Cluster Graph Modification Problems

Ron Shamir[*] Roded Sharan[*] Dekel Tsur[*]

Abstract

In a clustering problem one has to partition a set of elements into homogeneous and well-separated subsets. From a graph theoretic point of view, a cluster graph is a vertex-disjoint union of cliques. The clustering problem is the task of making fewest changes to the edge set of an input graph so that it becomes a cluster graph. We study the complexity of three variants of the problem. In the Cluster Completion variant edges can only be added. In Cluster Deletion, edges can only be deleted. In Cluster Editing, both edge additions and edge deletions are allowed. We also study these variants when the desired solution must contain a prespecified number of clusters.

We show that Cluster Editing is NP-complete, Cluster Deletion is NP-hard to approximate to within some constant factor, and Cluster Completion is polynomial. When the desired solution must contain exactly p clusters, we show that Cluster Editing is NP-complete for every $p \geq 2$; Cluster Deletion is polynomial for $p = 2$ but NP-complete for $p > 2$; and Cluster Completion is polynomial for any p. We also give a constant factor approximation algorithm for Cluster Editing when $p = 2$.

1 Introduction

Problem Definition and Motivation: Clustering is a central optimization problem with applications in numerous fields including computational biology (cf. [15]), image processing (cf. [16]), VLSI design (cf. [7]), and many more. The input to the problem is typically a set of elements and pairwise similarity values between elements. The goal is to partition the elements into subsets, which are called *clusters*, so that two meta-criteria are satisfied: *Homogeneity* - elements inside a cluster are highly similar to each other; and *separation* - elements from different clusters have low similarity to each other. Concrete realizations of these criteria generate a variety of combinatorial optimization problems (cf. [8]).

In the basic graph theoretic approach to clustering, one builds from the raw data a *similarity graph* whose vertices correspond to elements and there is an edge between two vertices if and only if the similarity of their corresponding elements exceeds a predefined threshold (cf. [8, 9]). Ideally, the resulting graph would be a *cluster graph*, that is, a graph composed of vertex-disjoint cliques. In practice, it is only close to being such, since similarity data is experimental and, therefore, error-prone.

Following [2] we formalize the resulting problem as the task of changing (adding or deleting) fewest edges of an input graph so as to obtain a cluster graph. We call this problem *Cluster Editing*. In the related *Cluster Deletion* problem one has to remove fewest edges from an input graph so that it becomes a cluster graph. The *Cluster Completion* problem is to add fewest edges to the input graph to make it a cluster graph. Completion (deletion) problems arise when the data contains only false negative (positive) errors. The above problems belong to the class of *edge modification problems* (cf. [13]), in which one has to minimally change the edge set of a graph so as to satisfy a certain property. Another variant of these problems arises when the solution is also required to contain a prespecified number of clusters. This variant is motivated by many real-life applications where a partition of elements into a known number of categories is desired (see, e.g., [1, 6]).

[*] School of Computer Science, Tel-Aviv University, Tel-Aviv 69978, Israel. Email: {rshamir,roded,dekelts}@post.tau.ac.il.

Previous Results: Edge modification problems were studied extensively in [13] where earlier studies are also reviewed. Most of these problems were shown to be NP-complete. Polynomial algorithms were given for bounded degree input graphs. In particular, a constant factor approximation algorithm was given for editing and deletion problems with respect to any property that can be characterized by a finite set of forbidden induced subgraphs. Since a graph is a cluster graph if and only if it is P_2-free (i.e., does not contain an induced path of two edges), this result implies a $3d$-approximation algorithm for Cluster Editing and Cluster Deletion on input graphs with degree bounded by d.

The Cluster Editing problem was first studied by Ben-Dor et al. [2], who presented a polynomial algorithm that solves the problem with high probability under a stochastic data model. The complexity of the problem was left open. The Cluster Deletion problem was shown to be NP-complete by Natanzon [12].

Contribution of This Paper: We prove that Cluster Editing is NP-complete, Cluster Deletion is NP-hard to approximate to within some constant factor, and Cluster Completion is polynomial. We also study the p-Cluster versions of these problems, in which the required graph must also be a vertex-disjoint union of p cliques. We show that p-Cluster Editing is NP-complete for every $p \geq 2$; p-Cluster Deletion is polynomial for $p = 2$ but NP-complete for $p > 2$; and p-Cluster Completion is polynomial for any p. We also give a 0.878-approximation algorithm for a weighted variant of 2-Cluster Editing.

Organization of The Paper: Section 2 contains terminology and problem definitions. In Section 3 we prove the NP-completeness of the Cluster Editing variants, and provide a 0.878-approximation algorithm to a weighted variant of 2-Cluster Editing. In Section 4 we give polynomial algorithms for the Cluster Completion variants. Finally, in Section 5 we study the complexity of the Cluster Deletion variants. For lack of space some proofs are only sketched or omitted.

2 Preliminaries

All graphs in this paper are simple, i.e., contain no parallel edges or self-loops. Let $G = (V, E)$ be a graph. We denote its set of vertices also by $V(G)$, and its set of edges also by $E(G)$. For a vertex $v \in V$, we denote by $N_G(v)$ the set of its neighbors in G. For a set $S \subseteq V$, we denote by G_S the subgraph of G induced by the vertices in S. For two disjoint subsets $A, B \subseteq V$, we denote by $E_{A,B}$ ($\overline{E}_{A,B}$) the set of all edges (non-edges) with one endpoint in A and the other in B. The *complement graph* of G is $\overline{G} = (V, \{(u,v) \in (V \times V) \setminus E : u \neq v\})$. See [3] for more definitions of graphs and hypergraphs.

A graph $G = (V, E)$ is called a *cluster graph* if every connected component of G is a complete graph. G is called a *p-cluster graph* if it is a cluster graph with p connected components, or equivalently, if it is a vertex-disjoint union of p cliques. If G is any graph and $F \subset V \times V$ is such that $G' = (V, E \triangle F)$ is a cluster graph, then F is called a *cluster editing set* for G ($E \triangle F$ denotes the symmetric difference between E and F, i.e., $(E \setminus F) \cup (F \setminus E)$). If in addition $F \subseteq E$, then F is called a *cluster deletion set* for G. If $F \cap E = \phi$ then F is called a *cluster completion set* for G. *p-cluster editing set*, *p-cluster deletion set*, and *p-cluster completion set* are similarly defined. We denote by $P(F)$ the partition of V into disjoint subsets of vertices according to the connected components (cliques) of G'. For a partition $P = (V_1, \ldots, V_l)$ of V, we denote by N_P the size of the cluster editing set implied by P, that is,

$$N_P \equiv |\bigcup_{i=1}^{l} \{(u,v) \notin E : u, v \in V_i\} \cup \{(u,v) \in E : u \in V_i, v \in V_j, i \neq j\}| \ .$$

The problems we study in this paper are of two types:

Problem 1 (Cluster Editing/Completion/Deletion) *Given a graph G and an integer k, determine if G has a cluster editing/completion/deletion set of size at most k.*

Problem 2 (p-Cluster Editing/Completion/Deletion) *Given a graph G and an integer k, determine if G has a p-cluster editing/completion/deletion set of size at most k.*

3 Cluster Editing

We prove in this section that Cluster Editing is NP-complete by reduction from a restriction of exact cover by 3-sets:

Problem 3 (3-Exact 3-Cover (3X3C)) *Given a collection C of triplets of elements from a set $U = \{u_1, \ldots, u_{3n}\}$, such that each element of U is a member of at most 3 triplets, determine if there exists a sub-collection $I \subseteq C$ of size n which covers U.*

The 3X3C problem is known to be NP-complete [4, Problem SP2].

Theorem 1 *Cluster Editing is NP-complete.*

Proof: Membership in NP is trivial. We prove NP-hardness by reduction from 3X3C. Let $m \equiv 30n$. Given an instance $< C, U >$ of 3X3C we build a graph $G = (V, E)$ as follows:

$$\begin{aligned}
V &= \bigcup_{S \in C} \{v_1(S), \ldots, v_m(S)\} \cup U \ , \\
E &= E_1 \cup E_2 \cup E_3 \ , \\
E_1 &= \{(v_i(S), u) : S \in C, 1 \leq i \leq m, u \in S\} \ , \\
E_2 &= \{(v_i(S), v_j(S)) : S \in C, 1 \leq i < j \leq m\} \ , \\
E_3 &= \{(u, u') : \exists S \in C \text{ s.t. } u, u' \in S\} \ .
\end{aligned}$$

In words, we build a clique of size $m + 3$ around each triplet S by fully connecting S and m additional vertices. For each triplet $S \in C$ we denote $V_S = \{v_1(S), \ldots, v_m(S)\}$ and call the elements of V_S, *S-vertices*. Let $q = \sum_{S \in C} |S| = 3|C|$. Define $N \equiv m(q - 3n)$ and $M \equiv |E_3| - 3n$. We prove that there is an exact cover of U iff there is a cluster editing set for G of size at most $N + M$:

$\Rightarrow$ Suppose that $I \subseteq C$ is an exact cover of U. Let $F_1 = \{(v_i(S), u) : S \notin I, 1 \leq i \leq m, u \in S\}$ and let $F_2 = \{(u, u') \in E_3 : \nexists S \in I \text{ s.t. } u, u' \in S\}$. It is easy to verify that $F = F_1 \cup F_2$ is a cluster editing set for G, whose size is $|F| = |F_1| + |F_2| = N + M$.

$\Leftarrow$ Let F' be a cluster editing set for G with $|F'| \leq N + M$. Let F be an optimum cluster editing set for G. Then $|F| \leq |F'| \leq N + M$. We shall prove that $|F| = N + M$ and one can derive from F an exact cover of U. This implies that $|F'| = |F|$, and hence, F' is an optimum cluster editing set from which an exact cover of U can be obtained.

Since each element of U occurs in at most 3 triplets, $q \leq 9n$. Also, $|N_G(u) \cap U| \leq 6$ for each vertex $u \in U$, implying that $|E_3| \leq \frac{3n \cdot 6}{2} = 9n$. Hence, $|F| \leq N + M \leq 6mn + 6n = 180n^2 + 6n < \frac{m}{2}(\frac{m}{2} - 2)$.

Let $G' = (V, E \triangle F)$ be the cluster graph obtained by editing G according to F. We shall prove that for every subset $S \in C$ there exists a unique clique in G' which contains V_S. To this end, we first show that there exists a clique K_S in G' such that $|K_S \cap V_S| \geq m/2 + 3$: Suppose that the vertices of V_S are partitioned among k cliques $X_1, \ldots, X_k$ in G'. Let $s(X_i) = |V_S \cap X_i|, i = 1, \ldots, k$. Suppose to the contrary that $s(X_i) \leq m/2 + 2$ for all i. Therefore,

$$2|F| \geq \sum_{i=1}^{k} s(X_i)(m - s(X_i)) \geq \sum_{i=1}^{k} s(X_i)(\frac{m}{2} - 2) = m(\frac{m}{2} - 2) \ .$$

A contradiction follows.

Let K_S be the clique X_i for which $s(X_i)$ is maximum ($|K_S \cap V_S| \geq m/2 + 3$). We next prove that $V_S \subseteq K_S \subseteq V_S \cup S$. Let $x = |K_S \setminus (V_S \cup S)|$. Consider a new partition P' of V, which is obtained from $P(F)$ by splitting K_S into $K_S \cap (V_S \cup S)$ and $K_S \setminus (V_S \cup S)$. Clearly, $N_{P(F)} - N_{P'} \geq (m/2 + 3)x - 3x = xm/2$. But F is an optimum cluster editing set. Therefore, $x = 0$ and $K_S \subseteq V_S \cup S$. To see that $K_S \supseteq V_S$, suppose to the contrary that there exists some index $1 \leq i \leq m$ such that $v_i(S) \notin K_S$. Let K' be the clique in G' which contains $v_i(S)$. Let P'' be a new partition of V, which is obtained from $P(F)$ by moving $v_i(S)$ from K' to K_S. Then $N_{P(F)} - N_{P''} \geq m/2 + 3 - (m/2 - 3 + 3) = 3$, a contradiction. We conclude that for every $S \in C$ there is a unique clique in G' which contains V_S and is contained in $V_S \cup S$.

Examine an element $u \in U$ which is a member of (at least) two subsets $S_1, S_2 \in C$. By the previous claim, V_{S_1} and V_{S_2} are subsets of distinct cliques in G'. Hence, either $E_{V_{S_1},\{u\}} \subseteq F$, or $E_{V_{S_2},\{u\}} \subseteq F$ (or both). Let $F_1 = F \cap E_1$. Then $|F_1| \geq N$, with equality iff each vertex $u \in U$ is adjacent in G' to the S-vertices of exactly one subset S and $u \in S$. Moreover, since $|F_1| \leq N + M$ and $M \leq 6n$, each vertex $u \in U$ must be adjacent in G' to at least $m - 6n \geq 24n$ S-vertices of exactly one subset $S \in C$ which contains u. Call this set a *majority set* of u.

Let $F_2 = F \setminus F_1$. For every two vertices $u, u' \in U$ such that $(u, u') \in E$, and the majority sets of u and u' differ, we must have $(u, u') \in F_2$. Since each subset in C contains 3 elements, $|N_{G'}(u) \cap U| \leq 2$ for every $u \in U$. Therefore, $|F_2| \geq M$, with equality iff there is a partition of U into triplets of elements, such that the majority set of the elements in each triplet is the same. Since $|F| \leq N + M$, we must have $|F| = N + M$, and the implied partition into triplets induces an exact cover of U. ■

We note, that the same reduction can be used to show that Cluster Deletion is NP-complete.

3.1 p-Cluster Editing

In this section we study the p-Cluster Editing problem. We first show that 2-Cluster Editing is NP-complete. We then conclude that p-Cluster Editing is NP-complete for every $p \geq 2$.

To prove the hardness of 2-Cluster Editing, we define the following problem:

Problem 4 (3-Uniform Hypergraph Balanced 2-Colorability) *Given a 3-uniform hypergraph G, determine if there exists a 2-coloring of G, such that the number of vertices that are colored by each color is the same.*

This problem can be shown to be NP-complete by a trivial reduction from Hypergraph 2-Colorability on 3-uniform hypergraphs, whose NP-hardness was shown by Lovasz [11].

Theorem 2 *2-Cluster Editing is NP-complete.*

Proof: Membership in NP is trivial. We reduce from 3-Uniform Hypergraph Balanced 2-Colorability. Given a hypergraph $G = (V, E)$, we build an instance of 2-Cluster Editing $< G' = (V', E'), k >$ as follows: Let n and m be the number of vertices and hyperedges, respectively, in G. Let $M \equiv 2n^3$. We define $V' = \cup_{i=1}^{n} V_i$ where $V_i = \{v_{i,j} : j = 1, \ldots, M\}$. For a triplet of indices $1 \leq i < j < l \leq n$ define the set $E_{i,j,l} = \{(v_{i,r}, v_{j,r}), (v_{j,r+1}, v_{l,r}), (v_{l,r+1}, v_{i,r+1})\}$, where $r = 2(n^2 i + nj + l) - 1$. We add edges to G' by building a clique around each V_i, and for every triplet of indices $i < j < l$ such that $(i, j, l) \notin E$, we add the edges of $E_{i,j,l}$. Finally, we set $k \equiv 2\binom{n/2}{2}(M^2 - (n-2)) + (\frac{n}{2})^2(n-2) - m$.

The sets $V_1, \ldots, V_n$ will be called *clusters*. We say that a partition $(S, V' \setminus S)$ *splits* a cluster V_i if $V_i \cap S \neq \phi$ and $V_i \not\subseteq S$. For convenience we also define a graph G'' which is built like G' except that it contains the edges $E_{i,j,l}$ for every triplet $i < j < l$. We now prove the correctness of this reduction, namely that there is a balanced 2-coloring of G iff there is a 2-cluster editing set of G' of size at most k.

⇒ Suppose that $f : V \to \{0, 1\}$ is a balanced 2-coloring of G. Let $S = \cup_{i:f(i)=0} V_i$, and let F', F'' be the 2-cluster editing sets of G' and G'', respectively, that correspond to the partition $(S, V \setminus S)$. Since f is balanced, each side of the partition $(S, V \setminus S)$ consists of $n/2$ clusters. We first compute the size of F''. For two clusters

V_i and V_j (with $i < j$), and for each $l \neq i, j$, one of $E_{i,j,l}$, $E_{i,l,j}$, or $E_{l,i,j}$ contains exactly one edge between V_i and V_j. Therefore, between each pair of clusters in G'' there are exactly $n-2$ edges. It follows that F'' consists of $2\binom{n/2}{2}(M^2-(n-2))$ edges that are not in E between clusters in the same set in the partition $(S, V \setminus S)$, and $(\frac{n}{2})^2(n-2)$ edges in E between clusters in different sets in the partition. Thus, $|F''| = 2\binom{n/2}{2}(M^2-(n-2)) + (\frac{n}{2})^2(n-2)$. We now compute the size of F'. For each edge $(i,j,l) \in E$, the edges of $E_{i,j,l}$ in G'' contribute two edges to F'' (as the clusters V_i, V_j, and V_k are not all in the same set in the partition), while the nonexistence of the edges of $E_{i,j,l}$ in G' contributes only one edge to F' (between the two clusters on the same side). It follows that $|F'| = |F''| - m = k$.

$\Leftarrow$ Suppose that F is a 2-cluster editing set of G' of minimum size, and $|F| \leq k$. Let $P(F) = (S, V' \setminus S)$. We first claim that $(S, V' \setminus S)$ splits no cluster. Suppose conversely that $(S, V' \setminus S)$ splits at least one cluster.

If $(S, V' \setminus S)$ splits more than one cluster, then let V_i be a cluster that is split by the partition such that $|V_i \cap S|$ is minimum, and let V_j be a cluster that is split by the partition such that $|V_j \cap S|$ is maximum and $j \neq i$. Denote $a = |V_i \cap S|$ and $b = |V_j \cap S|$. Select some vertex $u \in V_i \cap S$ and a vertex $w \in V_j \setminus S$. Let $S' = S \cup \{w\} \setminus \{u\}$, and let F' be the 2-cluster editing set that corresponds to $(S', V' \setminus S')$. We will show that $|F| - |F'| \geq 0$. Note that a vertex $v \in V_i$ has at most one neighbor outside V_i. If such a neighbor exists, denote it by n_v. The number of edges in F that are incident on u is at least $(M-a) + (|S| - a - 1)$ (the term -1 is due to the possibility that n_u exists and $n_u \in S$) and the number of edges in F that are incident on w is is at least $b + (nM - |S| - (M-b) - 1)$ (the term -1 is due to the possibility that n_w exists and $n_w \in V' \setminus S$). The total number of edges of these two kinds is $nM - 2a + 2b - 2$. Similarly, the number of edges in F' that are incident on u or w is at most $a + (nM - |S| - (M-a) - 1) + (M-b) + (|S| - b - 1) = nM + 2a - 2b - 2$. It follows that

$$|F| - |F'| \geq (nM - 2a + 2b - 2) - (nM + 2a - 2b - 2) = 4(b-a) \geq 0.$$

If $a < b$, we have that $|F'| < |F|$, a contradiction to the minimality of F. In the case when $a = b$, namely the value of $|V_l \cap S|$ is equal amongst all the clusters, we have that $|F'| = |F|$. We build a set S'' from S' using the same process as above, and since $|V_l \cap S'|$ is not equal amongst the clusters, it follows that the 2-cluster editing set F'' that corresponds to S'' satisfies $|F''| < |F'| = |F|$, and again we have a contradiction.

Now suppose that the partition $(S, V' \setminus S)$ splits exactly one cluster, and denote this cluster by V_i. Let $a = |V_i \cap S|$. Out of the rest $n-1$ clusters, suppose that r clusters are contained in S, and $n-r-1$ clusters are contained in $V' \setminus S$. W.l.o.g. suppose that $n-r-1 \leq r$, and since n is even we have $n-r-1 \leq r-1$. Define $S' = S \setminus V_i$, and let F' be the corresponding 2-cluster editing set. For each $v \in V_i \cap S$, there are at least $rM-1$ edges in F between v and $S \setminus V_i$, and $M-a$ edges between v and $V_i \setminus S$, so the number of edges in F that are incident on v is at least $rM - 1 + M - a$. On the other hand, an edge in F' that is incident on v is either between v and n_v, or between v and $(V' \setminus S) \setminus V_i$. The number of edges of the latter type is $(n-1-r)M$, so the number of edges in F that are incident on v is at most $(n-1-r)M + 1 \leq (r-1)M + 1$. It follows that

$$|F| - |F'| \geq a\,(rM - 1 + M - a - ((r-1)M + 1)) = a\,(2M - a - 2) > 0,$$

contradicting the minimality of F. Therefore F splits no cluster.

We now claim that the number of clusters that are contained in S is exactly $n/2$. Conversely, suppose w.l.o.g. that $r > n/2$. Let V_i be some cluster contained in S. Let $S' = S \setminus V_i$ and let F' be the corresponding 2-cluster editing set. Similarly to the above, we have that

$$|F| - |F'| \geq M((r-1)M - 1 - ((n-r)M + 1)) \geq M(M-2) > 0,$$

a contradiction. Hence, S contain $n/2$ clusters.

Define a coloring $f : V \to \{0,1\}$ by $f(i) = 0$ iff $V_i \subseteq S$. By the argument above, f is balanced. It remains to show that f is a legal 2-coloring. For a hyperedge $(i,j,k) \in E$, if i,j,k have the same color then $|F \cap E_{i,j,l}| = 3$. Otherwise, $|F \cap E_{i,j,l}| = 1$ since two of the edges in $E_{i,j,l}$ must connect vertices in clusters on different sides of the partition $(S, V' \setminus S)$. Hence, each monochromatic hyperedge adds two to the size of F. By the first direction of the proof, for a legal 2-coloring, the corresponding editing set is of size exactly k, and thus no monochromatic hyperedge is possible in f. It follows that f is a balanced 2-coloring of G. ■

Corollary 1 *p-Cluster Editing is NP-complete for any $p \geq 2$.*

Proof: Fix $p > 2$. We provide a reduction from 2-Cluster Editing. Given an input instance $< G = (V, E), k >$ of 2-Cluster Editing, $|V| = n$, we form an instance $< G' = (V', E'), k >$ of p-Cluster Editing as follows: Define $V' = V \cup \cup_{i=1}^{p-2} V_i$ where $V_i = \{w_{i,j} : j = 1, \ldots, n^2\}$. The edges of G' include all the edges in E and a clique on each V_i.

Clearly, every 2-cluster editing set of G is a p-cluster editing set of G (of the same size). Conversely, suppose that F' is a p-cluster editing set of G' of size at most k, and let $(S_1, \ldots, S_p)$ be the corresponding partition. We show that F' is also a 2-cluster editing set for G.

If there exists a set V_i such that $V_i \cap S_j \neq \phi$ and $V_i \not\subseteq S_j$ for some j, then F' contains $E_{V_i \cap S_j, V_i \setminus S_j}$. The number of such edges is at least $n^2 - 1 > k$, a contradiction. Therefore, every set V_i is contained in some set S_j. Furthermore, every set S_j contains at most one set V_i since otherwise we have $|F'| \geq n^4 > k$, a contradiction. It follows that all edges in F' are incident on vertices of V, which implies that F' is a 2-cluster editing set of G. ■

3.2 A 0.878-Approximation Algorithm

We give in this section a polynomial approximation algorithm for a weighted variant of 2-Cluster Editing which is defined as follows:

Problem 5 (Weighted 2-Cluster Editing) *Given a graph G and a weight function on vertex pairs $w : E(G) \cup E(\overline{G}) \rightarrow \mathcal{N}$, find in G a 2-cluster editing set of maximum total weight of unedited vertex pairs.*

Note, that the decision version of Weighted 2-Cluster Editing reduces to that of 2-Cluster Editing when $w \equiv 1$ (i.e., $w(e) = 1$ for every $e \in E(G) \cup E(\overline{G})$).

Let $n = |V|$ and let S_n denote the n-dimensional unit sphere. We define the following semi-definite relaxation of Weighted 2-Cluster Editing:

$$\begin{array}{ll} \max & \frac{1}{2}[\sum_{(i,j)\in E} (w_{ij}(1 + v_i \cdot v_j)) + \sum_{(i,j)\notin E} (w_{ij}(1 - v_i \cdot v_j))] \\ \text{s.t.} & v_i \in S_n \ \ \forall i \end{array}$$

We claim that this is indeed a relaxation of Weighted 2-Cluster Editing, that is, for every partition $P = (A, B)$ of G there exist vectors $v_1, \ldots, v_n \in S_n$ such that the total weight of unedited vertex pairs as implied by P is $\frac{1}{2}[\sum_{(i,j)\in E}(w_{ij}(1 + v_i \cdot v_j)) + \sum_{(i,j)\notin E}(w_{ij}(1 - v_i \cdot v_j))]$. Indeed, let (A, B) be a partition of G. Let v_0 be any unit vector in S_n. For every $i \in A$ set $v_i = v_0$, and for every $i \in B$ set $v_i = -v_0$. The claim follows.

Our approximation algorithm solves this semi-definite relaxation and then rounds the solution obtained using the random hyperplane technique.

Theorem 3 *The algorithm approximates Weighted 2-Cluster Editing with an expected approximation ratio of at least* 0.878.

Proof: Follows directly from [5, Theorem 6.1]. ■

4 Cluster Completion

The Cluster Completion problem is trivially polynomial: The optimum solution is obtained by simply transforming each connected component of the input graph into a complete graph. In this section we give a polynomial algorithm to p-Cluster Completion for any fixed $p \geq 2$.

Let $G = (V, E)$ be an input graph with n vertices and t connected components. If $t < p$ we output *False*. We assume henceforth that $t \geq p$. To find the optimum completion set we compute partitions of the t components of G

```
$S_0 = \{(0,\ldots,0)\}$
For $i = 1$ to $t$ do:
    $S_i = S_{i-1} \cup \{(v + C_i e_j : v \in S_{i-1}, j = 1,\ldots,p-1\}$
Pick in $S_t$ a vector $v^*$ minimizing $\sum_{i=1}^{p-1} v_i^2 + (n - \sum_{i=1}^{p-1} v_i)^2$.
```

Figure 1: An algorithm for p-Cluster Completion. e_j denotes a $(p-1)$-dimensional unit vector with 1 in position j.

into p sets (splitting no connected components) and choose the partition which results in a minimum completion set. Using dynamic programming, we only need to consider a polynomial number of partitions. Note that since we only add edges, we seek to minimize the sum of the number of edges in each of the p sets of the partition, or equivalently, the sum of the squared sizes of the sets.

Let $C_1,\ldots,C_t$ be the cardinalities of the connected components in G. Our algorithm will denote each possible partition by a $(p-1)$-long vector of integers which describes the sizes of the sets in the partition (the size of the last set is the difference from n). We will maintain a set S_i of the vectors that correspond to all possible partitions of the first i connected components. The algorithm is given in Figure 1. The actual partition can be obtained by maintaining for each $v \in S_i$ a pointer to its parent vector in S_{i-1}.

Theorem 4 *The algorithm correctly solves the p-Cluster Completion problem in $O(tn^{p-1})$ time.*

5 Cluster Deletion

We now focus on the cluster deletion problem. We shall give a gap preserving reduction (cf. [10]) from a restricted version of SET-COVER to Cluster Deletion. This reduction implies that there exists some constant $\epsilon > 0$ such that it is NP-hard to approximate Cluster Deletion to within a factor of $1+\epsilon$. We begin by introducing the SET-COVER restriction.

Problem 6 (Minimum Restricted Exact Cover (REC)) *The input is a set of elements $U = \{u_1,\ldots,u_t\}$, and a collection C of subsets of U which satisfies the following conditions:*

- *There exists a constant $k_1 > 0$ such that for each $S \in C$, $|S| \le k_1$.*
- *There exists a constant $k_2 > 0$ such that for all $u \in U$, $|\{S \in C : u \in S\}| \le k_2$.*
- *If $S \in C$ and $S' \subset S$ then $S' \in C$.*

The goal is to find a sub-collection $I \subseteq C$ of minimum cardinality, such that $\bigcup_{S \in I} S = U$, and the sets in I are pairwise-disjoint.

Note, that the third condition guarantees that a solution to REC always exists (we assume that $\bigcup_{S \in C} S = U$). REC can be shown to be MAX-SNP complete by a simple L-reduction from a restriction of SET-COVER in which the size of every set is bounded and each element occurs in a bounded number of sets. This latter problem is known to be MAX-SNP complete [14]. Hence, there exists a constant $\delta_{REC} > 0$ such that it is NP-hard to approximate REC to within a factor of $1 + \delta_{REC}$.

Theorem 5 *There exists some constant $\epsilon > 0$ such that it is NP-hard to approximate Cluster Deletion to within a factor of $1+\epsilon$.*

Proof: By a gap preserving reduction from REC (similar to the one in Theorem 1). For an instance I_{REC} of REC, the reduction produces in polynomial time an instance I_{CD} of Cluster Deletion such that $opt(I_{REC}) \le c$ implies $opt(I_{CD}) \le c'$ and $opt(I_{REC}) > (1+\delta_{REC})c$ implies $opt(I_{CD}) > (1+\epsilon)c'$, where $opt(I)$ denotes the optimal value for instance I.

We now describe the reduction. Let $I_{REC} =< U, C >$, and let $|U| = t$. Suppose that each set in C has size at most k_1, and each element occurs in at most k_2 sets. Let $m = \frac{k_1^2 k_2}{\delta_{REC}}$ and let $q = \sum_{S \in C} |S|$. We build an instance $I_{CD} =< G = (V, E) >$ of Cluster Deletion as follows:

$$\begin{aligned}
V &= \bigcup_{S \in C} \{v_1(S), \dots, v_m(S), w(S)\} \cup U \ , \\
E &= E_1 \cup E_2 \cup E_3 \cup E_4 \ , \\
E_1 &= \{(v_i(S), u) : S \in C, 1 \le i \le m, u \in S\} \ , \\
E_2 &= \{(v_i(S), v_j(S)) : S \in C, 1 \le i < j \le m\} \ , \\
E_3 &= \{(u, u') : \exists S \in C \text{ s.t. } u, u' \in S\} \ , \\
E_4 &= \{(v_i(S), w(S)) : S \in C, 1 \le i \le m\} \ .
\end{aligned}$$

In words, for each $S \in C$ we form a clique on S and a set of m new vertices, and also connect all the new vertices to a single extra vertex $w(S)$. For each subset $S \in C$ we denote $V_S = \{v_1(S), \dots, v_m(S)\}$ and call the elements of V_S, S-vertices. Note, that $|E_3| \le (k_1 - 1)k_2 t/2 < k_1 k_2 t/2$ and $q \le k_2 t$. Clearly, $t/k_1 \le opt(I_{REC}) \le t$. Let c be any constant such that $t/k_1 \le c \le t$. Define $c' \equiv (q - t + c)m + |E_3|$ and $\epsilon \equiv \frac{\delta_{REC}}{2k_1k_2 + \delta_{REC}}$. We prove that this reduction is gap preserving:

1. Suppose that $opt(I_{REC}) \le c$. Let $I \subseteq C$ be an exact cover of U, $|I| \le c$. For $u \in U$ denote by I_u the set in I which contains u. Let $\bar{I} = C \setminus I$.

   To obtain a cluster subgraph G' of G we delete the following edges:

   (a) For all $S \in \bar{I}, u \in S$ delete all the edges in $E_{V_S,\{u\}}$.

   (b) For all $S \in I$ delete all the edges in $E_{V_S,\{w(S)\}}$.

   (c) For all $u \in U, u' \in U \setminus I_u$ delete the edge (u, u') if it exists.

   One can easily verify that G' is a cluster graph, and therefore, $opt(I_{CD}) \le (q - t + c)m + |E_3| = c'$.

2. Suppose that $opt(I_{REC}) > (1 + \delta_{REC})c$. We can make the following observations with respect to $opt(I_{CD})$:

   (a) In any cluster subgraph of G, every $u \in U$ is adjacent to the S-vertices of at most one set $S \in C$. Therefore, $opt(I_{CD}) \ge (q - t)m$.

   (b) There exists an optimum solution F of I_{CD} for which the following is true: If a vertex $u \in U$ is adjacent to an S-vertex in $(V, E \setminus F)$, for some $S \in C$, then F contains all the edges in $E_{V_S,\{w(S)\}}$. Indeed, if F' is a cluster deletion set such that $u_1, \dots, u_r$ ($1 \le r \le k_1$) are adjacent to an S-vertex in $(V, E \setminus F')$, then $F'' = (F' \cup E_{V_S,\{w(S)\}}) \setminus (\bigcup_{i=1}^r E_{V_S,\{u_i\}} \cup \{v_i(S), v_j(S) : i \ne j\})$ is also such a cluster deletion set, and $|F''| \le |F'|$. Examine now F. For each $u \in U$, either F contains all edges connecting u to vertices in $V \setminus U$, or there exists a single set $S \in C$ such that $E_{V_S,\{u\}} \cap F = \phi$ and $E_{V_S,\{w(S)\}} \subseteq F$. Hence, F contains at least $(q - t)m$ edges between vertices in U and vertices in $V \setminus U$ (S-vertices), as well as at least $opt(I_{REC})m$ additional edges. It follows that $opt(I_{CD}) \ge (q - t + opt(I_{REC}))m > (q - t + (1 + \delta_{REC})c)m$.

   We conclude that

$$\begin{aligned}
opt(I_{CD}) &> (q - t + (1 + \delta_{REC})c)m = c' + (\delta_{REC} cm - |E_3|) \\
&> c'(1 + \frac{\delta_{REC} cm - |E_3|}{qm + |E_3|}) > c'(1 + \frac{\delta_{REC}(t/k_1)m - k_1k_2t/2}{k_2tm + k_1k_2t/2}) \\
&= c'(1 + \frac{2\delta_{REC} m/k_1 - k_1k_2}{2k_2m + k_1k_2}) = c'(1 + \frac{\delta_{REC}}{2k_1k_2 + \delta_{REC}}) \\
&= c'(1 + \epsilon) \ .
\end{aligned}$$

■

5.1 p-Cluster Deletion

We give in this section a polynomial algorithm for the optimization version of 2-Cluster Deletion. We then show that p-Cluster Deletion is NP-complete for every $p > 2$.

Let $G = (V, E)$ be an input graph with n vertices. W.l.o.g., G is connected, as otherwise, either G is already a 2-cluster graph, or we output *False*. The algorithm is described in Figure 2. Recall that $\overline{G}$ is the complement of G.

```
Let $C_1, \ldots, C_t$ be the connected components of $\overline{G}$.
For $i = 1, \ldots, t$ do:
    If $C_i$ is not bipartite then output False and halt.
    Else find a bipartition $(A_i, B_i)$ of $C_i$ such that $|A_i| \geq |B_i|$.
Output the set that corresponds to $(A_1 \cup \ldots \cup A_t, B_1 \cup \ldots \cup B_t)$.
```

Figure 2: An algorithm for 2-Cluster Deletion.

Theorem 6 *The algorithm correctly solves 2-Cluster Deletion in* $O(n + |E(\overline{G})|)$ *time.*

Proof: **Correctness**: Since the complement of a 2-cluster graph is a complete bipartite graph, a solution exists if and only if $\overline{G}$ is bipartite. Hence, the algorithm outputs *False* iff no solution exists. Moreover, the partition produced by the algorithm has the property that if two vertices are assigned to the same set then they are adjacent. Therefore, the set of edges F returned by the algorithm is a 2-deletion set of G. Hence, it suffices to prove that F is optimal.

Denote $S_1 = A_1 \cup \ldots \cup A_t$ and $S_2 = B_1 \cup \ldots \cup B_t$. Clearly, F consists of edges in G with one endpoint in S_1 and the other in S_2. Therefore,

$$|F| = |E_{S_1,S_2}| = |S_1||S_2| - E(\overline{G}) = |S_1|(n - |S_1|) - E(\overline{G}).$$

Let F^* be an optimal 2-deletion set of G, and let $P(F^*) = (S_1^*, S_2^*)$, where $|S_1^*| \leq |S_2^*|$. We have that $|F^*| = |S_1^*|(n - |S_1^*|) - E(\overline{G})$. For every $i \leq t$, either $A_i \subseteq S_1^*$ or $B_i \subseteq S_1^*$, and therefore $|S_1| \leq |S_1^*| \leq n/2$. It follows that $|F| \leq |F^*|$. Hence, F is an optimal 2-deletion set of G.

Complexity: The bottleneck in the complexity of the algorithm is computing the connected components of $\overline{G}$ and finding a bipartition for each of them. Each of these tasks can be performed in $O(n + |E(\overline{G})|)$ time. ■

Theorem 7 *p-Cluster Deletion is NP-complete for any* $p \geq 3$.

Proof: Membership in NP is trivial. We provide a reduction from p-Coloring. Given an input graph $G = (V, E)$, the reduction outputs its complement $\overline{G}$ and a bound $k = |\overline{E}|$. A p-coloring f of G trivially translates into a p-deletion set $\{(u, v) \notin E : f(u) \neq f(v)\}$ of $\overline{G}$ of size at most k. Conversely, suppose that F is a p-deletion set of $\overline{G}$ with $|F| \leq k$, and let $C_1, \ldots, C_p$ be the cliques of $(V, \overline{E} \setminus F)$. The coloring f defined by $f(v) = i$ for all $v \in C_i$ is a p-coloring of G. ■

Note that the reduction works with any $k \geq |\overline{E}|$ and in fact shows that even deciding whether a graph has a p-cluster deletion set is NP-hard, for $p \geq 3$.

Acknowledgments

R. Shamir was supported in part by the Israel Science Foundation formed by the Israel Academy of Sciences and Humanities. R. Sharan was supported by an Eshkol fellowship from the Ministry of Science, Israel.

References

[1] A. A. Alizadeh, M. B. Eisen, et al. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[2] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3/4):281–297, 1999.

[3] C. Berge. *Graphs and Hypergraphs*. North-Holland, Amsterdam, 1973.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., San Francisco, 1979.

[5] M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995.

[6] T. R. Golub, D. K. Slonim, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, October 1999.

[7] C. Hagen and A.B. Kahng. New spectral methods for ratio cut partitioning and clustering. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(9):1074–1085, 1992.

[8] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79:191–215, 1997.

[9] J.A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, 1975.

[10] D. S. Hochbaum, editor. *Approximation Alogrithms for NP-Hard Problems*. PWS Publishing, Boston, 1997.

[11] L. Lovasz. Covering and coloring of hypergraphs. In *Proc. 4th Southeastern Conf. on Combinatorics, Graph Theory, and Computing*. Utilitas Mathematica Publishing, 1973.

[12] A. Natanzon. Complexity and approximation of some graph modification problems. Master's thesis, Department of Computer Science, Tel Aviv University, 1999.

[13] A. Natanzon, R. Shamir, and R. Sharan. Complexity classification of some edge modification problems. *Discrete Applied Mathematics*, 113:109–128, 2001.

[14] C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. of Computer and System Science*, 43:425–440, 1991.

[15] R. Sharan and R. Shamir. CLICK: A clustering algorithm with applications to gene expression analysis. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pages 307–316, 2000.

[16] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.