$ fits D . To check the existence of

such a partition, we only need to find for each pair of a multi-set \mathcal{T} of types of a set of components in $Q - S$ and a type τ of a component in $G' - R$, how many sets of components of type \mathcal{T} the map η embeds to components of type τ . We use the following ILP formulation to solve this problem.

Let n_τ and n'_τ be the numbers of type- τ components in $Q - S$ and $G' - R$, respectively. These numbers can be computed in FPT time parameterized by k .

For each type τ and for each multi-set \mathcal{T} of types such that \mathcal{T} fits τ , we use a variable $x_{\mathcal{T},\tau}$ to represent the number of type- \mathcal{T} multi-sets of components in $Q - S$ that are mapped to type- τ components in $G' - R$. For each type τ of components in $G' - R$, we can embed at most n_τ sets of components in $Q - S$. This constraint is expressed as follows:

$$n_\tau \geq \sum_{\mathcal{T}: \mathcal{T} \text{ fits } \tau} x_{\mathcal{T},\tau} \quad \text{for each type } \tau. \tag{1}$$

For each type σ of components in $Q - S$, we need to embed all n_σ components of type σ into some components of $G - R'$. We can express this constraint as follows:

$$n_\sigma = \sum_{\mathcal{T}, \tau: \sigma \in \mathcal{T} \text{ and } \mathcal{T} \text{ fits } \tau} \mu_{\mathcal{T},\sigma} \cdot x_{\mathcal{T},\tau} \quad \text{for each type } \sigma, \tag{2}$$

where $\mu_{\mathcal{T},\sigma}$ is the multiplicity of σ in \mathcal{T} . This completes the ILP formulation of the problem. We do not have any objective function and just ask for the feasibility. The construction can be done in FPT time parameterized by k .

Observe that there are at most $\binom{4^{k^2}+k-1}{k} < 4^{k^3+k}$ multi-sets \mathcal{T} of types of components. Thus the ILP above has at most $4^{k^2} \cdot 4^{k^3+k}$ variables (the first factor for τ and the second for \mathcal{T}) and at most $4^{k^2} \cdot 4^{k^3+k} + 4^{k^2} \cdot 4^{k^2} \cdot 4^{k^3+k}$ constraints (the first term for (1) and the second for (2)) of length $O(4^{k^2} \cdot 4^{k^3+k})$. The coefficients are upper bounded by $|V(G')|$. It is known that the feasibility check of such an ILP can be done in FPT time parameterized by k [12, 18, 23]. Thus, the problem can be solved in FPT time when parameterized by k . \square

4.2 Subgraph Isomorphism on $4P_5$ -Minor Free Graphs

For this case, we show the NP-hardness by a reduction from (3, B2)-SAT [2], which is a restricted version of 3-SAT. (The proof is omitted in this version.)

Theorem 4.4 (★). SUBGRAPH ISOMORPHISM on $4P_5$ -minor free graphs is NP-complete.

5 Concluding Remarks

As we mentioned before, there are some unsettled cases for SUBGRAPH ISOMORPHISM on H -minor free graphs. If H is connected, then $H = P_5$ is only the unknown case. When H can be disconnected, we do not know the complexity

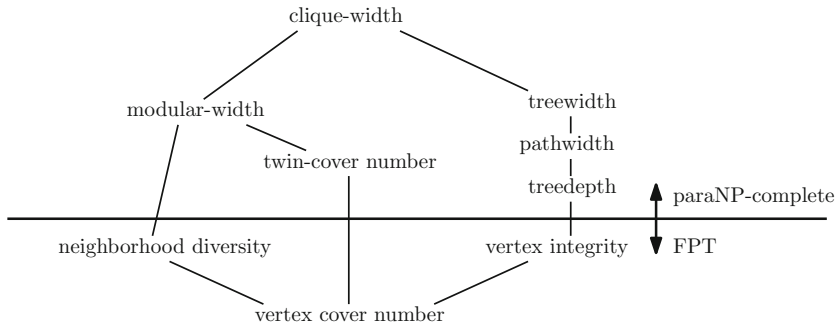


Fig. 3. Graph parameters and SUBGRAPH ISOMORPHISM. For each connection of parameters, there is a function in the parameter above that lower bounds the one below.

when H is a linear forest and either H contains kP_4 as a subgraph for $k \geq 2$ but no P_5 ; or H contains kP_5 as a subgraph for $k \in \{1, 2, 3\}$ but no P_6 .

Our results imply some parameterized results. See Fig. 3. (We omit the definitions of the parameters.) The proof of Theorem 3.2 implies that SUBGRAPH ISOMORPHISM is NP-complete even for graphs of tree-depth [28] at most 3. This bound is tight by Lemma 3.1 since graphs of tree-depth at most 2 does not contain P_4 as a subgraph. Proposition 1.1 implies it is NP-complete even for graphs of constant twin-cover number [13] because cluster graphs have twin-cover number 0. For the parameterization by neighborhood diversity [22], we can use techniques similar to the ones we used in this paper.

Theorem 5.1 (★). SUBGRAPH ISOMORPHISM on graphs of neighborhood diversity at most k is fixed-parameter tractable parameterized by k .

References

1. Barefoot, C.A., Entringer, R.C., Swart, H.C.: Vulnerability in graphs—a comparative survey. *J. Comb. Math. Comb. Comput.* **1**, 13–22 (1987)
2. Berman, P., Karpinski, M., Scott, A.D.: Approximation hardness of short symmetric instances of MAX-3SAT. Technical report TR03-049, Electronic Colloquium on Computational Complexity (ECCC) (2003)
3. Bodlaender, H.L., Nederlof, J., van der Zanden, T.C.: Subexponential time algorithms for embedding H -minor free graphs. In: ICALP 2016. LIPIcs, vol. 55, pp. 9:1–9:14 (2016)
4. Bonsma, P.: Surface split decompositions and subgraph isomorphism in graphs on surfaces. In: STACS 2012. LIPIcs, vol. 14, pp. 531–542 (2012)
5. Cygan, M., et al.: Tight lower bounds on graph embedding problems. *J. ACM* **64**(3), 18:1–18:22 (2017)
6. Cygan, M., et al.: *Parameterized Algorithms*. Springer, Cham (2015). <https://doi.org/10.1007/978-3-319-21275-3>
7. Damaschke, P.: Induced subgraph isomorphism for cographs is NP-complete. In: Möhring, R.H. (ed.) WG 1990. LNCS, vol. 484, pp. 72–78. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-53832-1_32

8. Dorn, F.: Planar subgraph isomorphism revisited. In: STACS 2010. LIPIcs, vol. 5, pp. 263–274 (2010)
9. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness II: on completeness for $W[1]$. *Theor. Comput. Sci.* **141**(1&2), 109–131 (1995)
10. Drange, P.G., Dregi, M.S., van 't Hof, P.: On the computational complexity of vertex integrity and component order connectivity. *Algorithmica* **76**(4), 1181–1202 (2016)
11. Eppstein, D.: Subgraph isomorphism in planar graphs and related problems. *J. Graph Algorithms Appl.* **3**(3), 1–27 (1999)
12. Frank, A., Tardos, É.: An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica* **7**(1), 49–65 (1987)
13. Ganian, R.: Improving vertex cover as a graph parameter. *Discret. Math. Theor. Comput. Sci.* **17**(2), 77–100 (2015)
14. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York (1979)
15. Gupta, A., Nishimura, N.: The complexity of subgraph isomorphism for classes of partial k -trees. *Theor. Comput. Sci.* **164**(1&2), 287–298 (1996)
16. Hajiaghayi, M., Nishimura, N.: Subgraph isomorphism, log-bounded fragmentation, and graphs of (locally) bounded treewidth. *J. Comput. Syst. Sci.* **73**(5), 755–768 (2007)
17. Jansen, B.M.P., Marx, D.: Characterizing the easy-to-find subgraphs from the viewpoint of polynomial-time algorithms, kernels, and Turing kernels. In: *SODA 2015*, pp. 616–629 (2015)
18. Kannan, R.: Minkowski's convex body theorem and integer programming. *Math. Oper. Res.* **12**(3), 415–440 (1987)
19. Kijima, S., Otachi, Y., Saitoh, T., Uno, T.: Subgraph isomorphism in graph classes. *Discrete Math.* **312**(21), 3164–3173 (2012)
20. Kiyomi, M., Otachi, Y.: Finding a chain graph in a bipartite permutation graph. *Inf. Process. Lett.* **116**(9), 569–573 (2016)
21. Konagaya, M., Otachi, Y., Uehara, R.: Polynomial-time algorithms for subgraph isomorphism in small graph classes of perfect graphs. *Discrete Appl. Math.* **199**, 37–45 (2016)
22. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. *Algorithmica* **64**(1), 19–37 (2012)
23. Lenstra Jr., H.W.: Integer programming with a fixed number of variables. *Math. Oper. Res.* **8**(4), 538–548 (1983)
24. Lingas, A.: Subgraph isomorphism for biconnected outerplanar graphs in cubic time. *Theor. Comput. Sci.* **63**(3), 295–302 (1989)
25. Marx, D., Pilipczuk, M.: Everything you always wanted to know about the parameterized complexity of subgraph isomorphism (but were afraid to ask). In: *STACS 2014. LIPIcs*, vol. 25, pp. 542–553 (2014)
26. Matoušek, J., Thomas, R.: On the complexity of finding iso- and other morphisms for partial k -trees. *Discrete Math.* **108**(1–3), 343–364 (1992)
27. Matula, D.W.: Subtree isomorphism in $O(n^{5/2})$. In: Alspach, B., Hell, P., Miller, D. (eds.) *Algorithmic Aspects of Combinatorics. Annals of Discrete Mathematics*, vol. 2, pp. 91–106. Elsevier (1978)
28. Nešetřil, J., de Mendez, P.O.: *Sparsity - Graphs, Structures, and Algorithms. Algorithms and Combinatorics*, vol. 28. Springer, Heidelberg (2012). <https://doi.org/10.1007/978-3-642-27875-4>