Theoretical Computer Science 607 (2015) 60–67

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Obtaining split graphs by edge contraction

Chengwei Guo *, Leizhen Cai *,[1]

Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong S.A.R., China

ARTICLE INFO

Article history:
Received 1 October 2014
Accepted 24 January 2015
Available online 23 April 2015

Keywords:
Parameterized complexity
Edge contraction
Split graphs
FPT algorithm
Incompressibility

ABSTRACT

We study the parameterized complexity of the following SPLIT CONTRACTION problem: Given a graph G, and an integer k as parameter, determine whether G can be modified into a split graph by contracting at most k edges. We show that SPLIT CONTRACTION can be solved in FPT time $2^{O(k^2)}n^5$, but admits no polynomial kernel unless $NP \subseteq coNP/poly$.

1. Introduction

Graph modification problems constitute a fundamental and well-studied family of problems in algorithmic graph theory. Many classical graph problems, such as MAXIMUM CLIQUE, FEEDBACK VERTEX SET, ODD CYCLE TRANSVERSAL, and MINIMUM FILL-IN, can be formulated as graph modification problems. A graph modification problem takes a graph G and an integer k as inputs, and asks whether G can be modified into a graph belonging to a specified graph class, using at most k operations of a given type, such as vertex deletion, edge deletion, or edge addition. The number k of operations measures how close a graph is to such a specified class of graphs. Besides vertex/edge deletion/addition, modifying graphs by *edge contraction* has been studied in the literature, yielding several NP-completeness results for the following decision problem:

Π-CONTRACTION
Instance: Graph $G = (V, E)$, positive integer k.
Question: Can we modify G into a Π-graph (i.e. a graph belonging to class Π) by contracting at most k edges?

Very recently, research on Π-CONTRACTION problems for various graph class Π has been initiated from the parameterized point of view. Π-CONTRACTION has been proved to be FPT for Π being bipartite graphs (Heggernes et al. [15], Guillemot and Marx [12]), trees and paths (Heggernes et al. [14]), planar graphs (Golovach et al. [11]), graphs with minimum degree $\geq d$ (Golovach et al. [10]), graphs with maximum degree $\leq d$ (Belmonte et al. [1]), complete graphs (Cai and Guo [4], Lokshtanov et al. [17]), and cographs (Lokshtanov et al. [17]). On the other hand, Cai and Guo [4], and Lokshtanov et al. [17] independently showed that Π-CONTRACTION is W[2]-hard for Π being chordal graphs. Furthermore, Cai and Guo [4] also proved W[2]-hardness of Π-CONTRACTION for Π being H-free for any fixed 3-connected graph H.

* Corresponding authors.
E-mail addresses: cwguo9@gmail.com (C. Guo), lcai@cse.cuhk.edu.hk (L. Cai).
[1] Partially supported by GRF grant CUHK410409 of the Research Grants Council, University Grants Committee, Hong Kong.

http://dx.doi.org/10.1016/j.tcs.2015.01.056

In this paper, we study the parameterized complexity of Π-Contraction when Π is the class of split graphs, which forms a subclass of chordal graphs.

Split Contraction
Instance: Graph $G = (V, E)$, positive integer k.
Question: Can we obtain a split graph from G by contracting at most k edges?
Parameter: k.

We shall mention its edge deletion and vertex deletion variants, known as Split Deletion and Split Vertex Deletion, asking whether an input graph can be modified into a split graph by deleting at most k edges or deleting at most k vertices respectively. Both of these problems were shown to be FPT (Cai [3]) and have polynomial kernels (Guo [13]). Recently, faster FPT algorithms and improved kernels have been obtained (Ghosh et al. [9], Cygan and Pilipczuk [7]). As split graphs are characterized by forbidden induced subgraphs $\{2K_2, C_4, C_5\}$, any solution set of Split Deletion or Split Vertex Deletion must hit every induced copy of these forbidden subgraphs in the input graph, implying that choices for branching can be bounded. This observation directly or indirectly yields above FPT algorithms and kernelization reductions for Split Deletion and Split Vertex Deletion. Unfortunately, such observation is no longer true for Split Contraction, as contractions can occur for edges not involved in any induced copies of forbidden subgraphs, making Split Contraction much harder than its edge deletion and vertex deletion variants.

Although most techniques for Split Deletion and Split Vertex Deletion seem unavailable for Split Contraction due to above reason, there is a simple relationship between Split Vertex Deletion and Split Contraction: Every yes-instance (G, k) of Split Contraction implies a yes-instance $(G, 2k)$ of Split Vertex Deletion. Therefore, every FPT algorithm for Split Vertex Deletion can be used to obtain a small vertex set whose deletion results in a split graph, in a yes-instance of Split Contraction. This observation will be used as a starting point of our FPT algorithm for Split Contraction.

1.1. Our contributions

Our main result is an FPT algorithm for Split Contraction, which complements the FPT results on other graph modification problems related to split graphs: Split Deletion and Split Vertex Deletion.

Theorem 2.12. Split Contraction *can be solved in time* $2^{O(k^2)}n^5$.

Our algorithm starts by finding a large split subgraph H in the input graph (Phase 1), and then considers two cases depending on the clique size of H.

If the clique of H is large (Phase 2), then we show that almost all vertices in this clique are finally included in the clique of a target split graph. We use a branch-and-search algorithm to enumerate edge contractions in small sets, and reduce the original instance into several instances of Clique Contraction, which is known to be FPT.

If the clique of H is small (Phase 3), then there will be a large independent set in the input graph. We develop reduction rules based on a variant of "modular decomposition" of the input graph: Partition vertices into groups such that each group induces an independent set, and all vertices in each group have the same set of neighbors. We can bound the number of such groups (Reduction 1), delete "irrelevant" vertices in each group (Reduction 2), and reduce the input graph to an "equivalent" graph with bounded number of vertices. We note that these reduction rules are useful for establishing kernelization algorithms of other contraction problems such as Clique Contraction and Biclique Contraction (see Fig. 1).

On the other hand, we prove, by a polynomial parameter transformation from the Red–Blue Dominating Set problem, that Split Contraction is incompressible.

Theorem 3.1. Split Contraction *admits no polynomial kernel unless NP* $\subseteq$ *coNP/poly.*

This result is in contrast to that Split Deletion and Split Vertex Deletion have polynomial kernels.

1.2. Preliminaries

Graphs. Throughout the paper, we only consider simple and undirected graph $G = (V, E)$, where V is the vertex set and E is the edge set. Two vertices $u, v \in V$ are *adjacent* iff $uv \in E$. A vertex v is *incident* with an edge e iff v is an endpoint of e. The *neighbor set* $N_G(v)$ of a vertex $v \in V$ is the set of vertices that are adjacent to v in G. The *closed neighbor set* of v is denoted by $N_G[v] = N_G(v) \cup \{v\}$. For a set X of vertices or edges in G, we use $G - X$ to denote the graph obtained by deleting X from G. For a set of vertices $V' \subseteq V$, we write $G[V']$ to denote the subgraph of G induced by V' and write $E[V']$ to denote the set of edges in G whose both endpoints are in V'.

A graph G is a *split graph* if its vertex set can be partitioned into a clique K and an independent set I, where $(K; I)$ is called a *split partition* of G. The class of split graphs is hereditary, and is characterized by a set $\{2K_2, C_4, C_5\}$ of forbidden induced subgraphs.