Theoretical Computer Science ••• (••••) •••–•••

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Vertex deletion problems on chordal graphs ☆

Yixin Cao [a], Yuping Ke [a], Yota Otachi [b], Jie You [c,a,*]

[a] *Department of Computing, Hong Kong Polytechnic University, Hong Kong, China*
[b] *Faculty of Advanced Science and Technology, Kumamoto University, Kumamoto, Japan*
[c] *School of Information Science and Engineering, Central South University, Changsha, China*

ARTICLE INFO

Article history:
Received 7 October 2017
Received in revised form 29 May 2018
Accepted 30 May 2018
Available online xxxx
Communicated by R. Klasing

Keywords:
Vertex deletion problem
Maximum (induced) subgraph
Chordal graph
(Unit) interval graph
Split graph
Hereditary property

ABSTRACT

Containing many classic optimization problems, the family of vertex deletion problems has an important position in algorithm and complexity study. The celebrated result of Lewis and Yannakakis gives a complete dichotomy of their complexity. It however has nothing to say about the case when the input graph is also special. This paper initiates a systematic study of vertex deletion problems from one subclass of chordal graphs to another. We give polynomial-time algorithms or proofs of NP-completeness for most of the problems. In particular, we show that the vertex deletion problem from chordal graphs to interval graphs is NP-complete.

1. Introduction

Generally speaking, a vertex deletion problem asks to transform an input graph to a graph in a certain class by deleting a minimum number of vertices. Many classic optimization problems belong to the family of vertex deletion problems, and their algorithms and complexity have been intensively studied. For example, the clique problem and the independent set problem are nothing but the vertex deletion problems to complete graphs and to edgeless graphs respectively. Most interesting graph properties are *hereditary*: If a graph satisfies this property, then so does every induced subgraph of it. For all the vertex deletion problems to hereditary graph classes, Lewis and Yannakakis [27] have settled their complexity once and for all with a dichotomy result: They are either NP-hard or trivial. Thereafter algorithmic efforts were mostly focused on the nontrivial ones, and the major approaches include approximation algorithms [28], parameterized algorithms [6], and exact algorithms [15].

Chordal graphs make one of the most important graph classes. Together with many of its subclasses, it has played important roles in the development of structural graph theory. (We defer their definitions to the next section.) Many algorithms have been developed for vertex deletion problems to chordal graphs and its subclasses,—most notably (unit) interval graphs, cluster graphs, and split graphs; see, e.g., [17,4,10,9,8,34,12,25,1] for a partial list. After the long progress of algorithmic

☆ An extended abstract of this paper has appeared in the proceedings of FSTTCS 2017. Supported in part by the Hong Kong Research Grants Council (RGC) under grants 252026 and 152261, and the National Natural Science Foundation of China (NSFC) under grants 61572414 and 61420106009.

* Corresponding author at: School of Information Science and Engineering, Central South University, Changsha, China.
E-mail addresses: yixin.cao@polyu.edu.hk (Y. Cao), yu.ke@polyu.edu.hk (Y. Ke), otachi@cs.kumamoto-u.ac.jp (Y. Otachi), youjie@csu.edu.cn (J. You).

https://doi.org/10.1016/j.tcs.2018.05.039
0304-3975/

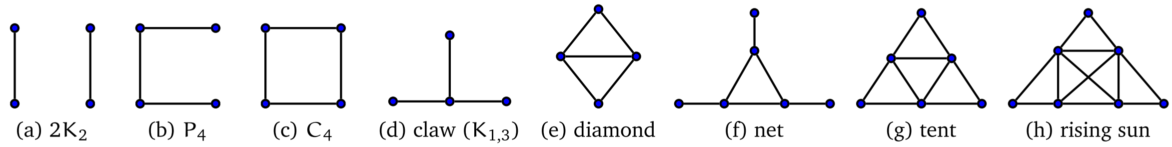

Fig. 1. Small subgraphs.

achievements, some natural questions arise: What is the complexity of transforming a chordal graph to a (unit) interval graph, a cluster graph, a split graph, or a member of some other subclass of chordal graphs? It is quite surprising that this type of problems has not been systematically studied, save few concrete results, e.g., the polynomial-time algorithms for the clique problem, the independent set problem, and the feedback vertex set problem (the object class being forests) [21,33].

The same question can be asked for other pair of source and object graph classes. The most important source classes include planar graphs [20,18,16], bipartite graphs [32], and degree-bounded graphs [19]. As one may expect, with special properties imposed on input graphs, the problems become easier, and some of them may not remain NP-hard. Unfortunately, a clear-cut answer to them seems very unlikely, since their complexity would depend upon both the source class and the object class. Indeed, some are trivial (e.g., vertex cover on split graphs), some remain NP-hard (e.g., vertex cover on planar graphs), while some others are in P but can only be solved by very nontrivial polynomial-time algorithms (e.g., vertex cover on bipartite graphs).

Throughout the paper we write the names of graph classes in small capitals; e.g., CHORDAL and BIPARTITE stand for the class of chordal graphs and the class of bipartite graphs respectively. We use $\mathcal{C}$, commonly with subscripts, to denote an unspecified hereditary graph class, and use $\mathcal{C}_1 \to \mathcal{C}_2$ to denote the vertex deletion problem from class $\mathcal{C}_1$ to class $\mathcal{C}_2$:

Given a graph G in $\mathcal{C}_1$, one is asked for a minimum set $V_- \subseteq V(G)$ such that $G - V_-$ is in $\mathcal{C}_2$.

It is worth noting that $\mathcal{C}_2$ may or may not be a subclass of $\mathcal{C}_1$, and when it is not, the problem is equivalent to $\mathcal{C}_1 \to \mathcal{C}_1 \cap \mathcal{C}_2$: Since $\mathcal{C}_1$ is hereditary, $G - V_-$ is necessarily in $\mathcal{C}_1$. For almost all classes $\mathcal{C}$, the complexity of problems PLANAR $\to \mathcal{C}$ and BIPARTITE $\to \mathcal{C}$ has been answered in a systematical manner [27,32], while for most graph classes $\mathcal{C}$, the complexity of problem DEGREE-BOUNDED $\to \mathcal{C}$ has been satisfactorily determined [19].

Apart from CHORDAL, we will also consider vertex deletion problems on its subclasses. Therefore, our purpose in this paper is a focused study on the algorithms and complexity of $\mathcal{C}_1 \to \mathcal{C}_2$ with both $\mathcal{C}_1$ and $\mathcal{C}_2$ being subclasses of CHORDAL. Since it is generally acknowledged that the study of chordal graphs motivated the theory of perfect graphs [24,2], the importance of chordal graphs merits such a study from the aspect of structural graph theory. However, our main motivation is from the recent algorithmic progress in vertex deletion problems. It has come to our attention that to transform a graph to class $\mathcal{C}_1$, it is frequently convenient to first make it a member of another class $\mathcal{C}_2$ that contains $\mathcal{C}_1$ as a proper subclass, followed by an algorithm for the $\mathcal{C}_2 \to \mathcal{C}_1$ problem [30,9,7,34].

There being many subclasses of CHORDAL, the number of problems fitting in our scope is quite prohibitive. The following simple observations will save us a lot of efforts.

Proposition 1.1. *Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two graph classes.*

(1) If the $\mathcal{C}_1 \to \mathcal{C}_2$ problem can be solved in polynomial time, then so is $\mathcal{C} \to \mathcal{C}_2$ for any subclass $\mathcal{C}$ of $\mathcal{C}_1$.
(2) If the $\mathcal{C}_1 \to \mathcal{C}_2$ problem is NP-complete, then so is $\mathcal{C} \to \mathcal{C}_2$ for any superclass $\mathcal{C}$ of $\mathcal{C}_1$.

For example, the majority of our hardness results for problems CHORDAL $\to \mathcal{C}$ are obtained by proving the hardness of SPLIT $\to \mathcal{C}$. Indeed, this is very natural as in literature, most (NP-)hardness of problems on chordal graphs is proved on split graphs, e.g., dominating set [3], Hamiltonian path [29], and maximum cut [5]. The most famous exception is probably the pathwidth problem, which can be solved in polynomial time on split graphs but becomes NP-complete on chordal graphs [23]. No problem like this surfaces during our study, though we do have the following hardness result proved directly on chordal graphs, for which we have no conclusion on split graphs.

Theorem 1.2. *Let F be a biconnected chordal graph. If F is not complete, then the CHORDAL $\to$ F-FREE problem is NP-complete.*

Another simple observation of common use to us is about complement graph classes. The *complement* $\overline{G}$ of graph G is defined on the same vertex set $V(G)$, where a pair of distinct vertices u and v is adjacent in $\overline{G}$ if $uv \notin E(G)$. It is easy to see that the complement of $\overline{G}$ is G. In Fig. 1, for example, the net and the tent are the complements of each other. The *complement* of a graph class $\mathcal{C}$, denoted by $\overline{\mathcal{C}}$, comprises all graphs whose complements are in $\mathcal{C}$; e.g., the complement of COMPLETE SPLIT is $\{2K_2, P_3\}$-FREE. A graph class $\mathcal{C}$ is *self-complementary* if it is its own complement, i.e., a graph $G \in \mathcal{C}$ if and

Please cite this article in press as: Y. Cao et al., Vertex deletion problems on chordal graphs, Theoret. Comput. Sci. (2018), https://doi.org/10.1016/j.tcs.2018.05.039