Information Processing Letters 116 (2016) 569–573

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Finding a chain graph in a bipartite permutation graph ☆

Masashi Kiyomi [a], Yota Otachi [b,*]

[a] International College of Arts and Sciences, Yokohama City University, 22-2 Seto, Kanazawa-ku, Yokohama, Kanagawa 236-0027, Japan
[b] School of Information Science, Japan Advanced Institute of Science and Technology, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan

ARTICLE INFO

Article history:
Received 29 May 2015
Received in revised form 7 April 2016
Accepted 8 April 2016
Available online 12 April 2016
Communicated by R. Uehara

Keywords:
Graph algorithms
Subgraph isomorphism
Chain graph
Bipartite permutation graph

ABSTRACT

We present a polynomial-time algorithm for solving Subgraph Isomorphism where the base graphs are bipartite permutation graphs and the pattern graphs are chain graphs.

1. Introduction

Given a *base* graph G and a *pattern* graph H, Subgraph Isomorphism (SGI for short) asks whether G contains a subgraph isomorphic to H, where a subgraph is a graph obtained by removing some edges and vertices. The problem SGI is in NP and generalizes many NP-complete problems such as Hamiltonicity, Clique, and Bandwidth. Thus SGI is NP-complete in general [2]. The complexity of SGI is studied in many aspects including the parameterized complexity and graph classes. In this paper, we study SGI by restricting input graphs to be in some graph classes. For studies in the parameterized complexity of SGI, see the recent papers by Marx and Pilipczuk [7], Jansen and Marx [4], and the references therein.

Since the problem SGI immediately becomes NP-complete if we allow the input graph class to contain all unions of disjoint paths or all unions of disjoint cliques [1], an easy NP-hardness reduction works for most of graph classes such as forests and cographs. Kijima et al. [5] thus studied a restricted version of SGI that they call Spanning Subgraph Isomorphism (SSGI, for short), where the base and pattern graphs are connected and have the same number of vertices. They showed that SSGI is NP-complete even for bipartite permutation graphs, proper interval graphs, and trivially perfect graphs. On the other hand, they also showed that SGI, the problem without the restrictions, is polynomial-time solvable for chain graphs, cochain graphs, and threshold graphs.

Recently, Konagaya et al. [6] have narrowed the complexity gap by showing that SGI is polynomial-time solvable if the base graphs are proper interval graphs (or the even larger class of chordal graphs) and the pattern graphs are cochain graphs, or if the base graphs are trivially perfect graphs and the pattern graphs are threshold graphs. The complexity of the case where the base graphs are bipartite permutation graphs and the pattern graphs are chain graphs remained unsettled.

In this paper, we study the unsettled case and show that it is polynomial-time solvable. That is, we show that SGI is polynomial-time solvable if the base graphs are bipartite permutation graphs and the pattern graphs are chain graphs.

☆ Partially supported by JSPS/MEXT KAKENHI Grant Numbers 25730003 and 24106004.

* Corresponding author.
E-mail addresses: masashi@yokohama-cu.ac.jp (M. Kiyomi), otachi@jaist.ac.jp (Y. Otachi).

http://dx.doi.org/10.1016/j.ipl.2016.04.006
0020-0190/

2. Preliminaries

Let $G = (V_G, E_G)$ and $H = (V_H, E_H)$ be graphs. We say that H is *subgraph-isomorphic* to G if there exists an injective adjacency-preserving map η from V_H to V_G; that is, $\eta(u) \neq \eta(v)$ if $u \neq v$ and $\{\eta(u), \eta(v)\} \in E_G$ holds for each $\{u, v\} \in E_H$. We call such a map η a *subgraph-isomorphism* from H to G. We call G and H the *base graph* and the *pattern graph*, respectively. Now SGI can be formally stated as follows:

Problem. SGI
Instance: A pair of graphs G and H.
Question: Is H subgraph-isomorphic to G?

Let $G = (V, E)$ be a graph. For $S \subseteq V$, we denote by $G[S]$ the subgraph of G induced by S. We denote the *neighborhood* of $v \in V$ in G by $N_G(v)$. A graph $G = (V, E)$ is a *bipartite graph* if the vertex set V can be partitioned into two sets X and Y of pairwise nonadjacent vertices. We denote such a graph by $G = (X, Y; E)$ to emphasize that it is bipartite. For a map f defined on A and a subset $B \subseteq A$ of the domain, we denote by $f(B)$ the set $\{f(b) : b \in B\}$.

2.1. Bipartite permutation graphs, chain graphs, and 2-layer chain graphs

A graph $G = (V, E)$ with $V = \{1, 2, \dots, n\}$ is a *permutation graph* if there is a permutation π over V such that $\{i, j\} \in E$ if and only if $(i - j)(\pi(i) - \pi(j)) < 0$. A *bipartite permutation graph* is a permutation graph that is bipartite.

A bipartite graph $H = (U, V; E_H)$ is a *chain graph* if the vertices can be ordered as $U = \{u_1, u_2, \dots, u_{|U|}\}$ and $V = \{v_1, v_2, \dots, v_{|V|}\}$ such that $N(u_1) \subseteq N(u_2) \subseteq \cdots \subseteq N(u_{|U|})$, and $N(v_1) \subseteq N(v_2) \subseteq \cdots \subseteq N(v_{|V|})$. It is known that such orderings can be computed in linear time [3].

A bipartite graph $G = (V_G, E_G)$ is a *2-layer chain graph* if the vertex set V_G can be partitioned into independent sets X, Y, Z so that there is no edge between X and Z, and the vertices can be ordered as $X = \{x_1, x_2, \dots, x_{|X|}\}$, $Y = \{y_1, y_2, \dots, y_{|Y|}\}$, and $Z = \{z_1, z_2, \dots, z_{|Z|}\}$ such that

- $N_G(x_1) \subseteq N_G(x_2) \subseteq \cdots \subseteq N_G(x_{|X|})$,
- $N_G(z_1) \subseteq N_G(z_3) \subseteq \cdots \subseteq N_G(z_{|Z|})$,
- $N_G^{(X)}(y_1) \supseteq N_G^{(X)}(y_2) \supseteq \cdots \supseteq N_G^{(X)}(y_{|Y|})$, and
- $N_G^{(Z)}(y_1) \subseteq N_G^{(Z)}(y_2) \subseteq \cdots \subseteq N_G^{(Z)}(y_{|Y|})$,

where $N_G^{(X)}(y_i) = N_G(y_i) \cap X$ and $N_G^{(Z)}(y_i) = N_G(y_i) \cap Z$. We denote such a 2-layer graph by $G = (X, Y, Z; E_G)$. See Fig. 1.

Note that a partition and orderings in the definition of 2-layer chain graphs above can be computed in polynomial time as follows. Given a 2-layer chain graph $B = (P, Q; E)$, we first guess which of P and Q is Y. Assume that we have guessed that $Q = Y$, and thus $P = X \cup Z$. Now for any two vertices p and p' in P check whether $N_B(p) \subseteq N_B(p')$ or $N_B(p) \supseteq N_B(p')$ holds. If not, then neither $\{p, p'\} \subseteq X$ nor $\{p, p'\} \subseteq Z$ can happen. Let R be the set of such pairs. Using a linear-time algorithm for 2-coloring, we can partition P into two sets X and Z in such a way that no pair in

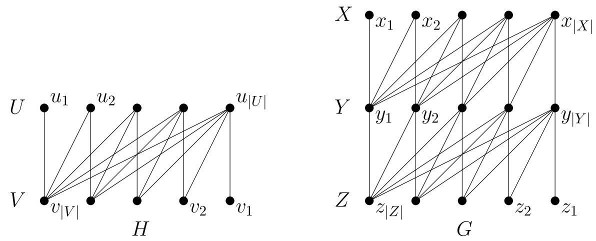

Fig. 1. A chain graph $H = (U, V; E_H)$ and a 2-layer chain graph $G = (X, Y, Z; E_G)$.

R is entirely included in X or in Z. Although this partition may not be unique, we can pick one arbitrarily. Next we can compute orderings on X and Z with respect to the inclusion ordering of neighborhoods. The ordering on Y is a linear extension of the intersection of two inclusion orderings $\supseteq$ and $\subseteq$ defined on $N_G^{(X)}(\cdot)$ and $N_G^{(Z)}(\cdot)$, respectively. This can be computed also in polynomial time.

Since they can be computed in polynomial time, we assume in the following that chain graphs and 2-layer chain graphs are given with vertex partitions and vertex orderings defined above.

Using a characterization given by Sprague [8], Konagaya et al. [6] showed a reduction from the unsettled case to a simpler case.

Theorem 1 *([6]). If* SGI *is polynomial-time solvable when the base graphs are connected 2-layer chain graphs and the pattern graphs are connected chain graphs, then* SGI *is also polynomial-time solvable when the base graphs are bipartite permutation graphs and the pattern graphs are chain graphs.*

By the theorem above, we can focus on the case where the base graphs are connected 2-layer chain graphs. For this case, we will present a polynomial-time algorithm for solving SGI.

3. Finding a connected chain graph in a connected 2-layer chain graph

Let $G = (X, Y, Z; E_G)$ be a connected 2-layer chain graph with $X = \{x_1, \dots, x_{|X|}\}$, $Y = \{y_1, \dots, y_{|Y|}\}$, and $Z = \{z_1, \dots, z_{|Z|}\}$. Let $H = (U, V; E_H)$ be a connected chain graph with $U = \{u_1, \dots, u_{|U|}\}$ and $V = \{v_1, \dots, v_{|V|}\}$.

Note that since both G and H are connected bipartite graphs, a subgraph-isomorphism η from H to G satisfies either $\eta(U) \subseteq Y$ or $\eta(U) \subseteq X \cup Z$. Also, $\eta(U) \subseteq Y$ implies $\eta(V) \subseteq X \cup Z$. In the rest of this section, we assume that we correctly guessed that $\eta(U) \subseteq Y$ and $\eta(V) \subseteq X \cup Z$ since we can perform the algorithm twice.

3.1. Guessing the used vertices in the base graph G

In this subsection, we show that the vertices of G used by a subgraph-isomorphism from H to G can be guessed from polynomially many candidates. First we show that vertices in Y can be chosen consecutively.

Lemma 2. *If there is a subgraph-isomorphism η from H to G with $\eta(U) \subseteq Y$, then there is a subgraph-isomorphism η' from H to G such that $\eta'(U) = \{y_s, y_{s+1}, \dots, y_{s+|U|-1}\}$ for some s.*