A kernelization algorithm for the min-max p-cluster editing problem

Li-Hsuan Chen and Bang Ye Wu
Department of Computer Science and Information Engineering
National Chung Cheng University, ChiaYi, Taiwan 621, R.O.C.
{clh100p,bangye}@cs.ccu.edu.tw

Abstract

A p-cluster graph is an undirected graph with at most p connected component and each component is a clique. Given a graph G and an integer p, the Min-Max p-cluster editing *problem asks for the minimum t such that G can be edited into a p-cluster graph by inserting or deleting edges and the maximum number of the editing edges incident to any vertex is at most t. In this paper, we design a kernelization algorithm to reduce the problem instances to kernels of size at most $3pt$.*
Key words: parameterized algorithm, kernelization, cluster graph, graph modification

1 Introduction

Graph clustering is an important issue in computer science. In general, we are given a graph with edges between similar objects, and the goal is to group the similar objects into clusters. Due to the wide applications, there are many formulated problem definitions [22]. A graph-theoretic formulation of one of the clustering problems is the following graph modification problem [23]. A *cluster graph* is an undirected graph consisting of disjoint maximal cliques and the maximal cliques in a cluster graph are called *clusters*. The Cluster Editing problem looks for the minimum number of edge insertions and deletions to modify the input graph to a cluster graph. For an integer p, a cluster graph is a *p-cluster graph* if the number of clusters is no more than p, and the p-Cluster Editing problem asks for a modification to a p-cluster graph. While p-Cluster Editing looks for the min-sum of insertions and deletions, it is natural to consider its min-max version, namely Min-Max p-cluster editing [8]: modifying a graph into p-cluster graph such that the maximum number of editing edges incident to any vertex is minimized.

In this paper, we focus on the decision version of Min-Max p-Cluster Editing. Let $G = (V, E)$ be the input graph and $\pi = (S_1, S_2, \ldots, S_p)$ be a p-partition of V. Each S_i is called a cluster in the p-partition. For $u, v \in V$, $\{u, v\}$ is a *conflict* if the two vertices are in the same cluster but $(u, v) \notin E$ or they are in different clusters but $(u, v) \in E$. Let $C_\pi(v)$ denote the set of vertices conflicting with v in π and $c_\pi(v) = |C_\pi(v)|$ be the *conflict number* of v. Clearly $u \in C_\pi(v)$ if and only if $v \in C_\pi(u)$. As shown in [25], the set of conflicting pairs corresponds to an editing set, i.e., $\bigcup_u \{(u, v) \mid v \in C_\pi(u)\}$ is the corresponding editing set. Thus, we shall call $c_\pi(v)$ the *editing number* of v in π.

Definition 1: A graph is *max t-editable to p-cluster graph* if there is a p-partition π of V such that $\max_{v \in V} c_\pi(v) \leq t$. For simplicity, we call such a graph (p, t)-editable in the remaining paper.

The problem is formally defined as follows.

> Problem: (p, t)-Editable
> Instance: A graph $G = (V, E)$ and integers p and t.
> Question: Is the input graph (p, t)-editable?

The (p, t)-Editable is the decision version of Min-Max p-Cluster Editing and the NP-completeness of the problem is proved in [8]. An instance of a parameterized problem consists of (I, k), where k is the parameter. A problem is *fixed-parameter tractable* (FPT) if it can be solved in time complexity $O(f(k) \cdot q(|I|))$, where f is an arbitrary computable function of k and q is a polynomial in the input size. For more details about parameterized complexity, we refer to the book of Downey and Fellows [11]. *Kernelization* is a widely-used technique for parameterized algorithms. In polynomial time, a kernelization algorithm converts an instance (I, k) to a reduced

instance (I', k'), called a *kernel* such that the answer is not changed, $k' \leq k$ and $|I'|$ is bounded by a computable function of k.

In this paper, using the parameter (p, t), we design a $O(n^3)$-time kernelization algorithm to reduce the problem instances to problem kernels with size at most $3pt$. In [21], the authors studied the p-Cluster editing problem with locally bounded modifications t and showed a $4pt$ kernel. Our kernelization algorithm can be also applied to the problem and improve their result to $3pt$.

Previous related works

Due to the wide applications, several related problems and variants of the clustering problem have been studied, such as Consensus Clustering [13, 6, 5], Correlation Clustering [2, 1, 15, 19, 24] and Cluster Editing [23]. Shamir et al. [23] studied the computational complexities of three edge modification problems. While Cluster Editing asks for the minimum total number of edge insertions and deletions, Cluster Deletion (respectively, Cluster Completion) only allows edge deletions (respectively, insertions). They showed that Cluster Editing is NP-hard, Cluster Deletion is Max SNP-hard but when the number of clusters is constrained by two it becomes polynomial time solvable, and Cluster Completion is polynomial-time solvable. There are several results on the fixed-parameter time complexities for Cluster Editing and Cluster Deletion, for example [3, 7, 9, 10, 16, 17, 18], and the most recent result can be referred to [4]. A variant with vertex (rather than edge) deletions was considered in [20], and another variant in which overlapping clusters are allowed was studied in [12]. For p-Cluster Editing, the currently best parameterized algorithm is due to Formin et el [14]. For the special case $p = 2$, the best time complexity of determining whether a graph can be modified into a 2-cluster graph by editing at most $2k$ edges is $O(n \cdot 2.619^{r/(1-4r/n)} + n^3)$ [25], where n is the number of vertices and $r = k/n$. Particularly, the time complexity is $O^*(2.619^{k/n})$ for $k \in o(n^2)$ and polynomial for $k \in O(n \log n)$, which implies that the problem can be solved in subexponential time when $k \in o(n^2)$.

2 The kernelization algorithm

In this paper, a graph is an undirected simple graph. For a graph G, the vertex set and the edge set are denoted by $V(G)$ and $E(G)$, respectively. For any $v \in V$, the closed neighborhood in graph G is denoted by $N_G[v]$ or $N[v]$ when there is no ambiguity. For two sets S_1 and S_2, the set difference is denoted by $S_1 \setminus S_2$, and the symmetric difference is denoted by $S_1 \ominus S_2 = (S_1 \setminus S_2) \cup (S_2 \setminus S_1)$. For simplicity, $S_1 \ominus v = S_1 \ominus \{v\}$.

The editing number $c_\pi(v)$ of a vertex v in π has been defined in the introduction. Suppose that π is a p-partition of V such that $c_\pi(v) \leq t$ for all $v \in V$. Let S be a cluster in π. We have the following lemmas. For simplicity we omit the subscript π.

Lemma 1: If $u \in S$, then $|S| - t \leq |N[u]| \leq |S| + t$.

Proof: Since the editing number of u is at most t, we have

$$\begin{aligned} c(u) &\leq t \\ \Rightarrow |S \ominus N[u]| &\leq t. \end{aligned}$$

Therefore, we have $|S \setminus N[u]| \leq t$ and $|N[u] \setminus S| \leq t$, which imply $|S| - t \leq |N[u]| \leq |S| + t$. □

Lemma 2: If u and v are in the same cluster, then $|N[u] \ominus N[v]| \leq 2t$.

Proof: Since the editing numbers of both u and v are at most t, we have

$$|S \ominus N[u]| \leq t,$$

and

$$|S \ominus N[v]| \leq t.$$

Since the symmetric difference $\ominus$ is commutative and associative, we have that

$$N[u] \ominus N[v] = (N[u] \ominus S) \ominus (N[v] \ominus S).$$

In addition,

$$|(N[u] \ominus S) \ominus (N[v] \ominus S)| \leq |N[u] \ominus S| + |N[v] \ominus S|.$$

We obtain that $|N[u] \ominus N[v]| \leq |S \ominus N[u]| + |S \ominus N[v]| \leq 2t$.

□

Lemma 3: If $|N[u] \cap N[v]| > 2t$, then u and v must be in the same cluster.

Proof: Suppose that $u \in S$ and $v \in S'$, where $S \neq S'$ are two clusters. Since the editing number of u, v are at most t, we have

$$\begin{aligned} |N[u] \setminus S| \leq c(u) &\leq t \\ \Rightarrow |(N[u] \cap N[v]) \setminus S| &\leq t, \end{aligned}$$

and

$$\begin{aligned} |N[v] \setminus S'| \leq c(v) &\leq t \\ \Rightarrow |(N[u] \cap N[v]) \setminus S'| &\leq t, \end{aligned}$$

Since $S \neq S'$, we have $|S \cap S'| = 0$ and $|N[u] \cap N[v]| \leq 2t$, a contradiction to the assumption that $|N[u] \cap N[v]| > 2t$. Thus, the the lemma is correct. □

Let $\alpha(u, v) = \max\{|N[u]|, |N[v]|\}$ for vertices u and v.

Lemma 4: Suppose that $u \in S$ and $|S| > 3t$. If $|N[u] \cap N[v]| \geq \alpha(u, v)/2$, then $v \in S$.

Proof: First, if $\alpha > 4t$, the result follows from the previous lemma. So we assume that $\alpha \leq 4t$ in the remaining proof. By the definition of α, we have both $|N[u]|$ and $|N[v]|$ at most $4t$.

We show that if $v \notin S$, then $|N[u] \cap N[v]| < \alpha/2$. Suppose that v belongs to another cluster S'. Let $X = S \cap (N[u] \cap N[v])$ and $Y = S' \cap (N[u] \cap N[v])$. Let $x = |X|$, $y = |Y|$, and $r = |N[u] \cap N[v]|$. Since $c(v) \leq t$ and $S \cap S' = \emptyset$, we have

$$\begin{aligned} x &= |S \cap (N[u] \cap N[v])| \\ &\leq |(N[u] \cap N[v]) \setminus S'| \\ &\leq |N[v] \ominus S'| \\ &= c(v) \leq t. \end{aligned} \quad (1)$$

Consider $c(u) = |S \ominus N[u]| = |N[u] \setminus S| + |S \setminus N[u]|$. For the first term, we have $|N[u] \setminus S| \geq r - x$. For the second term, since $|S \cap N[u]| \leq |N[u]| - (r - x)$, we have $|S \setminus N[u]| \geq |S| - (|N[u]| - r + x)$. By the assumption $c(u) \leq t$, we obtain

$$(r - x) + (|S| - (|N[u]| - r + x)) \leq t.$$

Since $|S| > 3t$ and $|N[u]| \leq \alpha$, we have

$$2(r - x) \leq t + |N[u]| - |S| < \alpha - 2t. \quad (2)$$

By (1) and (2), we have $|N[u] \cap N[v]| = r < (\alpha - 2t)/2 + x \leq \alpha/2$, a contradiction to the assumption that $|N[u] \cap N[v]| \geq \alpha(u, v)/2$. Thus, the the lemma is correct. □

Lemma 5: Suppose that $u \in S$ and $|S| > 3t$. For any vertex v with $|N[v]| > 2t$, if $|N[u] \cap N[v]| \leq \alpha(u, v)/2$, then $v \notin S$.

Proof: Since $u \in S$ with $|S| > 3t$, we have $|N[u]| > 2t$. Since

$$\begin{aligned} &|N[u] \ominus N[v]| \\ = \; &|N[u]| + |N[v]| - 2|N[u] \cap N[v]|, \end{aligned}$$

if $2|N[u] \cap N[v]| \leq \max\{|N[u]|, |N[v]|\}$, then

$$|N[u] \ominus N[v]| \geq \min\{|N[u]|, |N[v]|\} > 2t.$$

By Lemma 2, u and v are not in the same cluster. □

By Lemma 1, Lemma 4 and Lemma 5, we have the next necessary and sufficient condition.

Corollary 6 : Suppose that $u \in S$ and $|S| > 3t$. Then, for any vertex v, we have that $v \in S$ if and only if $|N[v]| > 2t$ and $|N[u] \cap N[v]| > \alpha(u, v)/2$.

The kernelization algorithm is shown in Algorithm 1, and the result is given in the next theorem.

Theorem 7: In $O(n^3)$ time, we can determine all clusters with size at least $3t$. Therefore, the problem admits a kernel of size at most $3pt$.

Proof: First we find the set V' of vertices with closed neighborhood size at least $2t$. The remaining vertices are put in R. In each iteration of the while-loop, we find the cluster which u belongs to. If u is in a cluster of size at least $3t$, by Corollary 6, S must be exactly the cluster. Otherwise, u is not at any cluster of size at least $3t$ and is moved to R. When the while-loop terminates, there should be no clusters of size at least $3t$. Therefore, if $|R| > 3p't$, then the algorithm return that there is no feasible solutions.

Since the while-loop can be done in $O(n^2)$ time, the total time complexity is $O(n^3)$. □

3 Concluding remarks

In this paper we design an $O(n^3)$ time kernelization algorithm for Min-Max p-cluster editing to obtain kernels of size at most $3pt$. By a naive branching algorithm, the problem can be solved in $O(p^{3pt})$ time, which is independent of the original problem size n. In other words,

Algorithm 1 :Kernelization

Input: a graph $G = (V, E)$ and integers p, t.
Output: a set $R \subseteq V$ and integers p' or report "False".

construct $N[v]$ for all vertices v;
$V' \leftarrow \{v \in V \mid |N[v]| > 2t\}$, $R \leftarrow V - V'$ and $p' \leftarrow p$;
while $V' \neq \emptyset$ and $p' > 0$ **do**
    pick a vertex $u \in V'$;
    $S \leftarrow \{v \in V' \mid |N[u] \cap N[v]| > 2t\} \cup \{u\}$;
    if $|S| > 3t$ and $|N[v] \ominus S| \leq t$, $\forall v \in S$ **then**
        $V' \leftarrow V' \setminus S$; $p' \leftarrow p' - 1$;
    else
        $V' \leftarrow V' \setminus \{u\}$ and $R \leftarrow R \cup \{u\}$;
    end if
end while
if $|R| > 3p't$ **then**
    report False;
end if
return R and p'.

Min-Max p-cluster editing is FPT(Fixed-Parameter Tractable). Both further reducing the kernel size and designing branching rules to improve the total time complexity are interesting future work.

Furthermore, our kernelization algorithm can be applied to any cluster editing problem with constraints that the local modification and the number of clusters are upper bounded by t and p, respectively, no matter what the objective function is. For example, our result improves the kernelization in [21], in which the author showed a kernel of size $4pt$ for the problem asking for the minimum total number of editing edges with such a constraint.

References

[1] N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):23:1–23:27, Nov. 2008.

[2] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56:89–113, 2004.

[3] S. Böcker, S. Briesemeister, Q. Bui, and A. Truss. Going weighted: Parameterized algorithms for cluster editing. *Theoretical Computer Science*, 410(52):5467–5480, 2009.

[4] S. Böcker and P. Damaschke. Even faster parameterized cluster deletion and cluster editing. *Information Processing Letters*, 111(14):717–721, 2011.

[5] P. Bonizzoni, G. D. Vedova, and R. Dondi. A PTAS for the minimum consensus clustering problem with a fixed number of clusters. In *Eleventh Italian Conference on Theoretical Computer Science*. 2009.

[6] P. Bonizzoni, G. D. Vedova, R. Dondi, and T. Jiang. On the approximation of correlation clustering and consensus clustering. *Journal of Computer and System Sciences*, 74(5):671–696, 2008.

[7] J. Chen and J. Meng. A $2k$ kernel for the cluster editing problem. *Journal of Computer and System Sciences*, 78(1):211–220, 2012.

[8] L.-H. Chen, M.-S. Chang, C.-C. Wang, and B. Y. Wu. On the min-max 2-cluster editing problem. *Journal of Information Science and Engineering*, 29:1109–1120, 2013.

[9] P. Damaschke. Bounded-degree techniques accelerate some parameterized graph algorithms. In J. Chen and F. Fomin, editors, *Parameterized and Exact Computation*, volume 5917 of *Lecture Notes in Computer Science*, pages 98–109. Springer Berlin Heidelberg, 2009.

[10] P. Damaschke. Fixed-parameter enumerability of cluster editing and related problems. *Theory of Computing Systems*, 46:261–283, 2010.

[11] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[12] M. R. Fellows, J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann. Graph-based data clustering with overlaps. *Discrete Optimization*, 8(1):2–17, 2011.

[13] V. Filkov and S. Skiena. Integrating microarray data by consensus clustering. *International Journal on Artificial Intelligence Tools*, 13(04):863–880, 2004.

[14] F. V. Fomin, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and Y. Villanger. Tight bounds for Parameterized Complexity of Cluster Editing. In N. Portier and T. Wilke, editors, *30th International Symposium on Theoretical Aspects of Computer Science (STACS 2013)*, volume 20 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32–43, Dagstuhl, Germany, 2013. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[15] I. Giotis and V. Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(13):249–266, 2006.

[16] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Graph-modeled data clustering: Fixed-parameter algorithms for clique generation. In R. Petreschi, G. Persiano, and R. Silvestri, editors, *Algorithms and Complexity*, volume 2653 of *Lecture Notes in Computer Science*, pages 108–119. Springer Berlin Heidelberg, 2003.

[17] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Automated generation of search tree algorithms for hard graph modification problems. *Algorithmica*, 39:321–347, 2004.

[18] J. Guo. A more effective linear kernelization for cluster editing. *Theoretical Computer Science*, 410(810):718–726, 2009.

[19] F. Harary. On the notion of balance of a signed graph. *The Michigan Mathematical Journal*, 2(2):143–146, 1953.

[20] F. Hüffner, C. Komusiewicz, H. Moser, and R. Niedermeier. Fixed-parameter algorithms for cluster vertex deletion. *Theory of Computing Systems*, 47:196–217, 2010.

[21] C. Komusiewicz and J. Uhlmann. Cluster editing with locally bounded modifications. *Discrete Applied Mathematics*, 160(15):2259–2270, 2012.

[22] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27–64, 2007.

[23] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(12):173–182, 2004.

[24] S. Wasserman and K. Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

[25] B. Y. Wu and L.-H. Chen. Parameterized algorithms for the 2-clustering problem with minimum sum and minimum sum of squares objective functions. *Algorithmica*, in print, on-line available, 2014.