Subexponential Algorithm for d-Cluster Edge Deletion: Exception or Rule?

Neeldhara Misra[1], Fahad Panolan[2], and Saket Saurabh[2]

[1] Indian Institute of Science, Bangalore, India
neeldhara@csa.iisc.ernet.in
[2] The Institute of Mathematical Sciences, Chennai, India
{fahad, saket}@imsc.res.in

Abstract. The correlation clustering problem is a fundamental problem in both theory and practice, and it involves identifying clusters of objects in a data set based on their similarity. A traditional modeling of this question as a graph theoretic problem involves associating vertices with data points and indicating similarity by adjacency. Clusters then correspond to cliques in the graph. The resulting optimization problem, Cluster Editing (and several variants) are very well-studied algorithmically. In many situations, however, translating clusters to cliques can be somewhat restrictive. A more flexible notion would be that of a structure where the vertices are mutually "not too far apart", without necessarily being adjacent.
One such generalization is realized by structures called *s-clubs*, which are graphs of diameter at most s. In this work, we study the question of finding a set of at most k edges whose removal leaves us with a graph whose components are s-clubs. Recently, it has been shown that unless Exponential Time Hypothesis fail (ETH) fails Cluster Editing (whose components are 1-clubs) does not admit sub-exponential time algorithm [*STACS, 2013*]. That is, there is no algorithm solving the problem in time $2^{o(k)}n^{O(1)}$. However, surprisingly they show that when the number of cliques in the output graph is restricted to d, then the problem can be solved in time $O(2^{O(\sqrt{dk})} + m + n)$. We show that this sub-exponential time algorithm for the fixed number of cliques is rather an exception than a rule. Our first result shows that assuming the ETH, there is no algorithm solving the s-Club Cluster Edge Deletion problem in time $2^{o(k)}n^{O(1)}$. We show, further, that even the problem of deleting edges to obtain a graph with d s-clubs cannot be solved in time $2^{o(k)}n^{O(1)}$ for any fixed s, $d \geq 2$. This is a radical contrast from the situation established for cliques, where sub-exponential algorithms are known.

Keywords: subexponential algorithms, s-clubs, cluster edge deletion, ETH-hardness

1 Introduction

The correlation clustering problem involves identifying clusters of objects in a data set based on their similarity. A traditional way of posing this as a graph theoretic question involves associating vertices with data points and indicating similarity by adjacency. In this setting, the natural notion of a cluster would correspond to a *clique*, a set of mutually adjacent vertices. Thus, we call a graph G a *cluster graph* if every connected component of G is a complete graph. The task of identifying clusters can now be viewed as an optimization problem. In particular, a subset $F \subseteq E$ is called a cluster edge deletion set if $G \backslash F = (V, E \backslash F)$ is a cluster. On the other hand, if for some $F \subset V \times V$, $G \Delta F = (V, E \Delta F)$ is a cluster, then F is called cluster editing set. (Here $E \Delta F$ is the symmetric difference between E and F.) In the Cluster Edge Deletion (Cluster Editing) problem, we are given a graph G and an integer k, and we want to check whether there exists a cluster edge deletion set (cluster editing set), F of size at most k.

The complexity of Cluster Edge Deletion and Cluster Editing is well-understood. The problems are NP-complete and admit constant-factor approximation algorithms. On the other hand, they are also known to be APX-hard. Further, it has been recently shown that Cluster Editing cannot be solved in time $2^{o(k)} n^{O(1)}$ unless the Exponential Time Hypothesis (ETH) fails. This led the authors of [3] to consider the question of editing at most k edges to obtain a graph with at most d clusters. This variant continues to be well motivated in several practical settings, where the number of clusters corresponds to an external constraint. With the restriction on the number of clusters in place, there is good news, as [3] describes an algorithm that solves the problem in time $O(2^{O(\sqrt{dk})} + m + n)$.

So far, we have considered the clustering problem in the graph theoretic context using cliques as a natural means for modeling the notion of a cluster. This effectively restricts us to a binary notion of similarity, in that a pair of data points are either similar or not, and we would like to maximize similarities within a cluster and minimize non-similarities across clusters. In many situations, however, this translation can be somewhat severe. A more flexible notion would be that of a structure where the vertices are mutually "not too far apart", without necessarily being adjacent. Additionally, note that cliques are also a popular choice for modeling highly correlated or connected substructures in applications. Given that cliques impose a very strict connectivity requirement, this modeling suffers from being overly restrictive.

A natural generalization of the notion of cliques would be along the lines of small-diameter graphs. These structures are called *clubs* and have been proposed as a more reasonable measure of connectivity and correlation. Formally, note that the complete graphs can be thought of as graphs of diameter one. A s-club is a graph of diameter at most s, and note that cliques are exactly 1-clubs. The notion of s-clubs was introduced in [1]. The s-club concept was defined in the context of social sciences [1], and it has recently been used in the analysis of social [10] and biological networks. In [5,6,11] parameterized studies of finding s-clubs were undertaken. It is worth to mention that several other generalizations of cliques such as s-cliques and s-plexes [4] and the related notion of clustering into these graphs have been studied in literature before.

The immediate question that arises in the context of clustering is the s-Club Cluster Edge Deletion problem: is there a set of at most k edges whose removal leaves us with a graph whose components are s-clubs? It is known that the problem is NP-complete for $s = 2$, and there is an algorithm that solves the problem in $O(2.74^k n^{O(1)})$ [7]. It is natural to ask if the problem admits a sub-exponential algorithm. Our first result shows that assuming the ETH, the answer is in the negative:

Theorem 1. 2-Club Cluster Edge Deletion *cannot be solved in time* $2^{o(k)} n^{O(1)}$*, unless ETH fails.*

In the setting of cliques, it was useful to consider the question with the additional dimension of the number of clusters: if we demanded deletion into at most d clusters, then the problem turned out to admit a sub-exponential algorithm. It is therefore natural to consider the corresponding question in the s-club setting: can we identify at most k edges whose removal leaves us with at most d s-clubs? It turns out that the slightest generalization of the cluster editing problem makes the problem significantly harder in the context of sub-exponential algorithms. In particular, we show:

Theorem 2. s-Club d-Cluster Edge Deletion *for* $s \geq 2$ *and* $d \geq 2$ *cannot be solved in time* $2^{o(k)} n^{O(1)}$*, unless ETH fails.*

Our Theorem 2 shows that the sub-exponential algorithm in the case of 1-Club d-Cluster Edge Deletion is rather an exception. All our results are obtained by reductions from 3-CNFSAT. The Exponential Time Hypothesis states that there is no algorithm that solves 3-CNFSAT in time $2^{o(m+n)}$ time (via sparsification). Our reductions produce instances where the size of the solution depends linearly on $(m + n)$. We refer to recent survey of Lokshtanov et al. [8] for a detailed discussions on ETH and to the books [2,9] for an introduction to the area of parameterized complexity.

Organization of the paper. In Section 2 we establish the notation and state the problems formally. In Sections 3 and 4, we prove Theorems 1 and 2, respectively. The proof of Theorem 2 is split into three cases, namely $s = 2$, $s = 3$, and $s \geq 4$.

2 Preliminaries

Graphs. For a finite set V, a pair $G = (V, E)$ such that $E \subseteq V^2$ is a graph on V. The elements of V are called *vertices*, while pairs of vertices (u, v) such that $(u, v) \in E$ are called *edges*. In the following, let $G = (V, E)$ and $G' = (V', E')$ be graphs, and $U \subseteq V$ some subset of vertices of G. Let G' be a subgraph of G. If E' contains all the edges $\{u, v\} \in E$ with $u, v \in V'$, then G' is an *induced subgraph* of G, *induced by* V'. For any set of vertices $U \subseteq V$, $G[U]$ denotes the subgraph of G induced by U. For $v \in V$, $N(v) = \{u \mid (u, v) \in E\}$ and $N[v] = N(v) \cup \{v\}$. For $U \subseteq V$, $N(U) = \left(\bigcup_{u \in U} N(u)\right) \setminus U$.

The *distance* between vertices u, v of G is the length of a shortest path from u to v in G; if no such path exists, the distance is defined to be ∞. The *diameter* of G is the greatest distance between any two vertices in G. A graph G is said to be *connected* if there is a path in G from every vertex of G to every other vertex of G. If $U \subseteq V$ and $G[U]$ is connected, then U itself is said to be connected in G. A subset of vertices U is said to induce a s-club if $G[U]$ has diameter at most s, or in other words, the distance between every pair of vertices in U is at most s in $G[U]$. A graph is said to be a s-club cluster if every connected component of the graph induces a s-club.

Satisfiability. Let P be an arbitrary set, whose elements we shall refer to as *variables*. It will be convenient to assume that P is a countably infinite set. The set of formulas over P is inductively defined to be the smallest set of expressions such that: (a) Each variable in the set P is a formula, (b) $(\neg\alpha)$ is a formula whenever α is, and (c) $(\alpha \,\Box\, \beta)$ is a formula whenever α and β are formulas and $\Box$ is one of the binary connectives $\wedge$, $\vee$.

We denote by $\mathcal{F}(P)$ the set of all formulas over P. An valuation or an assignment of P is a function $v : P \to \{0, 1\}$, which may be extended to a function $\bar{v} : \mathcal{F}(P) \to \{0, 1\}$, as follows. For each variable x in the set P, $\bar{v}(x) = v(x)$. Further, $\bar{v}(\neg\alpha) = 1 - \bar{v}(\alpha)$, $\bar{v}(\alpha \;\wedge \beta) = min\{\bar{v}(\alpha), \bar{v}(\beta)\}$, and $\bar{v}(\alpha \;\vee \beta) = max\{\bar{v}(\alpha), \bar{v}(\beta)\}$.

A formula is in *conjunctive normal form* (CNF) if it is a conjunction of clauses, where a clause is a disjunction of literals. Every propositional formula can be converted into an equivalent formula that is in CNF. The question of *satisfiability* is whether, given a formula α, there exists a valuation v such that $v(\alpha) = 1$. This is one of the most well-studied NP-complete problems. The problem continues to be NP-complete if the formula is offered in CNF even when every clause has no more than three variables.

Notice that given a 3-SAT formula, we may preprocess it effectively to ensure that each variable appears at least twice: at least once in a positive literal and at least once in a negative one. This is because any variable that appears only positively (respectively, negatively) can be assigned 1 (respectively, 0) by a satisfying assignment without loss of generality. Similarly, we assume that any variable will not appear both positively and negatively in a clause, because such a clause can be removed from the formula without affecting the satisfiability of the formula. Finally, we may assume that each clause of ϕ consists of exactly three literals by a standard padding argument using dummy variables. We say that a 3-CNF formula is standardized if it satisfies all three properties above. In our discussions, we work with standardized formulas.

The problems we study in this work are the following:

s-Club Cluster Edge Deletion

Instance: An undirected graph $G = (V, E)$ and a positive integer k.
Problem: Does there exist $E' \subseteq E$ with $|E'| \leq k$ such that $G \setminus E'$ is an s-club cluster?

s-Club d-Cluster Edge Deletion

Instance: An undirected graph $G = (V, E)$ and a positive integer k.
Problem: Does there exist $E' \subseteq E$ with $|E'| \leq k$ such that $G \setminus E'$ is an s-club cluster containing d components?

3 2-Club Cluster Edge Deletion

In this section we will show that 2-Club Cluster Edge Deletion cannot be solved in $2^{o(k)}n^{O(1)}$ unless ETH fails. To show this result we will give a reduction from 3-Sat to 2-Club Cluster Edge Deletion. More precisely, from an instance ϕ with m clauses and n variables, of 3-Sat, we will construct an instance (G, k) of 2-Club Cluster Edge Deletion with the property that ϕ is satisfiable iff (G, k) is an Yes instance, where $k = O(m + n)$.

Lemma 1. ($\star$)[3] *Let $G = (V, E)$ be an undirected graph. Let $X \subseteq V$ such that $G[X]$ is a clique, $\forall x, y \in X, N[x] = N[y]$ and $G[N(X)]$ is a clique. Then there exist an optimum solution F to* 2-Club Cluster Edge Deletion *such that X is contained in a single component in $G \setminus F$.*

Lemma 2. ($\star$) *There exists a polynomial-time algorithm that, given a 3-CNF formula ϕ with n variables and m clauses, constructs a* 2-Club Cluster Edge Deletion *instance (G, k) such that (i) ϕ is satisfiable if and only if (G, k) is a* Yes-instance, and *(ii) $k = O(n + m)$.*

Theorem 3. 2-Club Cluster Edge Deletion *cannot be solved in time $2^{o(k)}n^{O(1)}$, unless ETH fails.*

4 s-Club d-Cluster Edge Deletion

In this section, we show the hardness of s-Club d-Cluster Edge Deletion for all $s \geq 2$. The results are divided into three parts. First, we demonstrate a reduction from 3-Sat to 2-Club 2-Cluster Edge Deletion. With minor modifications, we show that this reduction works for the problem of edge deletion into two 3-clubs. For $s \geq 4$, we show a general reduction from 3-Sat to s-Club 2-Cluster Edge Deletion. The construction in the first reduction serves as a basis for the general reduction, but we note that the finer details involve several nuances. We also note that the problem of deleting into two s-clubs easily reduces to the problem of deleting into d s-clubs.

[3] Due to space restrictions, the proofs of Lemmas marked with a $\star$ are moved to the appendix.

4.1 2-Club 2-Cluster Edge Deletion

In this section we will show that 2-Club 2-Cluster Edge Deletion cannot be solved in $2^{o(k)}n^{O(1)}$ unless ETH fails. To this end, we will give a reduction from 3-Sat to 2-Club 2-Cluster Edge Deletion. More precisely, based on an instance ϕ of 3-Sat with m clauses and n variables, we will construct an instance (G, k) of 2-Club 2-Cluster Edge Deletion with the property that ϕ is satisfiable if and only if (G, k) is an Yes instance, where $k = O(m + n)$.

Lemma 3. *There exists a polynomial-time algorithm that, given a 3-CNF formula ϕ with n variables and m clauses, constructs a* 2-Club 2-Cluster Edge Deletion *instance (G, k) such that (i) ϕ is satisfiable iff (G, k) is a* Yes *instance, and (ii) $k = O(m + n)$.*

Proof. Let ϕ be a standardized 3-CNF formula with m clauses and n variables. Let $C_1, C_2, \ldots, C_m$ be the clauses and $x_1, x_2, \ldots, x_n$ be the variables.

Construction. We construct a graph $G = (V, E)$ based on ϕ as follows. The graph G contains two clause gadgets $\mathcal{C}_1$ and $\mathcal{C}_2$, two connection gadgets K_1 and K_2, two selection gadgets S_1 and S_2, one variable gadget $\mathcal{V}$ and four global vertices $\{p, p', g_1, g_2\}$. The clause gadget $\mathcal{C}_1$ contains m vertices $c_1, c_2, \ldots, c_m$ and there are no edges within $\mathcal{C}_1$. Similarly, $\mathcal{C}_2$ contains m vertices $c'_1, c'_2, \ldots, c'_m$ and there are no edges within $\mathcal{C}_2$. The variable gadget $\mathcal{V}$ contains $2n$ vertices, one for each literal. Let these vertices be named $x_1, x_2, \ldots, x_n$ and $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}$. The connection gadgets K_1 and K_2 are cliques of size $k + 2$. The selection gadget S_1 contains n vertices $a_1, a_2, \ldots, a_n$ and no edges within S_1. Similarly, S_2 contains n vertices $b_1, b_2, \ldots, b_n$ and no edges within S_2.

For each $1 \leq i, j \leq m$ we add an edge (c_i, c'_j) if $i \neq j$. For any literal $l = x_i$ or $l = \overline{x_i}$, and for every clause C_i that contains l, we add the edges (c_i, l) and (c'_i, l). For each $1 \leq i, j \leq n$ we add an edge (a_i, b_j) if $i \neq j$. For every pair of literals x_i and $\overline{x_i}$, add the edges (x_i, a_i), (x_i, b_i), $(\overline{x_i}, a_i)$ and $(\overline{x_i}, b_i)$. Also, add all possible edges between: K_1 and g_2; K_2 and g_2; K_1 and S_1; K_2 and S_2; g_1 and $\mathcal{V}$; g_2 and $\mathcal{V}$; p and $\mathcal{C}_1$; p' and $\mathcal{C}_2$. Finally, add the edges (g_1, p) and (g_1, p'). (See Fig. 1.) We set $k = 4(m + n)$.

Completeness. Let ϕ be satisfiable, and $f : \{x_1, \ldots, x_n\} \to \{0, 1\}$ be a satsfying assignment. Now we construct the edge deletion set $F \subseteq E(G)$ as follows. For each $1 \leq i \leq n$, if $f(x_i) = 1$, then include in F the edges between x_i and S_b $(b = 1, 2)$, the edges between $\overline{x_i}$ and $\mathcal{C}_b$ (for $b = 1, 2$), and the edges $(\overline{x_i}, g_1), (x_i, g_2)$. On the other hand, if $f(x_i) = 0$, the we include in F the edges between $\overline{x_i}$ and S_b $(b = 1, 2)$, edges between x_i and $\mathcal{C}_b$ (for $b = 1, 2$), and the edges $(\overline{x_i}, g_2), (x_i, g_1)$.

Note that the number of edges in F which are between $\mathcal{C}_b$ (for $b = 1, 2$) and $\mathcal{V}$ is at most $4m$. This is because every vertex $c \in \mathcal{C}_1 \cup \mathcal{C}_2$ has three neighbors in $\mathcal{V}$, of which F picks at most two (since the choice of F is based on a satisfying assignment f). The number of edges in F which are between S_i (for $i = 1, 2$)

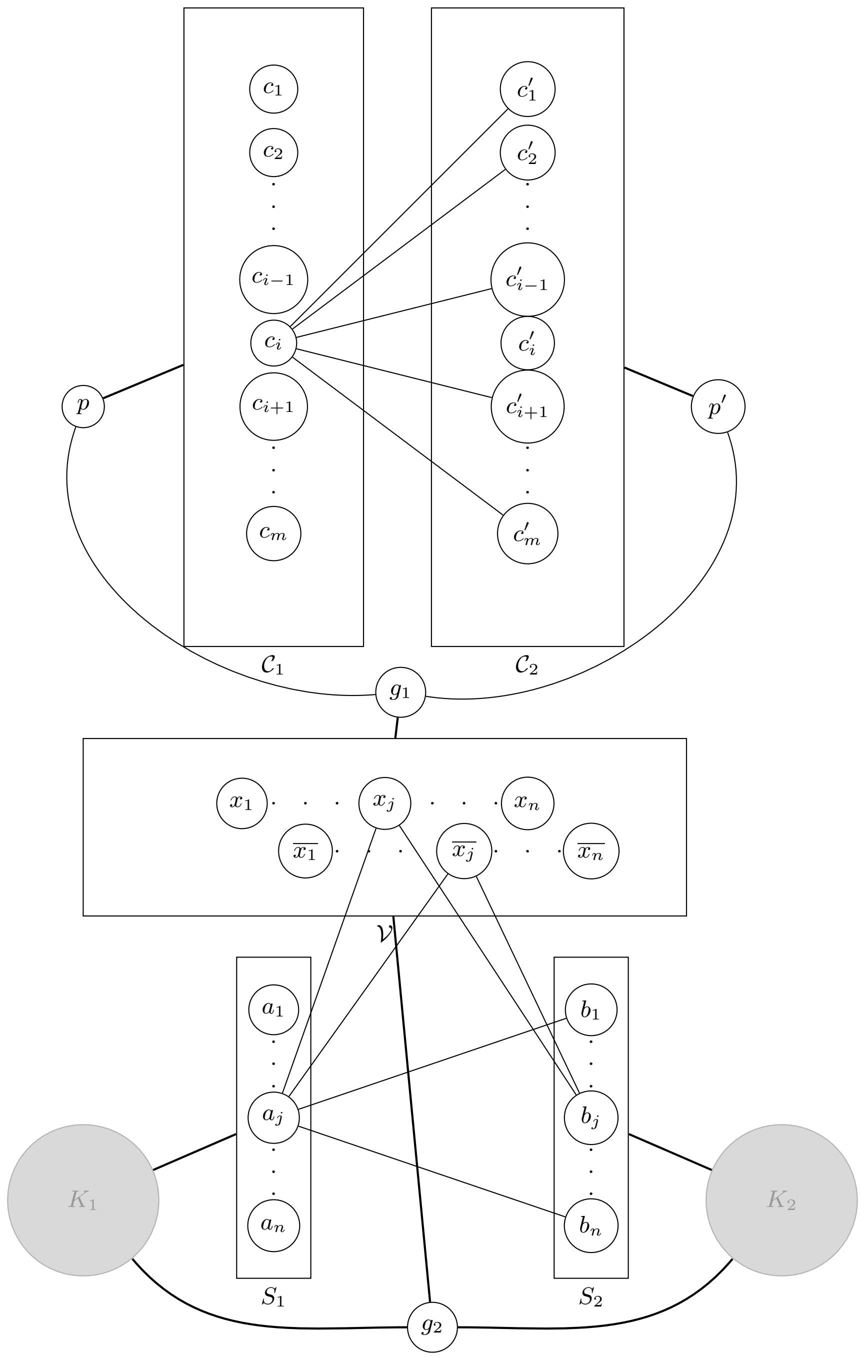

Fig. 1. Graph G constructed from ϕ. Vertices in a gadget which colored gray are completely connected. Edges between the clause gadgets and the variable gadget are not drawn in the figure. Thick lines are used to represent all possible edges between two sets of vertices.

and $\mathcal{V}$, is clearly $2n$ and the number of edges in F which are between $\mathcal{V}$ and $\{g_1, g_2\}$, is also $2n$. So we have that $|F| \leq 4(m+n)$, as desired.

Now, we need to show that $G[E \setminus F]$ consists of two components which are 2-clubs. For $b = 0, 1$, let V_b denote the set of vertices corresponding to literals that evaluate to b under the assignment f. It is easy to see that K_1, K_2, S_1, S_2, g_2 and V_0 form a connected component (call it G_0). Also $\mathcal{C}_1, \mathcal{C}_2, p, p', g_1$ and V_1 form a connected component (call it G_1). We now argue that G_0 and G_1 are 2-clubs.

Any pair of vertices except (a_i, b_i) for all $1 \leq i \leq n$, in the graph induced on S_1, S_2, K_1, K_2, g_2 are at a distance at most two. Since either x_i or $\overline{x_i}$ is in G_0, the distance between a_i and b_i is 2. Since each vertex $y \in G_0$ that corresponds to a literal is adjacent to g_2, the distance between y and vertices in K_b (for $b = 1, 2$) is two in G_0. Finally, since y is adjacent to one vertex from S_1 and one vertex from S_2 (say a_i, b_i), y is at a distance at most two from any vertex in S_1, S_2 (recall that a_i is adjacent to b_j for all $i \neq j$). Since all the vertices in $\mathcal{V} \cap G_0$ has a common neighbor g_2 in G_0, these vertices are also at a distance two from each other in G_0. Hence G_0 is a 2-club (see Table 1).

	S_1	S_2	K_1	K_2	V_0	g_2
S_1	2 (S_2)	2 (V_0)	1	2 (S_2)	2 (S_2)	2 (K_1)
S_2		2 (S_1)	2 (S_1)	1	2 (S_1)	2 (K_2)
K_1			1	2 (g_2)	2 (S_1)	1
K_2				1	2 (S_2)	1
V_0					2 (g_2)	1
g_2						0

Table 1. G_0 is a 2-club

	$\mathcal{C}_1$	$\mathcal{C}_2$	p	p'	V_1	g_1
$\mathcal{C}_1$	2 $(\mathcal{C}_2)$	2 (V_1)	1	2 $(\mathcal{C}_2)$	2 $(\mathcal{C}_2)$	2 (p)
$\mathcal{C}_2$		2 $(\mathcal{C}_1)$	2 $(\mathcal{C}_1)$	1	2 $(\mathcal{C}_1)$	2 (p')
p			0	2 (g_1)	2 $(\mathcal{C}_1)$	1
p'				0	2 $(\mathcal{C}_2)$	1
V_1					2 (g_1)	1
g_1						0

Table 2. G_1 is a 2-club

Now consider G_1. Again any pair of vertices except (c_i, c'_i) for all $1 \leq i \leq m$, in the graph induced on $\mathcal{C}_1, \mathcal{C}_2, p, p', g_1$ are at a distance at most two. Since f is a satisfying assignment, for all $1 \leq i \leq m$ there exists a literal from the clause C_i that is set to 1. Therefore, for each $1 \leq i \leq m$, vertices c_i, c'_i has a common neighbor in G_1. Using arguments similar to the case of G_0, we can show that all vertices $\mathcal{V} \cap G_1$ are at a distance of at most two from all other vertices in G_1. Hence G_1 is a 2-club (see Table 2 for details).

Soundness. Suppose (G, k) is an YES instance of 2-CLUB 2-CLUSTER EDGE DELETION. Let $F \subseteq E(G)$ is the edge deletion set. Let G_a, G_b be the two connected components in $G \setminus F$. We first claim that, without loss of generality, $(K_1 \cup K_2 \cup S_1 \cup S_2 \cup g_2) \subset G_a$. Since K_1 induces a clique of size $k+2$, no set of at most k edges will disconnect K_1. Thus, the vertices of K_1 will belong to one of the two connected components in $G \setminus F$. Without loss of generality, let $K_1 \subseteq G_a$. Since the number of edges between g_2 and K_1, between any vertex in

S_1 and K_1 is $k+2$, $\{g_2\} \cup S_1 \subset G_a$. By similar arguments $\{K_2 \cup S_2 \cup g_2\}$ will belong to the same component. Hence $\{K_1 \cup S_1 \cup g_2 \cup K_2 \cup S_2\} \subset G_a$.

Notice that $N(K_1) = \{a_1, \ldots, a_n, g_2\}$. Consider any $v \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \{p, p', g_1\}$. It is easily checked that for all $1 \leq i \leq n$, $a_i \notin N(v)$, and therefore, $N(v) \cap N(K_1) = \emptyset$. This implies that in G, the vertices of $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \{p, p', g_1\}$ are at a distance more than two from K_1, and therefore, $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \{p, p', g_1\} \subset G_b$.

Observe that for each $1 \leq i \leq m$ c_i and c'_i are at a distance more than two in the graph induced on $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \{p, p', g_1\}$. Also, for each $1 \leq i \leq n$, a_i, b_i are at a distance more than 2 in the graph induced on K_1, K_2, S_1, S_2, g_2. Therefore, these vertices can be made closer only via vertices in $\mathcal{V}$. In particular, for each $1 \leq i \leq n$, a_i, b_i are at a distance of at most two in G_a, at least one of x_i or $\overline{x_i}$ belongs to G_a, or equivalently, at most one of x_i and $\overline{x_i}$ belongs to G_b. Whenever either literal associated with x_i belongs to G_b, we define $f(x_i)$ as follows: $f(x_i) = 1$ if $x_i \in G_b$ and $f(x_i) = 0$ if $\overline{x_i} \in G_b$. If $x_i, \overline{x_i} \in G_a$, then let $f(x_i) = 1$ (the setting is arbitrary). Now we show that the f thus defined is a satisfying assignment. Consider any clause C_j. Since G_b is a 2-club, there exists a vertex y from $\mathcal{V}$ which is a common neighbor of c_j and c'_j. By the definition of f, we have that $f(l_y) = 1$, where l_y is the literal corresponding to the vertex y. So f is a satisfying assignment for ϕ. □

It is now easy to see that 2-Club d-Cluster Edge Deletion cannot be solved in time $2^{o(k)}n^{O(1)}$ unless ETH fails, for any $d \geq 2$. We would reduce from 3-CNF SAT as described in the proof of Lemma 3, and add $d-2$ disjoint cliques of size $k+2$ each to the reduced graph. With this, we have shown the following theorem.

Theorem 4. 2-Club d-Cluster Edge Deletion *for* $d \geq 2$, *cannot be solved in time* $2^{o(k)}n^{O(1)}$, *unless ETH fails.*

4.2 3-Club 2-Cluster Edge Deletion

In this section we will show that 3-Club 2-Cluster Edge Deletion cannot be solved in $2^{o(k)}n^{O(1)}$ unless ETH fails. The proof is a slight modification of the construction described in the proof of Lemma 3.

Lemma 4. (⋆) *There exists a polynomial-time algorithm that, given a 3-CNF formula* ϕ *with* n *variables and* m *clauses, constructs a* 3-Club 2-Cluster Edge Deletion *instance* (G, k) *such that (i)* ϕ *is satisfiable iff* (G, k) *is a* Yes *instance, and (ii)* $k = O(m+n)$.

Theorem 5. 3-Club d-Cluster Edge Deletion *for* $d \geq 2$, *cannot be solved in time* $2^{o(k)}n^{O(1)}$, *unless ETH fails.*

4.3 s-Club d-Cluster Edge Deletion

We now present a general reduction: for all $s \geq 4$, we show that s-Club d-Cluster Edge Deletion cannot be solved in time $2^{o(k)}n^{O(1)}$ unless the ETH fails.

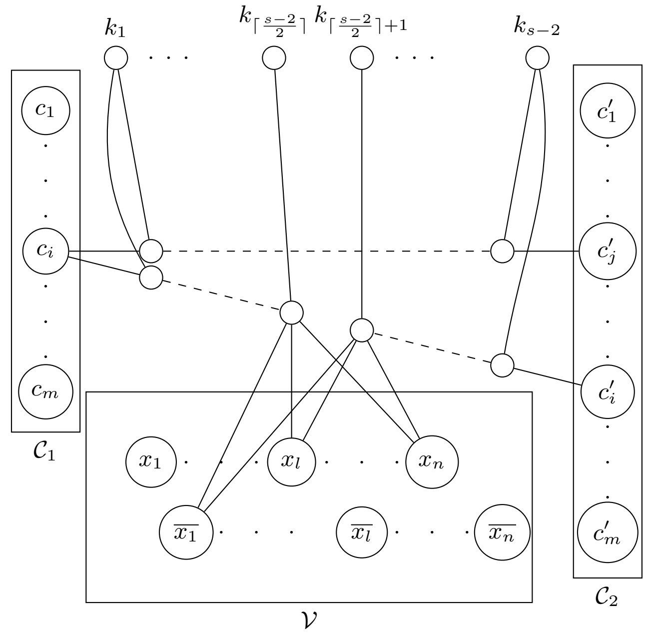

Fig. 2. Example of connection within clause gadget and between clause gadget and variable gadget with $C_i = \overline{x_1} \vee x_l \vee x_n$

Lemma 5. *There exists a polynomial-time algorithm that, given a 3-CNF formula ϕ with n variables and m clauses, constructs a s-*Club 2-Cluster Edge Deletion *instance (G, k) for $s \geq 4$ such that (i) ϕ is satisfiable iff (G, k) is a* Yes *instance, and (ii) $k = O(m + n)$.*

Proof. Let ϕ be a standardized 3-CNF formula with m clauses and n variables. Let $C_1, C_2, \ldots, C_m$ be the clauses and $x_1, x_2, \ldots, x_n$ be the variables.

Construction. We construct a graph $G = (V, E)$ based on ϕ as follows. The graph G contains two clause gadgets $\mathcal{C}_1$ and $\mathcal{C}_2$, s connection gadgets $K_1, \ldots, K_s$, two selection gadgets S_1 and S_2, one variable gadget $\mathcal{V}$ and vertices $\{k_1, \ldots, k_{s-2}\}$. The clause gadget $\mathcal{C}_1$ contains m vertices $c_1, c_2, \ldots, c_m$ and there are no edges within $\mathcal{C}_1$. Similarly, $\mathcal{C}_2$ contains m vertices $c'_1, c'_2, \ldots, c'_m$ and there are no edges within $\mathcal{C}_2$. The variable gadget $\mathcal{V}$ contains $2n$ vertices, one for each literal. Let these vertices be named $x_1, x_2, \ldots, x_n$ and $\overline{x_1}, \overline{x_2}, \ldots, \overline{x_n}$. The connection gadgets $\{K_i\}_{i=0}^{s}$ are cliques of size $k + 2$. The selection gadget S_1 contains n vertices $a_1, a_2, \ldots, a_n$ and no edges within S_1. Similarly, S_2 contains n vertices $b_1, b_2, \ldots, b_n$ and no edges within S_2.

For each $1 \leq i, j \leq m$ we add an edge (c_i, c'_j). For each $1 \leq i, j \leq m$ subdivide the edge (c_i, c'_j) $s-2$ times and let the new vertices be named $t_{ij}(1), \ldots, t_{ij}(s-2)$. Let T denote the set of these newly introduced subdivision vertices. For each

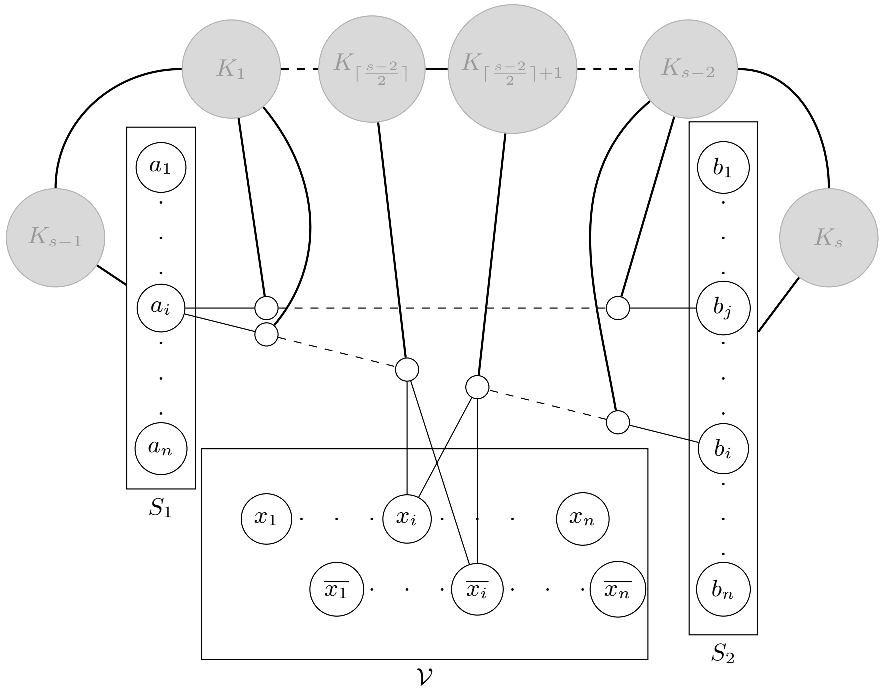

Fig. 3. Example of connection within selection gadget and between selection gadget and variable gadget. Thick lines are used to represent all possible edges between two sets of vertices.

$1 \le i \le m$ delete the edge $(t_{ii}(\lceil\frac{s-2}{2}\rceil), t_{ii}(\lceil\frac{s-2}{2}\rceil + 1))$. Further, add the edges $(k_l, t_{ij}(l))$ for all $1 \le i, j \le m$, $1 \le l \le s - 2$.

If a clause C_i contains a literal x_j then we add two edges $(t_{ii}(\lceil\frac{s-2}{2}\rceil), x_j)$ and $(t_{ii}(\lceil\frac{s-2}{2}\rceil+1), x_j)$. See Fig 2 for a sketch of the clause gadgets and its connection with the variable gadget as described so far.

We now perform an analogous construction between the selection gadgets. For each $1 \le i, j \le n$ we add an edge (a_i, b_j). For each $1 \le i, j \le n$ subdivide the edge (a_i, b_j) $s - 2$ times and let the new vertices be named $u_{ij}(1), \ldots, u_{ij}(s-2)$. In this case, let U denote the set of these newly introduced subdivision vertices. For each $1 \le i \le n$ delete the edge $(u_{ii}(\lceil\frac{s-2}{2}\rceil), u_{ii}(\lceil\frac{s-2}{2}\rceil + 1))$. For each $1 \le j \le n$ add edges $(u_{jj}(\lceil\frac{s-2}{2}\rceil), x_j), (u_{jj}(\lceil\frac{s-2}{2}\rceil + 1), x_j), (u_{jj}(\lceil\frac{s-2}{2}\rceil), \overline{x_j})$ and $(u_{jj}(\lceil\frac{s-2}{2}\rceil), \overline{x_j})$. We add all possible edges between K_l and $t_{ij}(l)$ for all $1 \le i, j \le n$, $1 \le l \le s - 2$. Finally, we add all possible edges between K_i and K_{i+1} for all $1 \le i \le s - 3$. We add all possible edges between K_{s-2} and K_s, between K_{s-1} and K_1, between K_{s-1} and S_1, between K_s and S_2. Fig. 3 shows the selection gadget and its connection with variable gadget. We set $k = 4m+2n$. For a proof of completeness and soundness, we refer the reader to the appendix. □

Theorem 6. *s*-Club *d*-Cluster Edge Deletion *for* $s \geq 4$ *and* $d \geq 2$ *cannot be solved in time* $2^{o(k)}n^{O(1)}$*, unless ETH fails.*

5 Conclusions

In this work, we established that assuming the ETH, there is no algorithm solving the s-Club Cluster Edge Deletion question in time $2^{o(k)}n^{O(1)}$. We also showed that even the problem of deleting edges to obtain a graph with d s-clusters cannot be solved in time $2^{o(k)}n^{O(1)}$ for any $s \geq 2$.

In the context of cluster editing, the exact and approximation results are consistent, in that the general Cluster Editing problem is APX-hard, and does not admit a sub-exponential algorithm unless the ETH fails. On the other hand, the problem of deleting into a sub-linear number of cliques allows for both a sub-exponential algorithm and a PTAS. A natural direction would be to pursue the approximation of these problems so as to either establish or disprove a similar connection in the context of s-clubs.

References

1. R. D. Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3:113–126, 1973.
2. J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
3. F. V. Fomin, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and Y. Villanger. Subexponential fixed-parameter tractability of cluster editing. *To appear in STACS 2013.*, abs/1112.4419, 2013.
4. J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann. A more relaxed model for graph-based data clustering: s-plex cluster editing. *SIAM J. Discrete Math.*, 24(4):1662–1683, 2010.
5. S. Hartung, C. Komusiewicz, and A. Nichterlein. Parameterized algorithmics and computational experiments for finding 2-clubs. In *IPEC*, volume 7535 of *Lecture Notes in Computer Science*, pages 231–241. Springer-Verlag, 2012.
6. S. Hartung, C. Komusiewicz, and A. Nichterlein. On structural parameterizations for the 2-club problem. In *SOFSEM*, pages 233–243, 2013.
7. H. Liu, P. Zhang, and D. Zhu. On editing graphs into 2-club clusters. In *FAW-AAIM*, volume 7285 of *Lecture Notes in Computer Science*, pages 235–246. Springer-Verlag, 2012.
8. D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.
9. R. Niedermeier. *Invitation to Fixed Parameter Algorithms (Oxford Lecture Series in Mathematics and Its Applications)*. Oxford University Press, USA, March 2006.
10. S. Pasupuleti. Detection of protein complexes in protein interaction networks using n-clubs. In *EvoBIO*, volume 4973 of *Lecture Notes in Computer Science*, pages 153–164. Springer-Verlag, 2008.
11. A. Schäfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6(5):883–891, 2012.

6 Appendix

Proof of Lemma 1, Section 3.

Proof. Let F be an optimum solution with the property that number of components in $G \setminus F$ which contains atleast one vertex from X, is minimized. If there exist only one component in $G \setminus F$ which contains X, then we are done. Otherwise there exists atleast two components C_1 and C_2 in $G \setminus F$ such that $X_1 = C_1 \cap X \neq \emptyset$ and $X_2 = C_2 \cap X \neq \emptyset$. Let $p = |C_1 \cap N(X)|$ and $q = |C_2 \cap N(X)|$.
Case 1: $p \leq q$. We claim that $F' = F \cup \{(u,v)|u \in X_1 \wedge v \in C_1 \setminus X_1\} \setminus \{(u,v)|u \in X_1 \wedge v \in C_2\}$ is a solution with less number of components containing vertices from X. Essentially we are making same set of components in $G \setminus F$ with the only difference that X_1 from C_1 is moved to C_2. $|F'| = |F| + |X_1|p - |X_1|(|X_2| + q) \leq |F|$. Note that since $G[N(X)]$ and $G[X]$ are cliques, $C_1 \setminus X_1$ and $C_2 \cup X_1$ are 2-clubs.
Case 2: $p > q$. Similar way we can prove that $F' = F \cup \{(u,v)|u \in X_2 \wedge v \in C_2 \setminus X_2\} \setminus \{(u,v)|u \in X_2 \wedge v \in C_1\}$ is a solution with less number of components containing vertices from X and $|F'| \leq |F|$. ☐

Proof of Lemma 2, Section 3.

Proof. Let ϕ be a standardized 3-CNF formula with m clauses and n variables. Let $C_1, C_2, \ldots, C_m$ be the clauses and $x_1, x_2, \ldots, x_n$ be the variables. For a variable x, let s_x be the number of appearances of x (positively or negatively) in the formula ϕ.

Construction. We construct a graph $G = (V, E)$ as follows. For each variable x we introduce two cycles A_x and B_x of length $2S_x$. Let $a_1, a_2, ..., a_{2s_x}, a_1$ be the cycle A_x and $b_1, b_2, ..., b_{2s_x}, b_1$ be the cycle B_x. Now we will add edges (a_i, b_i) for all i. For each clause C where x appears we assign a C_4 (cycle of length 4) in $G[A_x \cup B_x]$. Name these vertices $a^1_{x,C}, a^2_{x,C}, b^2_{x,C}$ and $b^1_{x,C}$. For example let x be a variable which appears in l clauses (say $C_{i_1}, ..., C_{i_l}$). Then the variable gadget G_x for x will be as shown in Fig. 4.

For each Clause C containing variables x, y, z we will have a clause gadget G_C as shown in the Fig. 5. If x appears positively in C, then q_x will be connected to vertices $a^1_{x,C}, a^2_{x,C}, b^1_{x,C}$ and $b^2_{x,C}$. If x appears negatively in C, then q_x will be connected to vertices $a^2_{x,C}, b^2_{x,C}, a^1_{x,C'}$ and $b^1_{x,C'}$ where $a^1_{x,C'}, b^1_{x,C'}$ are the vertices adjacent to $a^2_{x,C}, b^2_{x,C}$ respectively and $C' \neq C$. We add edges from q_y and q_z in the similar way. For different clauses the vertices appear in the clause gadget will be different. Now we set $k = 8m + 2\sum_{i=1}^{n} s_{x_i} = 14m$. Now we prove that (G, k) is an YES instance if and only if ϕ is satisfiable.

Completeness. Let ϕ is satisfiable and let $f : \{x_1, \ldots, x_n\} \to \{0, 1\}$ be a satisfying assignment. Now we construct the a set $F \subseteq E(G)$ as follows. For each variable x we take edges $(a^2_{x,C}, a^1_{x,C'})$ and $(b^2_{x,C}, b^1_{x,C'})$ where $C \neq C'$ if $f(x) = 1$ and the edges $(a^1_{x,C}, a^2_{x,C})$ and $(b^1_{x,C}, b^2_{x,C})$ if $f(x) = 0$. For each clause C containing x, y, z, C is satisfied via at least one varaibale. Arbiterarly choose such a varaiable (say x). Then add edges between q_y and variable gadgets (4 edges) and edges between q_z and varaiable gadgets (4 edges) to F. It is clear

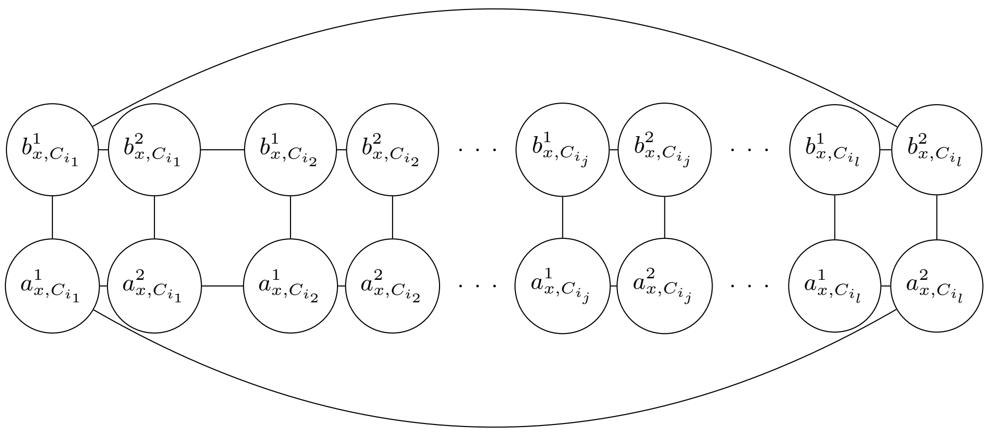

Fig. 4. Variable Gadget G_x corresponding to x which appears in clauses $C_{i_1}, ..., C_{i_l}$

that $|F| = 8m + 2\sum_{i=1}^{n} s_{x_i}$. We calim that every component in $G \setminus F$ is a 2-club. Each clause gadget with a C_4 from a varaible gadget form a 2-club. The remaining vertices from variable gadgets form C_4s (cycle of length 4).

Soundness. Let F is an optimum solution obeying Lemma 1. So for each clause C with variables x, y, z, the vertices p_x, p_y, p_z are contained in a single component in $G \setminus F$. Let G_x be the varaiable gadget we constructed for the varaiable x. We claim that $|F \cap E(G_x)| \geq 2s_x$. In fact, we show that $|F \cap E(G_x)| = 2s_x$ iff any component in $G \setminus F$ either contain a C_4 from G_x or containing no vertices from G_x. We can easily verify that any component from $G \setminus F$ will contain at most 4 vertices from G_x (beacuse vertices that are out side G_x either connects adjacent vertices or connects vertices which are at a distance 2 apart). Any component in $G \setminus F$ which contain 4 vertices from G_x will have either a star with 3 leaves or a C_4 from G_x. Let $r_1, r_2, r_3, r_4^s, r_4^c$ be the number of components from $G \setminus F$ which has a single vertex, two vertices, three vertices, star, C_4 respectively from G_x. Since number of vertices in G_x is $4s_x$

$$r_1 + 2r_2 + 3r_3 + 4r_4^s + 4r_4^c = 4s_x \tag{1}$$

Now we will lower bound $|F \cap E(G_x)|$. For each edge $(u, v) \in F \cap E(G_x)$ we assign two components in $G \setminus F$ (one containing u and other containg v). So a component that contain only single vertex from G_x will be assigned by 3 edges. A component that contain two vertices from G_x will be assigned by at least 4 edges. A component that contain three vertices from G_x will be assigned by at least 5 edges (since G_x is traingle free). A component that contain a star from G_x will be assigned by 6 edges. A component that contain a C_4 from G_x will be

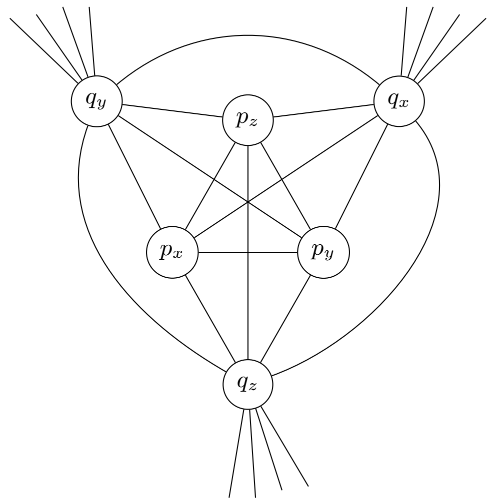

Fig. 5. Clause Gadget G_C corresponding to a clause C which conatins variables x, y, z

assigned by 4 edges. Hence

$$|F \cap E(G_x)| \geq \frac{3r_1 + 4r_2 + 5r_3 + 6r_4^s + 4r_4^c}{2} \quad (2)$$

$$\geq \frac{4s_x + 2r_1 + 2r_2 + 3r_3 + 2r_4^s}{2} \quad \text{(Due to } \textit{equation } 1\text{)} \quad (3)$$

So due to equation 3, $|F \cap E(G_x)| \geq 2s_x$ and $|F \cap E(G_x)| = 2s_x$ only if each component in $G \setminus F$ will either contain a C_4 from G_x or no vertices from G_x.

Let C be a clause containing variables x, y, z. Let G_C be the clause gadget we created. Let E_C be the set of edges in G which has at least one end point in G_C. Now we claim that $|F \cap E_C| \geq 8$ and in fact $|F \cap E_C| = 8$ only if entire G_C with 4 vertices from a variable gadget (a C_4) is a component in $G \setminus F$. Due to Lemma 1, p_x, p_y, p_z are in a single component Q.

Case 1. If q_x, q_y, q_z are not in Q then $|F \cap E_C| \geq 9$.

Case 2. If one of q_x, q_y, q_z is in Q and others are outside Q (say q_x is in Q), then set of edges between $\{q_y, q_z\}$ and $\{p_x, p_y, p_z, q_x\}$ are in F (which counts to 8). So $|F \cap E_C| \geq 8$. In fact, if $|F \cap E_C| = 8$ and $q_y, q_z \notin Q$, then all four vertices adjacent to q_x in G_x (i.e, C_4 adjacent to q_x) will be part of Q.

Case 3. If one of q_x, q_y, q_z is out side Q and others are in Q (say q_z is outside Q). Then the edges between q_z and $\{p_x, p_y, p_z, q_x, q_y\}$ are in F (which counts to 5). We claim that only neigbours of q_x or q_y can be present in Q. If one neighbor v of q_x and another neighbor u of q_y in variable gadgets are present in Q, then the distance between u and v in Q is 3, which is a contradiction that Q is a 2-club. Hence in this case also $|F \cap E_C| \geq 9$.

Case 4. If q_x, q_y, q_z are in Q. By same argument in case 3, we claim that only neigbours of one of q_x, q_y, q_z can be there in Q. Hence $|F \cap E_C| \geq 8$. If $|F \cap E_C| = 8$ one of q_x, q_y, q_z should contain all of its neighbors (i.e a C_4).

So far we proved that optimum solution is at least $8m + 2\sum_{i=1}^{n} s_{x_i}$ and we can get a solution of size $8m + 2\sum_{i=1}^{n} s_{x_i}$ only in limited way, i.e variable gadget will be decomposed into C_4 s and clasue gadget will form a single component with 4 vertices from a variable gadget (which is a C_4). Now suppose G has a solution of size $8m + 2\sum_{i=1}^{n} s_{x_i}$. Let the solution be F (obeying Lemma 1). Now we know that variable gadget will decompose into C_4 s. Suppose $a^1_{x,C}, a^2_{x,C}, b^2_{x,C}, b^1_{x,C}$ form a C_4 in $G \setminus F$, then we set $f(x) = 1$ otherwise $f(x) = 0$. We know that each component containg clause gadget G_C will contain 4 more vertices (that will be a C_4) and hence the corresponding variable will satisfy C. □

Proof of Lemma 4, Section 4.2.

Proof. Let ϕ be a standardized 3-CNF formula with m clauses and n variables. Let $C_1, C_2, \ldots, C_m$ be the clauses and $x_1, x_2, \ldots, x_n$ be the variables.
Construction. Here we modify the construction as given in the proof of Lemma 3 slightly. We first construct the graph G as as given in the proof of Lemma 3. Then we subdivide all the edges between $\mathcal{C}_1$ and $\mathcal{C}_2$, and let P denote the set of these subdivision vertices. Similarly, subdivide all the edges between S_1 and S_2 and let Q denote the set of these subdivision vertices. Finally, add a new connection gadget K_3, a clique of size $k+2$. The additional edges are as follows: make g_1 and g_2 global to P and Q, respectively, and add all possible edges between K_3 and K_1, between K_3 and K_2. This completes the description of the reduced graph obtained from ϕ. We set $k = 4(m+n)$.
Completeness. Let ϕ be satisfiable and $f : \{x_1, \ldots, x_n\} \rightarrow \{0, 1\}$ be a satisfying assignment. Now we construct the edge deletion set $F \subseteq E(G)$ exactly as in the proof of Lemma 3, and thus, we have $|F| \leq 4(m+n)$.

Now we need to show that $G[E \setminus F]$ consists of two components which are 3-clubs. For reasons similar to before, it is easy to see that $K_1 \cup K_2 \cup K_3 \cup S_1 \cup S_2 \cup Q \cup \{g_2\}$ along with the vertices corresponding to literals which are assigned 0 (we denote this set by V_0) form a connected component (call this G_0). Also $\mathcal{C}_1 \cup \mathcal{C}_2 \cup P \cup \{p, p', g_1\}$ together with the vertices corresponding to literals which are assigned 1 (we denote this set by V_1) form a connected component (say G_1). We now demonstrate that G_0 and G_1 are 3-clubs.

Notice that other than the pairs (a_i, b_i) (for $1 \leq i \leq n$), any pair in the graph induced by $K_1 \cup K_2 \cup K_3 \cup S_1 \cup S_2 \cup Q \cup \{g_2\}$ is at a distance of at most 3. Since either x_i or $\overline{x_i}$ is in G_0, the distance between a_i and b_i is 2. Further, since each vertex y corresponding to a literal in G_0 is adjacent to g_2, and g_2 is adjacent to K_i (for $i = 1, 2$) and Q, the distance between y and any vertex in $K_1 \cup K_2 \cup Q$ is two, and the distance between y and K_3 is three (since vertices in K_3 are adjacent to K_1 and K_2). Also, since y is adjacent to one vertex from S_1 and one vertex from S_2 (say a_i, b_i), y is at a distance at most 3 from any vertex in S_1, S_2 (recall that a_i is at a distance 2 from b_j for all $i \neq j$). Finally, since all the vertices in V_0 have a common neighbor g_2 in G_0, these vertices are also at a distance two from each other in G_0. Hence G_0 is a 3-club.

Now consider G_1. Again, other than the pairs (c_i, c_i') for all $1 \leq i \leq m$, all pairs of vertices in the graph induced on $\mathcal{C}_1 \cup \mathcal{C}_2 \cup P \cup \{p, p', g_1\}$ are at a distance at most 3. Since f is a satisfying assignment for all $1 \leq i \leq m$, there exists a literal from the clause C_i that evaluates to 1 under f, and therefore for each $1 \leq i \leq m$, the vertices c_i, c_i' have a common neighbor in G_1 via V_1. Again, by virtue of having $g_1 \in G_1$ has a common neighbor, and using arguments similar to the ones above, we see that all vertices in V_1 are at a distance at most 3 from all other vertices in G_1. Hence G_1 is a 3-club.

Soundness. Suppose (G, k) is an YES instance of 3-CLUB 2-CLUSTER EDGE DELETION. Let $F \subseteq E(G)$ is the edge deletion set. Let G_a, G_b be the two connected components in $G \setminus F$.

Since K_1 induces a clique of size $k+2$, no set of at most k edges will disconnect K_1. Thus, the vertices of K_1 will belong to one of the two connected components in $G \setminus F$. Without loss of generality, let $K_1 \subseteq G_a$. Since the number of edges between g_2 and K_1, between any vertex in S_1 and K_1 is $k + 2$, $\{g_2\} \cup S_1 \subset G_a$. Similarly, $\{K_2 \cup S_2 \cup g_2\}$ and $\{K_3, g_2\}$ are inseparable sets, and hence $K_1 \cup K_2 \cup K_3 \cup S_1 \cup S_2 \cup \{g_2\} \subset G_a$.

Now, note that the neighbors of K_3 are $K_1 \cup K_2$, and $N(K_1 \cup K_2) = S_1 \cup S_2$. On the other hand, for any $v \in \mathcal{C}_1 \cup \mathcal{C}_2 \cup \{p, p', g_1\}$, it is clear that $N(v) \cap (S_1 \cup S_2) = \emptyset$. Therefore, the vertices of K_3 are at a distance of more than three from vertices in $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \{p, p', g_1\}$. This implies that $\mathcal{C}_1 \cup \mathcal{C}_2 \cup \{p, p', g_1\}$ will belong to the other component, namely, G_b.

It is easily checked that there are no paths of length exactly three between a_i and b_i; and between c_i and c_i'. So the only way to make them close in G_a and G_b, respectively, is via a path of length 2, i.e having a common neighbor for these pairs from $\mathcal{V}$. In particular, for each $1 \leq i \leq n$, at least one of x_i or $\overline{x_i}$ belongs to G_a to ensure that (a_i, b_i) are distance two apart. Equivalently, at most one of x_i and $\overline{x_i}$ belongs to G_b. Whenever either literal associated with x_i belongs to G_b, we define $f(x_i) = 1$ if $x_i \in G_b$ and $f(x_i) = 0$ if $\overline{x_i} \in G_b$. Finally, if $x_i, \overline{x_i} \in G_a$, then let $f(x_i) = 1$ (the setting is arbitrary). Now we show that the f thus defined is a satisfying assignment. Consider any clause C_j. Since G_b is a 3-club, and there are no paths of length exactly three between c_j and c_j', there exists a vertex y from $\mathcal{V}$ which is a common neighbor of c_j and c_j'. By the definition of f, we have that $f(l_y) = 1$, where l_y is the literal corresponding to the vertex y. So f is a satisfying assignment for ϕ. ☐

The proof of soundness and completeness, for the reduction described in Lemma 5, Section 4.3.

Proof. (Contd.) **Completeness.** Let ϕ be satisfiable and $f : \{x_1, \ldots, x_n\} \rightarrow \{0, 1\}$ be the satisfying assignment. Now we construct the edge deletion set $F \subseteq E(G)$ as follows. For each $1 \leq i \leq n$, if $f(x_i) = 1$, then include the following edges in F:

$$\left\{ \left(t_{jj} \left(\left\lceil \frac{s-2}{2} \right\rceil \right), \overline{x_i} \right), \left(t_{jj} \left(\left\lceil \frac{s-2}{2} \right\rceil + 1 \right), \overline{x_i} \right) \;\middle|\; \overline{x_i} \in C_j \right\}$$

$$\bigcup \left\{ \left(u_{ii} \left(\left\lceil \frac{s-2}{2} \right\rceil \right), x_i \right), \left(u_{ii} \left(\left\lceil \frac{s-2}{2} \right\rceil + 1 \right), x_i \right) \right\}.$$

Note that these are the edges between x_i and the selection gadgets, and also edges between $\overline{x_i}$ and the clause gadgets. On the other hand, if $f(x_i) = 0$, then we add the following edges to F:

$$\left\{ \left(t_{jj} \left(\left\lceil \frac{s-2}{2} \right\rceil \right), x_i \right), \left(t_{jj} \left(\left\lceil \frac{s-2}{2} \right\rceil + 1 \right), x_i \right) \;\middle|\; x_i \in C_j \right\}$$
$$\bigcup \left\{ \left(u_{ii} \left(\left\lceil \frac{s-2}{2} \right\rceil \right), \overline{x_i} \right), \left(u_{ii} \left(\left\lceil \frac{s-2}{2} \right\rceil + 1 \right), \overline{x_i} \right) \right\}.$$

This is similar to the choice before, except that it is flipped; in that these are the edges between $\overline{x_i}$ and the selection gadgets, and edges between x_i and the clause gadgets.

Clearly, we have $|F| \leq 4m + 2n$. Now we need to show that $G[E \setminus F]$ consists of two components which are s-clubs. For $b \in \{0, 1\}$, let V_b the set of vertices corresponding to literals which are assigned b. It is easy to see that $K_1, \ldots, K_s, S_1, S_2, U$ and V_0 form a connected component (which we denote by G_0). Also $\mathcal{C}_1, \mathcal{C}_2, T, k_1, \ldots, k_{s-2}$ and V_1 form a connected component (which we denote by G_1). Now we need to show that G_0 and G_1 are s-clubs.

First, observe that other than (a_i, b_i) (for $1 \leq i \leq n$), every pair of vertices in the graph induced by $S_1 \cup S_2 \cup K_1 \cup \cdots \cup K_s \cup U$, are at a distance at most s (for $s \geq 4$). Since either x_i or $\overline{x_i}$ is in G_0, the distance between a_i and b_i is s. Since each vertex y corresponding to a literal in G_0 is adjacent to $u_{ii}(\lceil \frac{s-2}{2} \rceil)$ and $u_{ii}(\lceil \frac{s-2}{2} \rceil + 1)$ for some i, the distance between y and vertices in K_i (for $i \in [s]$) is at most s in G_0. Also, y is at a distance at most s from vertices in S_1, S_2 and the vertices of U (for $s \geq 4$). All the vertices in V_0 are at a distance 4, via, for instance, a vertex in $K_{\lceil \frac{s-2}{2} \rceil}$. Hence G_0 is a s-club.

Now consider G_1. Again, other than (c_i, c_i') (for $1 \leq i \leq m$), any pair of vertices in the graph induced by $\mathcal{C}_1 \cup \mathcal{C} \cup_2 \cup \{k_1, \ldots, k_{s-2}\} \cup T$ are clearly at a distance at most s. Since f is a satisfying assignment, for all $1 \leq i \leq m$, there exists a literal from the clause C_i that is set to 1. Therefore, the vertices c_i, c_i' are s-apart in G_1. Using arguments analogous to those in the case for G_0, we observe that all vertices V_1 are at a distance at most s from all other vertices in G_1. Hence, G_1 is a s-club.

Soundness. Now, suppose (G, k) is an Yes instance of s-Club 2-Cluster Edge Deletion. Let $F \subseteq E(G)$ is the edge deletion set. Let G_a, G_b be the two connected components in $G \setminus F$. As in the previous reductions, notice that the connection gadgets $K_1, \ldots, K_s$ are inseparable by edge sets of size at most k, and since every vertex in $S_1 \cup S_2$ are attached to these gadgets by more than k edges, we have (without loss of generality) that $K_1 \cup \cdots \cup K_s \cup S_1 \cup S_2 \subset G_a$.

On the other hand, note that all vertices in $\mathcal{C}_1 \cup \mathcal{C}_2$ are at a distance of more than s from K_{s-1} in G, and therefore G_a will not contain $\mathcal{C}_1 \cup \mathcal{C}_2$. Thus we have that $\mathcal{C}_1, \mathcal{C}_2 \subset G_b$.

It is also easy to see that the shortest path length between a_i and b_i is s and any shortest path between a_i and b_i should contain either x_i or $\overline{x_i}$. Notice that this implies that for all $1 \leq i \leq n$, at most one of x_i or $\overline{x_i}$ belongs to G_b. Additionally, recall that the shortest path length between c_i and c_i' is s and any shortest path between c_i and c_i' should contain a vertex corresponding to a literal which is there in the clause C_i.

With this, we are ready to describe a satisfying assignment f for ϕ. Whenever either literal associated with x_i belongs to G_b, we define $f(x_i) = 1$ if $x_i \in G_b$ and $f(x_i) = 0$ if $\overline{x_i} \in G_b$. Otherwise, if $x_i, \overline{x_i} \in G_a$, then let $f(x_i) = 1$ (the setting is arbitrary). To see that f is indeed satisfying, consider any clause C_j. Since the distance between c_j and c_j' is s in C, there exists a vertex y from $\mathcal{V}$ such that clause C_j contains literal corresponding to y. In our satisfying assignment we set $f(y) = 1$. So f is a satisfying assignment for ϕ. □