Theoretical Computer Science 609 (2016) 583–593

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the hardness of labeled correlation clustering problem: A parameterized complexity view ☆

Xianmin Liu, Jianzhong Li *, Hong Gao

P.O. Box 750, Harbin Institute of Technology, West DaZhi Street 92, Harbin, 150001, China

ARTICLE INFO

Article history:
Received 29 October 2014
Received in revised form 9 March 2015
Accepted 13 March 2015
Available online 18 March 2015

Keywords:
Labeled correlation clustering
Parameterized complexity
Lower bound
Strong Exponential Time Hypothesis

ABSTRACT

Motivated by practical applications, the Labeled Correlation Clustering problem, a variant of Correlation Clustering problem, is formally defined and studied in this paper. Since the problem is NP-*complete*, we consider the parameterized complexities. Three different parameterizations are considered, and the corresponding parameterized complexities are studied. For the two parameterized problems which are fixed-parameter-tractable, the lower bounds of them are analyzed under SETH (Strong Exponential Time Hypothesis).

1. Introduction

A general *graph*[1] G is composed of a node set V_G and an edge set $E_G \subseteq V_G \times V_G$, denoted by $G = (V_G, E_G)$. A graph G is *clustered* if every connected component of G is a clique. An *edge labeled* graph, *el*-graph for short, can be denoted by $\widetilde{G} = (V_{\widetilde{G}}, E_{\widetilde{G}}, f_{\widetilde{G}})$ where $V_{\widetilde{G}}$ and $E_{\widetilde{G}}$ define a general graph and $f_{\widetilde{G}}$ is a mapping $E_{\widetilde{G}} \mapsto \{0, 1\}$. Graph G *disagrees* with an *el*-graph $\widetilde{G}$, if there is some edge $e \in E_{\widetilde{G}}$ such that $f_{\widetilde{G}}(e) = 1 \wedge e \notin E_G$ or $f_{\widetilde{G}}(e) = 0 \wedge e \in E_G$, and such edges are called *disagreed edges* between G and $\widetilde{G}$. G *agrees* with $\widetilde{G}$, if there are no disagreed edges between them. Given an *el*-graph set $\widetilde{\mathcal{G}} = \{\widetilde{G}_1, \ldots, \widetilde{G}_m\}$ and a graph G, let Disagree be a function such that $\textsc{Disagree}(G, \widetilde{\mathcal{G}})$ is the number of graphs in $\widetilde{\mathcal{G}}$ with which G disagrees.

In the classic Correlation Clustering problem (CC for short) [1], given an *el*-graph $\widetilde{G}$, the goal is to find a *clustered* graph G such that the size of disagreed edges between G and $\widetilde{G}$ is minimum. It has many applications, such as entity identification [2], coreference resolution [3] and so on. The CC problem is NP-*hard*, and there have been a lot of works focusing on it, for example [1,4–6].

In this paper, a variant of Correlation Clustering is studied, which is called Labelled Correlation Clustering (LCC for short) and can be defined as follows. The input of an LCC instance is an *el*-graph set $\widetilde{\mathcal{G}} = \{\widetilde{G}_1, \ldots, \widetilde{G}_m\}$, the goal is to find a *clustered* graph G such that the size $\textsc{Disagree}(G, \widetilde{\mathcal{G}})$ is minimum.

☆ This work was supported in part by the National Basic Research Program of China (973 Program) under Grant No. 2012CB316200, the Major Program of National Natural Science Foundation of China under grant 61190115, the Key Program of the National Natural Science Foundation of China under grant 61033015, 60933001, the General Program of the National Natural Science Foundation of China under grant 61402130 and the Shandong Provincial Natural Science Foundation of China under grant no. ZR2013FQ028.

* Corresponding author.
E-mail addresses: liuxianmin@hit.edu.cn (X. Liu), lijzh@hit.edu.cn (J. Li), honggao@hit.edu.cn (H. Gao).

[1] We only consider *undirected graph* here.

http://dx.doi.org/10.1016/j.tcs.2015.03.021
0304-3975/© 2015 Elsevier B.V. All rights reserved.

This problem has many applications. A typical example comes from the entity identification problem in managing dirty data [7]. Suppose there are a set of records, each of them contains the values of several attributes about one person. For example, the record $\{name = \text{"Bob"}, age = \text{"12"}, \dots\}$ represents one person named Bob is 12-year-old. There may be several records describing the same person, and the entity identification problem is to find a clustering way for the records such that each cluster exactly contains all records of one person. To solve entity identification problem directly is difficult, previous works usually focus on *entity matching* problem. Given two records, an entity matching algorithm will determine whether they represent the same person. Treating each record as a node in graph and using the edge between nodes to represent the two corresponding records belong to the same person, the output of an entity matching algorithm can be a general graph, while the output of entity identification problem is required to be a *clustered* graph. To fill the gap between the output of entity matching and entity identification, intuitively, the CC problem is to transform the entity matching result to the entity identification result while minimizing the difference between them. Given the same records, different matching algorithms may output different results, combining the results of multiple matching algorithms brings opportunities to improve the accuracy of identification result [8]. Given the results of several matching algorithms, the LCC problem is to generate the identification result such that it is compatible to as many matching algorithms as possible. Another application example can be found in information retrieval area. The core problem in the area of retrieving information is to rank a set of alternatives based on possibly conflicting preferences, which can be formalized as combining k different rank lists into a single one, known as *rank aggregation* problem [9]. A natural idea is to integrate several rank aggregation methods to a single one and output consistent ranking results. The LCC problem studied by this paper is just the formalization of that idea.

The LCC problem has been already proved to be NP-*hard* [10]. In fact,treating each labeled edge in CC as an *el*-graph, CC is a special case of LCC. Therefore, we analyze the LCC problem from the point of view of parameterized complexity [11], to study whether there are efficient algorithms when some parameters of the input are small. According to [12], a parameterized problem is a set $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed alphabet and $\mathbb{N}$ is the positive integer set. For $(x, k) \in \Sigma^* \times \mathbb{N}$, x is the input and k is the parameter. A parameterized problem P is *fixed-parameter tractable* if there is a computable function $f : \mathbb{N} \to \mathbb{N}$, a constant $c \in \mathbb{N}$, and an algorithm that, given a pair $(x, k) \in \Sigma^* \times \mathbb{N}$, decides if $(x, k) \in P$ in at most $f(k) \cdot |x|^c$ steps. The class of all the fixed-parameter tractable problems is FPT. Beyond FPT, a hierarchy of parameterized complexity classes FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ W[P] have been defined by [13–15], which play a central role in identifying parameterized intractable problems. For example, the standard parameterization version of classic Clique problem is W[1]-*complete*, and the standard parameterization version of Dominating Set problem is W[2]-*complete*, which implies that Dominating Set is intuitively harder than Clique. To study the LCC problem from the perspective of parameterized complexity, we consider different parameterization methods for LCC and study which class of W-hierarchy each parameterized LCC belongs to.

In this paper, the parameterized versions of the LCC problem, denoted by p-LCC, are defined on the following three parameters.

Problem: p-LCC
Instance: An *el*-graph set $\widetilde{\mathcal{G}} = \{\widetilde{G}_1, \dots, \widetilde{G}_m\}$ and a positive integer k.
Question: Is there a clustered graph G such that $\text{Disagree}(G, \widetilde{\mathcal{G}})$ is not larger than k?
Parameter 1: $m = |\widetilde{\mathcal{G}}|$, the size of the *el*-graph set.
Parameter 2: $n = |\bigcup\{V_{\widetilde{G}_i}\}|$, the size of node set of graphs in $\widetilde{\mathcal{G}}$.
Parameter 3: k.

The three parameterized problems are denoted by p-LCCm, p-LCCn, and p-LCCk respectively.

Our results. Because the LCC problem is NP-*hard*, a natural question is whether it is fixed-parameter tractable. That is, when some parameter of one LCC instance is small, whether or not it can be solved efficiently. Three parameterization methods are considered. When the LCC problem is parameterized with m and n, we give the positive answers. Specifically, the corresponding problems p-LCCm and p-LCCn are shown to be fixed-parameter tractable, and they can be solved in time $O(2^m \cdot p(|x|))$ and $O(2^{n \log n} \cdot p(|x|))$ where $|x|$ is the length of input and p is a polynomial function. When it is parameterized with k, negative answer is given. By means of checking *bad* circles which will be introduced later, it is shown that the p-LCCk problem is W[t]-*hard* for any $t > 0$ and in W[P]. It means that, unless for any $t > 0$ we have FPT = W[t], the p-LCCk problem is not fixed-parameter tractable. Moreover, for the two tractable problems, p-LCCm and p-LCCn, a natural question is whether the given algorithms are optimal. We study the lower bounds for that two problems under the SETH assumption, shows that the proposed algorithm for p-LCCm is optimal and the algorithm for p-LCCn is "nearly optimal".

1.1. Related work

To the best of our knowledge, there are only few works focusing on the Labelled Correlation Clustering problem. In [10], the LCC problem is proved to be NP-*complete*, and an $O(\sqrt{n \log k})$-approximation algorithm is given where n is the input size and k is $|\widetilde{\mathcal{G}}|$.

Some other related works focus on the classic Correlation Clustering problem. Treating each edge e labeled 1 (resp. 0) as a requirement for the existence (resp. inexistence) of e, the similarity of that two problems is to seek to find a clustered