Discrete Applied Mathematics 160 (2012) 1791–1797

Contents lists available at SciVerse ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Restricted vertex multicut on permutation graphs

Charis Papadopoulos *

Department of Mathematics, University of Ioannina, GR-45110, Ioannina, Greece

ARTICLE INFO

Article history:
Received 25 August 2011
Received in revised form 13 March 2012
Accepted 19 March 2012
Available online 12 April 2012

Keywords:
Multicut problem
Permutation graphs
Split graphs
Polynomial-time algorithm
NP-completeness

ABSTRACT

Given an undirected graph and pairs of terminals the RESTRICTED VERTEX MULTICUT problem asks for a minimum set of nonterminal vertices whose removal disconnects each pair of terminals. The problem is known to be NP-complete for trees and polynomial-time solvable for interval graphs. In this paper we give a polynomial-time algorithm for the problem on permutation graphs. Furthermore we show that the problem remains NP-complete on split graphs whereas it becomes polynomial-time solvable for the class of co-bipartite graphs.

1. Introduction

One of the well-studied problems that fall in the area of cut and separation problems is the MULTICUT problem introduced by Hu in [14]. Given a graph G and a list L of pairs of vertices that are called terminals, the objective for the MULTICUT problem is to disconnect each terminal pair of the predefined list by removing a minimum set of edges or vertices of G. Problems of this kind arise from areas concerning with the reliability and robustness of network communications [6]. The MULTICUT problem is NP-complete [7] and several algorithms that approximate a solution on general or restricted graph classes are known [5,9,16], while the parameterized version was proved to be fixed-parameter tractable only very recently [3,17]. This problem includes as a special case the MULTIWAY CUT problem where instead of the list L we are given a set of terminals that need to be pairwise separated from each other; see [7,11,15]. Depending on the multicut, that is, the set of vertices or edges whose deletion disconnects each terminal pair, there are three variations of the problem.

The history of multicut problems begins with the edge variation, known as the EDGE MULTICUT, that allows only the removal of edges. If L contains at most two terminal pairs the EDGE MULTICUT problem admits a polynomial-time algorithm [21] whereas for at least three terminal pairs it becomes NP-hard [7]. Furthermore EDGE MULTICUT remains NP-hard even when the input graph is a star (tree of height 1) [10] and therefore excluding any possible polynomial solution on many interesting graph classes. Similar to the EDGE MULTICUT is the VERTEX MULTICUT problem in which one is only allowed to remove vertices of the input graph.

As it was introduced by Călinescu et al. in [5] the VERTEX MULTICUT problem splits into two subproblems depending on whether one is allowed to remove terminal vertices. The UNRESTRICTED VERTEX MULTICUT problem refers to finding *any* minimum set of vertices of G whose removal separates each terminal pair of L whereas the RESTRICTED VERTEX MULTICUT refers to minimizing a set of *nonterminal* vertices for the same objective. UNRESTRICTED VERTEX MULTICUT is NP-hard on interval graphs [13], graphs with bounded treewidth [5], and planar graphs of bounded degree [5]. From the positive side there is a polynomial-time algorithm for the latter problem on trees [5]. Looking at the line graph of G (that is, the graph representing the adjacencies between the edges of G) an interesting reduction shows that the vertex variant is more general than the edge variant [5]. More precisely, consider an instance of EDGE MULTICUT whose input is a graph G and a set of

* Tel.: +30 26510 08224; fax: +30 26510 08201.
E-mail address: charis@cs.uoi.gr.

0166-218X/$ – see front matter
doi:10.1016/j.dam.2012.03.021

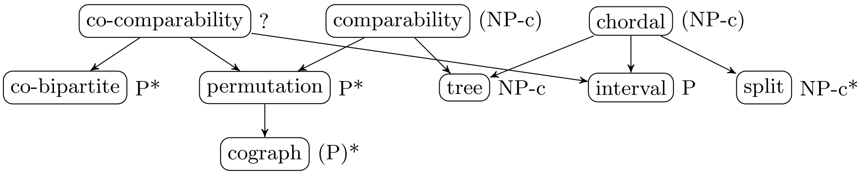

Fig. 1. An inclusion relationship of the considered graph classes and the complexity of Restricted Vertex Multicut problem in each graph class. The arrow → represents the ⊃ relation. NP-c means NP-complete, P means polynomial-time algorithm, the asterisk ∗ indicates that the result is obtained here, the pair of parenthesis () means that the complexity is obtained from graph inclusion relationships, and ? stands for unknown complexity.

terminal pairs L, denoted by (G, L). Construct the line graph of G, denoted by $L(G)$ and construct the set of terminal pairs L' which contains for each pair (s, t) of L all pairs (e_i, f_i) such that e_i has s as endpoint and f_i has t as endpoint. Then it is known that a solution for the Edge Multicut on (G, L) is a solution for the Unrestricted Vertex Multicut on $(L(G), L')$ and vice versa [5]. Observe that the line graph of a star graph is a complete graph and, therefore, Unrestricted Vertex Multicut on complete graphs is NP-complete. Thus for many interesting and even constraint graph classes (e.g., cographs, split graphs, and co-bipartite graphs) the Unrestricted Vertex Multicut problem is already NP-hard. Here we focus on the Restricted Vertex Multicut problem.

Although many optimization problems that are NP-hard on arbitrary graphs are polynomially solvable on restricted graph classes [4,12], not much seems to be known for the restricted variation of multicut on particular graph classes. The problem admits a polynomial-time algorithm for interval graphs [13], whereas it becomes NP-hard for trees [5,13] and, thus, for chordal graphs which form a proper superclass of interval graphs. Therefore it is interesting to study the complexity of the Restricted Vertex Multicut problem on graph classes that are characterized without induced trees and are not included in the class of interval graphs.

In this paper we consider the restricted version of the multicut problem on split graphs, permutation graphs, and other related graph classes. Split graphs form a proper subclass of chordal graphs and such graphs are unrelated to interval graphs [12]. We prove that the problem remains NP-hard on split graphs. A natural superclass of interval graphs is the class of co-comparability graphs (complements of comparability graphs) where the complexity of the Restricted Vertex Multicut problem is still unknown. Co-bipartite (complements of bipartite graphs) and permutation graphs are two unrelated subclasses of co-comparability graphs [4,12]. Interestingly most problems that are hard on co-comparability graphs are already hard on co-bipartite graphs. Here we show that the problem admits a simple and efficient (polynomial-time) solution on co-bipartite graphs and therefore excluding such an approach through a hardness result on co-bipartite graphs. Our main result is a polynomial-time algorithm for the class of permutation graphs. To do so, we take advantage of the notion of scanlines already efficiently applied for other problems such as treewidth and minimum fill-in on permutation graphs [1,19]. We also give an independent result for cographs that can been seen as a special case of the polynomial-time algorithm on permutation graphs. An overall picture of our results is depicted in Fig. 1.

2. Preliminaries

We consider undirected finite graphs with no loops or multiple edges. For a graph G, we denote its vertex and edge set by $V(G)$ and $E(G)$, respectively, with $n = |V(G)|$ and $m = |E(G)|$. For a vertex subset $S \subseteq V(G)$, the subgraph of G induced by S is denoted by $G[S]$. Moreover, we denote by $G - S$ the graph $G[V(G) \setminus S]$.

The *neighborhood* $N(x)$ of a vertex x of the graph G is the set of all the vertices of G which are adjacent to x. The *closed neighborhood* of x is defined as $N[x] = N(x) \cup \{x\}$. If $S \subseteq V(G)$, then the neighbors of S, denoted by $N(S)$, are given by $\bigcup_{x \in S} N(x) - S$. The *complement* $\overline{G}$ of a graph G has vertex set $V(G)$ and all edges not in G. A *clique* is a set of pairwise adjacent vertices while an *independent set* is a set of pairwise non-adjacent vertices. A graph is *connected* if there is a path between any pair of vertices. A *connected component* of G is a maximal connected subgraph of G. A chordless path on k vertices is denoted by P_k. A tree of height one is called a *star graph* and a star graph on four vertices is called *claw*.

For a set $\mathcal{F}$ of graphs, a graph is called *$\mathcal{F}$-free* if it does not contain a graph from $\mathcal{F}$ as induced subgraph. For all graph classes mentioned here proper definitions and characterizations can be found in [4,12], though we give the corresponding characterizations at the appropriate places. We only mention at the moment that for every graph class Π that we consider here Π is closed under vertex removals, that is, Π is *hereditary*. The relationships between the considered graph classes are shown in Fig. 1.

Let $G = (V, E)$ be a graph and let $L = \{(s_1, t_1), \ldots, (s_l, t_l)\}$ be a specified set of pairs of vertices, where the vertices of each pair are distinct, but vertices in different pairs are not required to be distinct. The set of vertices of L are called *terminals* denoted by T whereas the rest of the vertices are called *nonterminals*.

The Restricted Vertex Multicut problem can be formulated as follows.

Restricted Vertex Multicut
Input: An undirected graph $G = (V, E)$, a collection of pairs of vertices $L \subseteq V \times V$, and an integer $k \geq 0$.
Task: Find a subset S of V that contains only nonterminal vertices such that $|S| \leq k$ and vertices of each pair of L belong to different connected components of $G - S$.