Journal of Discrete Algorithms ••• (••••) •••–•••

Contents lists available at ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

On the complexity of multi-parameterized cluster editing ☆

Faisal N. Abu-Khzam

Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

ARTICLE INFO

Article history:
Received 30 November 2015
Received in revised form 15 May 2017
Accepted 11 July 2017
Available online xxxx

Keywords:
Cluster editing
Fixed-parameter tractability
Multi-parameterization
Kernelization

ABSTRACT

The Cluster Editing problem seeks a transformation of a given undirected graph into a disjoint union of cliques via a minimum number of edge additions or deletions. A multi-parameterized version of the problem is studied, featuring a number of constraints that bound the amounts of both edge-additions and deletions per single vertex, as well as the size of a clique-cluster. We show that the problem remains $\mathcal{NP}$-hard even when only one edge can be deleted and at most two edges can be added per vertex. However, the new formulation allows us to solve Cluster Editing (exactly) in polynomial time when the number of edge-edit operations per vertex is smaller than half the minimum cluster size. In other words, Cluster Editing can be solved efficiently when the number of false positives/negatives per single data element is expected to be small compared to the minimum cluster size. As a byproduct, we obtain a kernelization algorithm that delivers linear-size kernels when the two edge-edit bounds are small constants.

1. Introduction

Given a simple undirected graph $G = (V, E)$ and an integer $k > 0$, the Cluster Editing problem asks whether k or less edge additions or deletions can transform G into a graph whose connected components are cliques. Cluster Editing is $\mathcal{NP}$-Complete [22,26], but it is fixed-parameter tractable with respect to the parameter k [7,18].[1] The problem received considerable attention recently as can be seen from a long sequence of continuous algorithmic improvements (see [4–6,8, 9,18,19]). The current asymptotically fastest fixed-parameter algorithm runs in $O^*(1.618^k)$ time [4]. Moreover, a kernel of order $2k$ was obtained recently in [9]. This means that an arbitrary Cluster Editing instance can be reduced in polynomial time into an equivalent instance where the number of vertices is at most $2k$. The number of edges in the reduced instance can be quadratic in k.

Cluster Editing can be viewed as a model for accurate unsupervised "correlation clustering." In such context, edges to be deleted from or added to a given instance are considered false positives or false negatives, respectively. Such errors could be small in some practical applications, and they tend to be even smaller per input object, or vertex. In fact, a single data element that is causing too many false positives/negatives might be considered as outlier.

We consider a parameterized version of Cluster Editing where both the number of edges that can be deleted and the number of edges that can be added, per vertex, are bounded by input constraints. We refer to these two bounds by *error parameters*. Similar parameterizations appeared in [20] and [21]. In [20], two parameters p and q were used to bound

☆ A preliminary version of a portion of this paper appeared in the 7th Annual International Conference on Combinatorial Optimization and Applications (COCOA 2013) [1].

E-mail address: faisal.abukhzam@lau.edu.lb.

[1] We assume familiarity with the notions of fixed-parameter tractability and kernelization algorithms [10,16,17,25].

http://dx.doi.org/10.1016/j.jda.2017.07.003
1570-8667/

(respectively) (i) the number of edges that can be added between elements of the same cluster and (ii) the number of edges that can be deleted between a cluster and the rest of the graph. Further work on this problem formulation appeared in [24]. In [21], the total number of edge-edit operations, per vertex, and the number of clusters in the target solution are used as additional parameters. Further work on this version appeared recently in [12]. We shall see that setting separate bounds on the two error parameters could affect the complexity of the problem.

We introduce another constraint that bounds, from below, the minimum acceptable cluster size and present a polynomial time algorithm that solves Cluster Editing exactly whenever the sum of error parameters $(a+d)$ is small compared to the minimum cluster size. This condition could be of particular interest in applications where the cluster size is expected to be large or when error parameters per data element are not expected to be high. In this respect, the message conveyed by our work bears the same theme as another, rather experimental, study of various clustering methods conducted in [13], where it was suggested that Clustering is not as hard as claimed by corresponding $\mathcal{NP}$-hardness proofs.

We shall also study the complexity of the multi-parameterized version of Cluster Editing when the error-parameters are small constants. In particular, we show that Cluster Editing remains $\mathcal{NP}$-hard even when at most one edge can be deleted and at most two edges can be added per vertex. Moreover, we show in this case that a simple reduction procedure yields a problem kernel whose total size is linear in the parameter k. Previously known kernelization algorithms cannot be applied to the considered multi-parameterized version and they deliver kernels whose order (number of vertices only) is linear in k.

The paper is organized as follows: Section 2 presents some preliminaries; in Section 3 we study the complexity of Cluster Editing when parameterized by the error-parameters; Section 4 is devoted to a general reduction procedure; the consequent complexity results are presented in Sections 5; and Section 6 concludes with a summary.

2. Preliminaries

We adopt common graph theoretic terminologies[2] such as neighborhood, vertex degree and adjacency. The term non-edge is used to designate a pair of non-adjacent vertices. Given a graph $G=(V,E)$, and a set $S\subset V$, the subgraph induced by S is denoted by $G[S]$. The neighborhood of a vertex $v\in V$ is denoted by $N(v)$ and, for $S\subset V$, the neighborhood of S in G is $N(S)=\cup_{v\in S}N(v)$. A clique in a graph is a subgraph induced by a set of pair-wise adjacent vertices. An edge-editing operation is either a deletion or an addition of an edge. We shall use the term *cluster graph* to denote a transitive undirected graph, which consists of a disjoint union of cliques, as connected components.

For a given graph G and parameter k, the Parameterized Cluster Editing problem asks whether G can be transformed into a cluster graph via k or less edge-editing operations. In this paper, we consider a multi-parameterized version of this problem that assumes a set of constraints, or parameters (independent of the input). The (a,d,s,k)-Cluster Editing is formally defined as follows.

(a,d,s,k)**-Cluster Editing**:
Input: A graph G, parameters a,d,s,k, and two functions $\alpha : V(G)\to\{0,1,\cdots,a\}$, $\delta : V(G)\to\{0,1,\cdots,d\}$.
Question: Can G be transformed into a disjoint union of cliques, each of size s or more, by at most k edge-edit operations such that:
for each vertex $v\in V(G)$, the number of added (deleted) edges incident on v is at most $\alpha(v)$ ($\delta(v)$ respectively)?

We shall further use some special terminology to better present our results. The expression *solution graph* may be used instead of cluster graph, when dealing with a specific input instance. Edges that are not allowed to be in the cluster graph are called *forbidden* edges, while edges that are (decided to be) in the solution graph are *permanent*. Three vertices that induce a path of length two are called a *conflict triple*, which is so named because it can never be part of a solution graph. To *cliquify* a set S of vertices is to transform $G[S]$ into a clique by adding edges.

A clique is permanent if each of its edges is permanent. To *join* a vertex v to a clique C is to add all edges between v and vertices of C that are not in $N(v)$. This operation makes sense only when C is permanent or when turning C to a permanent clique. If v already contains C in its neighborhood, then *joining* v to C is equivalent to making $C\cup\{v\}$ a permanent clique. To *detach* v from C is the opposite operation (of deleting all edges between v and the vertices of C).

The first, and simplest, algorithm for Cluster Editing finds a conflict triple in the input graph and “resolves” it by exploring the three cases corresponding to deleting one of the two edges in the path or inserting the missing edge [18]. In each of the three cases, the algorithm proceeds recursively. As such, the said algorithm runs in $O(3^kn^2)$ time (3 cases per conflict triple). The same idea has been used in almost all subsequent algorithms, which added more sophisticated branching rules.

We first study the (a,d)-Cluster Editing problem, which corresponds to the case where k and s are not parameters or their values are set to infinity and one respectively. This version is similar to the one introduced in [21] where a bound c is placed on the total number of edge-edit operations per single vertex. The corresponding problem is called c-Cluster Editing. When $c\geq 4$, c-Cluster Editing is $\mathcal{NP}$-hard (shown also in [21]). This does not imply, however, that (a,d)-Cluster Editing is $\mathcal{NP}$-hard when $a+d=4$. To see this, note that $(a,0)$-Cluster Editing is solvable in polynomial time for any $a\geq 0$: any

[2] We refer the reader to the book of Diestel [15] for common graph theoretic terminologies.

Please cite this article in press as: F.N. Abu-Khzam, On the complexity of multi-parameterized cluster editing, J. Discret. Algorithms (2017), http://dx.doi.org/10.1016/j.jda.2017.07.003