A Local Search for a Graph Correlation Clustering

Victor Il'ev[1,2] and Anna Navrotskaya[1,2]

[1] Sobolev Institute of Mathematics,
4 Acad. Koptyug Ave., 630090 Novosibirsk, Russia,
[2] Omsk State University,
55a Mira Ave., 644077 Omsk, Russia
iljev@mail.ru, nawrocki@ya.ru

Abstract. In the clustering problems one has to partition a given set of objects into some subsets (called clusters) taking into consideration only similarity of the objects. We consider a version of the clustering problem when the number of clusters does not exceed a positive integer k and the goal is to minimize the number of edges between clusters and the number of missing edges within clusters. This problem is NP-hard for any $k \geq 2$. We propose a polynomial time k-approximation algorithm for this problem.

Keywords: Graph clustering, local search, approximation algorithm.

1 Introduction

We consider only the *simple* graphs, i.e., the graphs without loops and multiple edges. A graph is called *cluster graph* [11] if each of its connected components is a complete graph. Let V be a finite set. Denote by $\mathcal{M}(V)$ the set of all cluster graphs on the vertex set V; by $\mathcal{M}_k(V)$, the set of all cluster graphs on V consisting of exactly k nonempty connected components, and by $\mathcal{M}_{1,k}(V)$, the set of all cluster graphs on V consisting of at most k connected components, $2 \leq k \leq |V|$.

If $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$ are graphs on the same vertex set V, then the distance $d(G_1, G_2)$ between them is defined as follows:

$$d(G_1, G_2) = |E_1 \setminus E_2| + |E_2 \setminus E_1|,$$

i.e., $d(G_1, G_2)$ is the number of noncoinciding edges in G_1 and G_2.

In the *clustering problems* one has to partition a given set of objects (a data set) into some subsets (called *clusters*) taking into consideration only similarity of the objects.

Bansal, Blum and Chawla [3] proposed the following version of the clustering problem:

CORRELATION CLUSTERING (**CC**). Given a complete graph $G = (V, E)$ with edges labelled $+1$ (similar) or -1 (different), find an *optimal clustering*, i.e., a partition of the vertex set of G into clusters minimizing disagreements (the number of -1 edges inside the clusters plus the number of $+1$ edges between clusters).

In: A. Kononov et al. (eds.): DOOR 2016, Vladivostok, Russia, published at http://ceur-ws.org

The variants of **CC** in which the number of clusters is bounded were also studied [3, 6].

One of the most visual formalizations of the clustering problem is the *graph clustering* [10], that is grouping the vertices of a graph into clusters taking into consideration the edge structure of the graph whose vertices are objects and edges represent similarities between the objects. In this setting, the clustering can be understood as cluster graph M whose connected components correspond to the clusters.

Obviously, the following setting is equivalent to **CC**.

GRAPH CORRELATION CLUSTERING (**GCC**). Given a graph $G = (V, E)$, find a graph $M^* \in \mathcal{M}(V)$ such that

$$d(G, M^*) = \min_{M \in \mathcal{M}(V)} d(G, M).$$

We focus on the following bounded variants of GRAPH CORRELATION CLUSTERING.

GCC$_\mathbf{k}$. Given a graph $G = (V, E)$ and an integer k, $2 \leq k \leq |V|$, find a graph $M^* \in \mathcal{M}_k(V)$ such that

$$d(G, M^*) = \min_{M \in \mathcal{M}_k(V)} d(G, M).$$

GCC$_{\mathbf{1,k}}$. Given a graph $G = (V, E)$ and an integer k, $2 \leq k \leq |V|$, find a graph $M^* \in \mathcal{M}_{1,k}(V)$ such that

$$d(G, M^*) = \min_{M \in \mathcal{M}_{1,k}(V)} d(G, M).$$

In Section 2 of this paper we present a short survey of the known results involving results on computation complexity and approximability of different versions of the graph clustering problem. In section 3 we propose a new k-approximation algirithm for **GCC$_{\mathbf{1,k}}$**.

2 Known results

Apparently, NP-hardness of problem **GCC** was first proved by Křivánek and Morávek [9] in 1986.

At the beginning and in the middle of the 2000s, several groups of authors were independently dealing with the different versions of the graph clustering problems. Chen, Jiang, and Lin [5] considered the closest phylogenetic root problems and also proved NP-hardness of **GCC**. Bansal, Blum, and Chawla [3], using a reduction from PARTITION INTO TRIANGLES, show that **GCC** is NP-hard even if all clusters are of size 3 as a maximum.

Shamir, Sharan, and Tsur [11] independently showed NP-hardness of problem **GCC** by a reduction from the 3-exact 3-Cover problem. They also reduced the known NP-complete problem of 2-coloring of 3-Uniform Hypergraph to problem **GCC$_\mathbf{2}$** and as

a result they showed that problem $\mathbf{GCC_k}$ is NP-hard for any fixed $k \geq 2$. In both cases Shamir, Sharan, and Tsur use rather complicated reduction. Later, Giotis and Guruswami [7] published a more simple proof of the same result using a polynomial reduction from the graph bisection problem.

At the same time, Ageev, Il'ev, Kononov, and Talevnin [1] independently proved that problems $\mathbf{GCC_2}$ and $\mathbf{GCC_{1,2}}$ are NP-hard on cubic (i.e., 3-regular) graphs and deduced from this that all the above-mentioned variants of the graph clustering problems (including $\mathbf{GCC_{1,k}}$) are NP-hard.

In 2004, Bansal, Blum, and Chawla [3] presented a simple polynomial time 3-approximation algorithm for $\mathbf{GCC_{1,2}}$. For each $v \in V$ their algorithm considers the following pair of clusters. The first cluster contains v and all neighbors of v in $G = (V, E)$. The second cluster contains all other vertices. The algorithm outputs the pair that minimizes the number of mismatched edges. More formally this algorithm may be described as follows.

Algorithm $\mathbf{N_{1,2}}(G)$.

Step 1. For each vertex $v \in V$ construct the cluster graph $M_v \in \mathcal{M}_{1,2}(V)$ with $V_1 = \{v\} \cup N(G)$ and $V_2 = V \setminus V_1$ as the vertex sets of connected components of M_v.

Step 2. Pick the graph M_v with minimal distance from G, i.e.,

$$d(G, M) = \min_{v \in V} d(G, M_v).$$

In 2006, Ageev, Il'ev, Kononov, and Talevnin [1] proved the existence of a randomized PTAS for problem $\mathbf{GCC_{1,2}}$ by reducing this problem to the graph bisection problem, and Giotis and Guruswami [7] presented a randomized PTAS for problem $\mathbf{GCC_k}$ (for any fixed $k \geq 2$). In 2007, Il'ev, Navrotskaya, and Talevnin [8] considered a local search algorithm for problem $\mathbf{GCC_{1,2}}$. They showed that if the number of edges in a graph is subquadratic of the number of vertices, the local search algorithm is asymptotically exact.

In 2008, Coleman, Saunderson, and Wirth [6] pointed out that complexity of PTAS from [7] make this scheme practically useless. They presented a 2-approximation algorithm for problem $\mathbf{GCC_{1,2}}$ applying local search to the feasible solution obtained by the 3-approximation algorithm from [3].

In 2005, Charicar, Guruswami, and Wirth [4] proved that problem **GCC** is APX-hard. They also constructed a 4-approximation algorithm for problem **GCC** by rounding a natural LP relaxation using the region growing technique. In 2008, Ailon, Charicar, and Newman [2] proposed a randomized 2.5-approximation algorithm for **GCC**.

3 An approximation algorithm for $\mathbf{GCC_{1,k}}$

As it was already said, Coleman, Saunderson, and Wirth [6] presented a 2-approximation algorithm for problem $\mathbf{GCC_{1,2}}$ applying local search to the feasible solution obtained by the 3-approximation algorithm $\mathbf{N_{1,2}}$ from [3]. Extending this strategy we propose a polynomial time approximation algorithm $\mathbf{NLS_{1,k}}$ for problem $\mathbf{GCC_{1,k}}$ when $k > 2$ and use algorithm from [6] for $k = 2$.

First we describe the algorithm formally. Denote by $N(v)$ the set of all neighbors of a vertex v.

Algorithm $\mathbf{NLS_{1,k}}$

Input: Given graph $G = (V, E)$ and $k > 2$.

Step 1. For each vertex $v \in V$ construct the cluster graph $M_v \in \mathcal{M}_{1,k}(V)$ as follows: if $\{v\} \cup N(v) = V$, then $M_v = K_{|V|}$, else $M_v = \mathbf{N_{1,k}}(G, \{v\} \cup N(v), k)$.

Step 2. Use procedure $\mathbf{LS_{1,k}}$ to modify each graph M_v.

Step 3. Pick the graph M_v with minimal distance from G, i.e.,

$$d(G, M) = \min_{v \in V} d(G, M_v).$$

Function $\mathbf{N_{1,k}}(G, V_1, k)$.

Step 0. Let $i = 1$.

Step 1. Denote by G_i the subgraph of graph G induced by the vertex set $V \setminus (V_1 \cup V_2 \cup \ldots \cup V_i)$.

Step 2. $V_{i+1} = \mathbf{N_{1,2}}(G_i)$.

Step 3. If $V \setminus (V_1 \cup V_2 \cup \ldots \cup V_{i+1}) \neq \emptyset$ and $i < k - 2$, then $i = i + 1$ and go to step 1. If $V \setminus (V_1 \cup V_2 \cup \ldots \cup V_{i+1}) = \emptyset$, then $V_{i+2} = V_{i+3} = \ldots = V_k = \emptyset$. If $i = k - 2$, then $V_k = V \setminus (V_1 \cup V_2 \cup \ldots \cup V_{k-1})$.

Step 4. The sets $V_1, V_2, \ldots, V_k$ induce the connected components of the cluster graph M.

Return: The cluster graph M.

Function $\mathbf{N_{1,2}}(G)$.

Step 1. For each vertex $v \in V$ construct the cluster graph $M_v \in \mathcal{M}_{1,2}(V)$ with $V_1 = \{v\} \cup N(G)$ and $V_2 = V \setminus V_1$ as the vertex sets of connected components of M_v.

Step 2. Pick the graph M_v with minimal distance from G, i.e.,

$$d(G, M) = \min_{v \in V} d(G, M_v).$$

Return: Subset V_1 of the vertex set of the graph M.

For any vertex $u \in V$ define the value

$$imp_u(V_i, V_j) = V_i^+(u) + V_j^-(u) - V_i^-(u) - V_j^+(u),$$

where $V_i^+(u)$ is the number of vertices in the set V_i adjacent to u and $V_i^-(u)$ is the number of vertices in V_i not adjacent to u.

Procedure $\mathbf{LS_{1,k}}$

Input: A graph $G = (V, E)$ and a cluster graph $M \in \mathcal{M}_{1,k}(V)$. The sets $V_1, \ldots, V_k$ are the vertex sets of the connected components of M.

Step s ($s = 1, 2, ...$). Pick the vertex $v \in V$ such that

$$imp_v(V_p, V_q) = \max_{\substack{i=1,\ldots k,\\ u \in V_i}} \Big(\max_{\substack{j=1,\ldots k,\\ j \neq i}} imp_u(V_i, V_j) \Big).$$

If $imp_v(V_p, V_q) > 0$, then move v from V_p to V_q and go to step $s+1$, else stop.

Comments. Algorithm $\mathbf{NLS_{1,k}}$ uses the function $\mathbf{N_{1,k}}$ for constructing a cluster graph and the local search procedure $\mathbf{LS_{1,k}}$ for improving the found cluster graph. In the end the algorithm picks a cluster graph 'nearest' to G.

Function $\mathbf{N_{1,k}}$ makes a partition of the vertex set of a given graph $G = (V, E)$. Consider the subgraph of G without the subsets of the vertex set V constructed at the previous step. Function $\mathbf{N_{1,2}}$ solves problem $\mathbf{GCC_{1,2}}$ approximately and returns a vertex set of the first connected component of the found cluster graph. This subset is added to already constructed ones. At the next step the algorithm considers the subgraph without all found subsets involving the last constructed subset. At step 4 we have k subsets (some of which may be empty) which are a partition of the vertex set of given graph G. This partition induces a cluster graph M and the algorithm returns M.

Function $\mathbf{N_{1,2}}$ takes as an input a graph G and for each vertex v of G constructs a cluster graph whose first connected component has vertex set consisting of v and all adjacent to v vertices. The second connected component consists of all other vertices of G. Later, cluster graph 'nearest' to G among all the constructed cluster graphs is selected. Function $\mathbf{N_{1,2}}$ returns the vertex set of the first connected component of the cluster graph, i.e., vertex v and all its neighbors.

Procedure $\mathbf{LS_{1,k}}$ takes as an input a given graph $G = (V, E)$ and a cluster graph $M \in \mathcal{M}_{1,k}(V)$ constructed by algorithm $\mathbf{N_{1,k}}$. On each iteration the algorithm finds a vertex whose moving to one of the rest connected components leads to steepest decreasing the distance bitween graphs G and M. If such vertex is found, it is moved to the respective corresponding connected component, else the algorithm finishes.

The followimg theorem shows that algorithm $\mathbf{NLS_{1,k}}$ is k-approximation algorithm for problem $\mathbf{GCC_{1,k}}$.

Theorem 1. *For any graph $G = (V, E)$ and an integer k, $2 \le k \le |V|$, the following bound occurs:*

$$d(G, M) \le kd(G, M^*),$$

where M is the cluster graph constucted by algorithm $\mathbf{NLS_{1,k}}$ and M^* is an optimal solution to $\mathbf{GCC_{1,k}}$.

Acknowledgement

This research was supported by the RSF grant 15-11-10009.

References

1. Ageev, A. A., Il'ev, V. P., Kononov, A. V., Talevnin, A. S.: Computational complexity of the graph approximation problem. Diskretnyi Analiz i Issledovanie Operatsii. Ser. 1. 13(1), 3–11 (2006) (in Russian). English transl. in J. of Applied and Industrial Math. 1(1), 1–8 (2007)

2. Ailon, N., Charikar, M., Newman, A.: Aggregating inconsistent information: Ranking and clustering. J. ACM. 55(5), 1–27 (2008)
3. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning. 56, 89–113 (2004)
4. Charikar, M., Guruswami, V., Wirth, A.: Clustering with qualitative information. J. Comput. Syst. Sci. 71(3), 360–383 (2005)
5. Chen, Z.-Z., Jiang, T., Lin, G.: Computing phylogenetic roots with bounded degrees and errors. SIAM J. Comput. 32(4), 864–879 (2003)
6. Coleman, T., Saunderson, J., Wirth, A.: A local-search 2-approximation for 2-correlation-clustering. In: Algorithms – ESA 2008. LNCS, vol. 5193, pp. 308–319. Springer, Heidelberg (2008)
7. Giotis, I., Guruswami, V.: Correlation clustering with a fixed number of clusters. Theory of Computing. 2(1), 249–266 (2006)
8. Il'ev, V. P., Navrotskaya, A. A., Talevnin, A. S.: Polynomial time approximation scheme for the graph approximation problem. Vestnik Omskogo Universiteta. 4, 24–27 (2007) (in Russian)
9. Křivánek, M., Morávek, J.: NP-hard problems in hierarchical-tree clustering. Acta informatica. 23, 311–323 (1986)
10. Schaeffer, S. E.: Graph clustering. Computer Science Review. 1(1), 27–64 (2007)
11. Shamir, R., Sharan, R., Tsur, D.: Cluster graph modification problems. Discrete Appl. Math. 144(1–2), 173–182 (2004)