γ-clustering problems: Classical and parametrized complexity

Julien Baste, Antoine Castillon, Clarisse Dhaenens, Mohammed Haddad, Hamida Seba

▶ **To cite this version:**

Julien Baste, Antoine Castillon, Clarisse Dhaenens, Mohammed Haddad, Hamida Seba. γ-clustering problems: Classical and parametrized complexity. Theoretical Computer Science, 2024, 1018, pp.114784. 10.1016/j.tcs.2024.114784 . hal-04693347

HAL Id: hal-04693347
https://hal.science/hal-04693347v1

Submitted on 13 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

γ-clustering problems: classical and parametrized complexity

Julien Baste*, Antoine Castillon*†, Clarisse Dhaenens*, Mohammed Haddad†, and Hamida Seba†

*Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL, F-59000 Lille, France
†Univ Lyon, UCBL, CNRS, INSA Lyon, LIRIS, UMR5205, F-69622 Villeurbanne, France

Abstract

We introduce the γ-clustering problems, which are variants of the well-known Cluster Editing/Deletion/Completion problems, and defined as: given a graph G, how many edges must be edited in G, deleted from G, or added to G in order to have a disjoint union of γ-quasi-cliques. We provide here the complete complexity classification of these problems along with FPT algorithms parameterized by the number of modifications, for the NP-complete problems. We also study here a variant of these problems where the number of final clusters is a fixed constant, obtaining mostly the same results regarding classical and parameterized complexity.

1 Introduction

Among important problems in graph theory, only a few of them are as central as clustering, i.e., finding a partition of a graph into relevant clusters. From a practical point of view, clustering problems are used in data analysis to identify community structures in social networks [7, 9] as well as in computational biology to identify molecular modules in protein interaction networks [19] or to identify similar genes [3]. From a theoretical point of view, the Cluster Editing problem, where given a graph G, one has to transform G into a cluster graph with a minimal number of modifications, has been studied extensively. Plenty of results have been found, and later improved, regarding the classical complexity [17], the approximability [1, 6] or the parameterized complexity [5, 10].

Note that a graph G is a cluster graph if, and only if, P_3 is not an induced subgraph of G. Hence, the Cluster Editing problem is often introduced as the edge modification problem with respect to the property of being P_3-free, i.e. without any induced P_3. This implies a very strict and ordered structure, especially on the small induced subgraphs of a solution graph. Such a structure is usually hard to match with the complexity and chaos of real life graphs such as social networks or protein networks which are the most common application cases of the Cluster Editing problem [3, 9, 7, 19]. In this paper we investigate a variant of Cluster Editing where we relax the requirement of the solution graph, replacing cliques with quasi-cliques.

Quasi-cliques are a natural way to extend the definition of cliques to other dense graphs. Several definitions of quasi-cliques exist in the literature: the density-based version [15] where the proportion of existing edges in the quasi-clique must be greater than a constant γ, the degree-based version [14] where the vertices must be adjacent to a proportion γ of the quasi-clique and the degree/density-based version [4] which is an amalgam of the previous two definitions. In this paper, we focus on the degree-based definition which encapsulates well the intuitive notion of a quasi-clique.

	Clique	$\gamma < 1$	Clique, p-clusters	$\gamma < 1$, p-clusters
Completion	P	P	P	P
Deletion	NPC	NPC	NPC for $p \geq 3$	NPC for $p \geq 2$
Editing	NPC	NPC	NPC for $p \geq 2$	NPC for $p \geq 2$

Table 1: Complexity classification of the γ-clustering problems. The columns corresponding to Clique are results from [17].

	Clique	$\gamma < 1$	$\gamma < 1$, p-clusters
Completion	P	P	P
Deletion	$2^k \cdot n^{O(1)}$	$2^{k \log(2k)} \cdot n^{O(1)}$	$2^{O(k \log k)} \cdot n^{O(1)}$
Editing	$2.27^k \cdot n^{O(1)}$	$2^{O(k \log k)} \cdot n^{O(1)}$	$2^{O(k \log k)} \cdot n^{O(1)}$

Table 2: Parameterized complexity of the γ-clustering problems. Again, the columns corresponding to Clique are known results from [5, 10].

Unsurprisingly, most hardness results holding with cliques also holds with quasi-cliques [2, 12, 14, 15]. For instance, it is NP-hard to find a quasi-clique of a given size [14, 15]. However, other hardness results tend to prove that the study of quasi-cliques is usually even harder than the study of cliques [16, 20]. For instance, even checking the maximality of a γ-degree-based quasi-clique is NP-hard [16]. This difficulty can be attributed to the non-heredity of being a quasi-clique, i.e. an induced subgraph of a quasi-clique is not necessarily a quasi-clique (see Remark 4).

As previously said, we introduce here new variants of the Cluster Editing problem where quasi-cliques replace cliques in the solution graph. To the best of our knowledge, quasi-cliques or other relaxations of cliques such as s-plexes, s-clubs or s-cliques have only been studied regarding the maximal induced subgraph problem [12] and never as a relaxation of cliques in other problems such as the Cluster Editing problem. The closest relaxation of Cluster Editing that we know is the research of (p, q)-clusters [11, 13], where each cluster misses at most p-edges to be a clique and is still linked to at most q other vertices. However, this approach only focuses on recognizing such graphs and does not involve edge modifications.

Our contributions are the introduction of the γ-clustering problems as well as their extensive complexity classification listed in Table 1. We also introduce natural variants of these problems, called the (γ, p)-clustering problems, where the solution graph must have exactly p clusters. Finally, we also provide FPT algorithms parameterized by k for our NP-complete problems as shown in Table 2.

The remainder of this article is organized as follows. Section 2 introduces our notations, the formal definitions, the problems studied as well as a few useful properties. In Section 3, we provide the proofs of the results presented in Table 1. Section 4 contains the two FPT algorithms announced in Table 2. Finally, we conclude this article in Section 5.

2 Preliminaries

In this paper, the graphs are always considered simple and undirected. Also, we use the following notations.

Given two reals a and b we use the usual notations $[a, b]$, $[a, b[$, $]a, b]$ and $]a, b[$ for respectively the closed, the two half-open and the open intervals of numbers between a and b. We also note: $[\![a, b]\!] = \{x \in \mathbb{Z} \mid a \leq x \leq b\}$, where $\mathbb{Z}$ denotes the set of all integers. Given two sets E and F, we denote by $E \Delta F$ the symmetric difference between E and F, i.e. $E \Delta F = (E \cup F) \setminus (E \cap F)$. Given a set X and an integer $k \in \mathbb{Z}_{\geq 0}$, we note $\binom{X}{k}$ the set of subsets of X of size k. Hence, given a set of vertices V, the set of all possible edges between the vertices of V is $\binom{V}{2}$. Also,

given two disjoint sets X and Y, we divert from the usual notation, denoting $X \times Y$ the set of unordered pairs containing one element of X and one element of Y: $X \times Y = \{\{x, y\} \mid x \in X, y \in Y\}$.

Given a graph G, we denote $V(G)$ the vertex set of G and $E(G) \subseteq \binom{V(G)}{2}$ its edge set. The subgraph (resp. bipartite subgraph) of G induced by a set of vertices X (resp. two sets of vertices X, Y) is noted $G[X] = (X, E(G) \cap \binom{X}{2})$ (resp. $G[X, Y] = (X \cup Y, E(G) \cap (X \times Y))$). We denote by $E_G(X)$ the set $E(G[X])$ and $E_G(X, Y)$ the set $E(G[X, Y])$, if G is clear from the context, we refer to them as $E(X)$ and $E(X, Y)$. We note $N_G(u)$ the neighborhood of u, and $d_G(u) = |N_G(u)|$ its degree. Again, if G is clear from the context, we refer to them as $N(u)$ and $d(u)$. Also, if we want to focus on the neighborhood of u within a set of vertices $X \subseteq V(G)$ we note : $N_X(u) = N(u) \cap X$ and $d_X(u) = |N_X(u)|$, note that these definitions also holds if $u \notin X$. The minimal and maximal degree of G are denoted respectively $\delta(G)$ and $\Delta(G)$. Finally, the distance between two vertices $u, v \in V(G)$, i.e. the number of edges on the shortest path connecting u to v is denoted $\text{dist}_G(u, v)$ and the diameter of G is denoted $\text{diam}(G) = \max_{u,v \in V} \text{dist}_G(u, v)$.

The well-known Cluster Editing problem [17] can be defined as follows.

Problem 1 (Cluster Editing)**.** Given a graph $G = (V, E)$ and an integer k, does it exist a set $S \subseteq \binom{V}{2}$ such that $|S| \leq k$ and each connected component of $G \Delta S = (V, E \Delta S)$ is a clique ?

The graph $G \Delta S$ is often referred to as a *solution graph* and a disjoint union of cliques is referred to as a *cluster graph*.

Definition 2 (γ-quasi-clique)**.** Given $\gamma \in [0, 1]$, a graph $G = (V, E)$ is a *γ-quasi-clique*, if for all $u \in V$, $d(u) \geq \gamma(|V| - 1)$.

Given a graph G, we also say that a set of vertices $X \subseteq V(G)$ is a γ-quasi-clique of G if $G[X]$ is a γ-quasi-clique. Note that we provide the general definition of γ-quasi-cliques with values of γ varying from 0 to 1. However, since we want quasi-cliques to represent dense graphs, in this paper, we only consider values of γ strictly larger than $\frac{1}{2}$. Also, note that 1-quasi-cliques are exactly cliques. Since we want to emphasize the similarities and differences with cliques, i.e. cases where $\gamma = 1$, we focus here on cases where $\gamma < 1$. In the remainder of this paper and unless it is clearly specified otherwise, we always assume that $\frac{1}{2} < \gamma < 1$.

We introduce the γ-Cluster Editing problem as follows:

Problem 3 (γ-Cluster Editing)**.** Given a graph $G = (V, E)$ and an integer k, does it exist a set $S \subseteq \binom{V}{2}$ such that $|S| \leq k$ and each connected component of $G \Delta S = (V, E \Delta S)$ is a γ-quasi-clique ?

We will also refer to $G \Delta S$ as a *solution graph* and a disjoint union of γ-quasi-cliques as a *γ-cluster graph*.
We introduce in a similar way the γ-Cluster Deletion problem, where edges can only be removed, i.e. $S \subseteq E$, and the γ-Cluster Completion problem, where edges can only be added, i.e. $S \cap E = \emptyset$.

Remark 4. Contrary to the Cluster Editing problem, there is no strict structure on the small induced subgraphs of a solution graph. In fact, for any $\gamma \in [0, 1[$ and any given graph G on n vertices, there exists a graph G' on $\lceil \frac{1}{1-\gamma} n \rceil$ vertices such that G' is a γ-quasi-clique and G is an induced subgraph of G'. Hence a solution graph can contain any given graph as an induced subgraph.

Proof. (Remark 4) Let G be a graph on n vertices. Let G' be the graph obtained by adding $\lceil \frac{1}{1-\gamma} |V| \rceil - |V|$ new universal vertices to G. It holds that $G = G'[V]$ is an induced subgraph

of G' and G' is a γ-quasi-clique. Indeed, for all $v \in V(G')$, $d(v) \geq \lceil \frac{1}{1-\gamma}|V| \rceil - |V| \geq \frac{\gamma}{1-\gamma}|V| \geq \gamma(\lceil \frac{1}{1-\gamma} n \rceil - 1) = \gamma(|V(G')| - 1)$. □

We also study variants of the γ-clustering problems where the number of clusters is fixed to a constant p.

Problem 5 ((γ, p)-Cluster Editing)**.** Given a graph $G = (V, E)$ and an integer k, does it exist a set $S \subseteq \binom{V}{2}$ such that $|S| \leq k$, $G \Delta S = (V, E \Delta S)$ have exactly p connected components and each one of them is a γ-quasi-clique ?

Again we introduce in a similar way the (γ, p)-Cluster Deletion and (γ, p)-Cluster Completion problems.

We provide here three useful properties used later in the proofs. These properties describe respectively a relation between $\delta(G)$ and $|V(G)|$ when G is a γ-quasi-clique, the diameter of a γ-quasi-clique and an edge-connectivity result.

Remark 6. Given G a γ-quasi-clique it holds that $\delta(G) \geq \gamma(|V(G)|-1)$. Conversely, $|V(G)| \leq \frac{1}{\gamma}\delta(G) + 1$.

Lemma 7. *Given $\gamma \in]\frac{1}{2}, 1[$, G a γ-quasi-clique and $u, v \in V$ two vertices. It holds that u and v have at least $(2\gamma - 1)(|V| - 1)$ common neighbors.*

Proof. (Lemma 7) Let $\overline{N}(u)$ be the vertices of V not adjacent to u. It holds that $|\overline{N}(u)| \leq (1 - \gamma)(|V| - 1)$ hence : $|N(v) \setminus \overline{N}(u)| \geq (2\gamma - 1)(|V| - 1)$. □

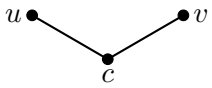
A direct consequence of this lemma is that the diameter of a γ-quasi-clique is at most 2 if $\gamma > \frac{1}{2}$.

Lemma 8. *Let $G = (V, E)$ be a graph and d be an integer such that $\delta(G) \geq d$ and $\mathrm{diam}(G) \leq 2$. It holds that G is d-edge-connected, i.e. it remains connected even after removing $d - 1$ edges.*

Proof. (Lemma 8) Let $G = (V, E)$ be a graph and d be a non-negative integer such that $\delta(G) \geq d$ and $\mathrm{diam}(G) \leq 2$. Given two vertices $u, v \in V$ let's prove that there are at least d edge-disjoint paths connecting u to v.

Let's note $A = N(u) \setminus N(v)$, $B = N(v) \setminus N(u)$ and $C = N(u) \cap N(v)$.

For all $c \in C$, (u, c, v) is a path connecting u to v.



We note $|A| = s$, $|B| = t$, and:

$$A = \{a_1, ..., a_s\}, \qquad B = \{b_1, ..., b_t\}.$$

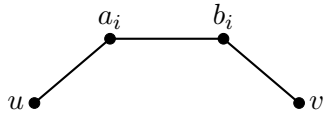
Without loss of generality, we also assume that $t \geq s$ (i.e. $d(u) \leq d(v)$).

Let M be a maximal matching of $G[A, B]$, up to a permutation we can assume that:

$$M = \{\{a_1, b_1\}, ..., \{a_r, b_r\}\}.$$

Let $A_1 = \{a_1, ..., a_r\}$, $A_2 = A \setminus A_1$, $B_1 = \{b_1, ..., b_r\}$ and $B_2 = B \setminus B_1$.

For all $i \in [\![1, r]\!]$, (u, a_i, b_i, v) is a new path connecting u to v.



Also, since M is maximal, it holds that $E(A_2, B_2) = \emptyset$. So, for all $j \in [\![r+1, s]\!]$, a_j and b_j are not adjacent. However, it holds that $\text{dist}_G(a_j, b_j) \leq 2$. Hence, let $z_j \in V$ be a vertex such that $\{a_j, z_j\}, \{z_j, b_j\} \in E$.

It holds that:

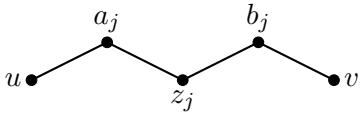
- $z_j \neq u$: otherwise $b_j \in C = N(u) \cap N(v)$.

- $z_j \neq v$: otherwise $a_j \in C = N(u) \cap N(v)$.

- $z_j \notin A_2$: otherwise $\{z_j, b_j\} \in E(A_2, B_2)$.

- $z_j \notin B_2$: otherwise $\{a_j, z_j\} \in E(A_2, B_2)$.

Also, if $z_j \in A_1, B_1, C$ or $V \setminus (A \cup B \cup C)$ then (u, a_j, z_j, b_j, v) is a new path connecting u to v.



To sum up, we have $|C|$ paths of length 2, r paths of length 3 and $s - r$ paths of length 4. Hence, there is a total of $|C| + s = d(u) \geq d$ such paths as shown in Figure 1. Since these paths are edge-disjoints, the result holds.

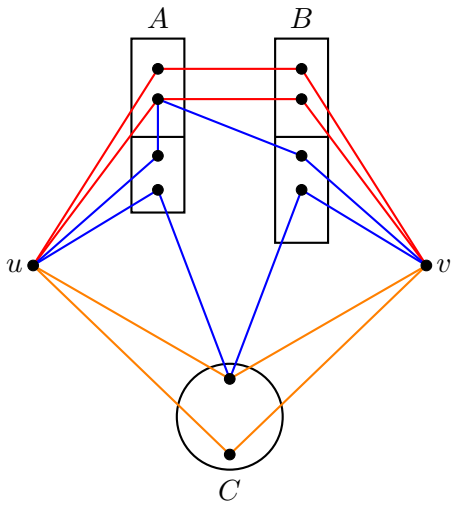


Figure 1: $d(u)$ disjoint paths between u and v: $|C|$ yellow paths of length 2, r red paths of length 3 and $s - r$ blue paths of length 4.

□

3 Classical Complexity

In this section we classify between polynomial and NP-hard the γ-clustering problems. Before starting the proofs, it is important to note that all of these problems are in NP. Indeed given a graph G and a set of modification S, one can compute $G' = (V, E \cup S)$, or $G' = (V, E \setminus S)$, or $G' = (V, E \Delta S)$ in polynomial time. Moreover, one can also verify in polynomial time that G' is a γ-cluster graph, i.e. if its connected components are all γ-quasi-cliques. Finally, if we have a given number of clusters p, it is easy to check if G' has exactly p connected components. Hence, for all of these problems we provide either a polynomial time algorithm which solves the

problem or a polynomial reduction from another NP-complete problem, usually the Clique problem.

This section is divided into four subsections, in Sub-section 3.1 we show that the γ-Cluster Completion problem is polynomial, in Sub-section 3.2 we prove the NP-hardness of γ-Cluster Deletion and γ-Cluster Editing and in Sub-section 3.3 we tackle the (γ, p)-clustering problems.

3.1 γ-Cluster Completion

This subsection is dedicated to the proof of the following theorem.

Theorem 9. *For $\gamma \in \left]\frac{1}{2}, 1\right[$, the $\gamma-$Cluster Completion problem is solvable in polynomial time.*

Proof. (Theorem 9) We start with Lemma 10 which proves the existence of an optimal solution that never add any edge between the connected components of the input graph. Hence, we can solve the problem independently on each connected component. Then, Lemma 14 concludes the proof of the theorem by showing that an optimal solution to the γ-Cluster Completion problem can be found in polynomial time on a connected graph.

Lemma 10. *Given $G = (V, E)$ a graph with two connected components A and B, there exists S an optimal solution to the $\gamma -$ Cluster Completion problem on G such that S does not contain any edge between A and B.*

Proof. (Lemma 10) We first assume that $|A|, |B| \geq 2$. Let S be an optimal solution to the γ-Cluster Completion problem. We assume that S contains at least an edge between A and B. Thus, it holds that $A \cup B$ is a γ-quasi-clique of $G \cup S = (V, E \cup S)$ and the degree of every vertex of $A \cup B$ is greater than $\gamma(|A| + |B| - 1)$ in $G \cup S$. Let's construct another optimal solution S', such that S' does not contain any edge between A and B.

If we construct such S' note that the degree required for each $a \in A$ drop from $\gamma(|A| + |B| - 1)$ to $\gamma(|A| - 1)$, and similarly for each $b \in B$.

We start with $S' = S$ and we apply the following rules. We also assume that a rule is applied only if all previous rules have already been extensively applied everywhere.

We explain here how to manage the edges between A and B, i.e. $S' \cap (A \times B)$. For each edge $e \in S' \cap (A \times B)$ we note $e = \{a, b\}$ with $a \in A$ and $b \in B$, if a or b already exists we use instead respectively $a', a'', ...$ or $b', b'', ...$. In the following rules the degree and neighborhood are considered with respect to the edge set $E \cup S'$. However we can only modify edges in S'.

Rule 1.1: If there is $e = \{a, b\} \in S' \cap (A \times B)$ such that both a has at least $\gamma(|A| - 1)$ neighbors in A and b has at least $\gamma(|B| - 1)$ neighbors in B. We can remove the edge e as shown in Figure 2.

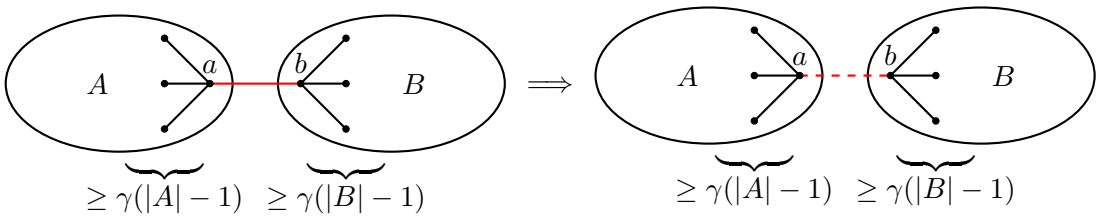


Figure 2: Application of Rule 1.1.

Rule 1.2: If there is $e = \{a, b\} \in S' \cap (A \times B)$ such that a has at least $\gamma(|A| - 1)$ neighbors in A and b has a non-neighbor b' in B, then we can remove $\{a, b\}$ from S' and add the edge $\{b, b'\}$ to S' as shown in Figure 3.

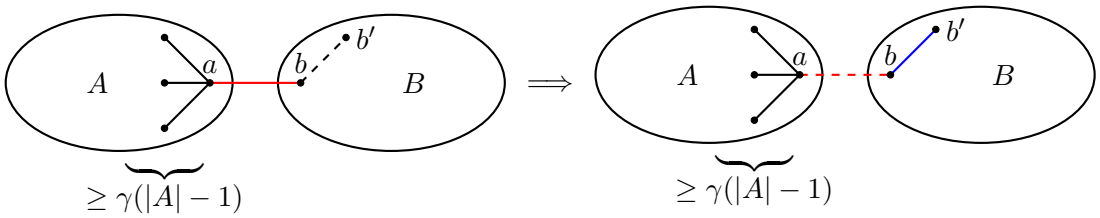


Figure 3: Application of Rule 1.2.

Rule 1.3: If there is $e = \{a, b\} \in S' \cap (A \times B)$ such that b has at least $\gamma(|B| - 1)$ neighbors in B and a has a non-neighbor in A, we proceed similarly.

We assume now that Rules 1.1, 1.2 and 1.3 cannot be applied anywhere. Note that when applied these rules decrease the degree of a vertex $a \in A$ only if $d_A(a) \geq \gamma(|A| - 1)$. Hence, after applying these rules if there is $a \in A$ such that $d_A(a) < \gamma(|A| - 1)$, then the total degree of a has not changed and is still greater than $\gamma(|A| + |B| - 1)$. A similar point holds for the vertices of B. Also, every edge $e = \{a, b\} \in S' \cap (A \times B)$ verifies that a has strictly less than $\gamma(|A| - 1)$ neighbors in A and b has strictly less than $\gamma(|B| - 1)$ neighbors in B otherwise one of the three previous rules could be applied. Hence, for such an edge $e = \{a, b\} \in S' \cap (A \times B)$ it holds that $d(a) \geq \gamma(|A| + |B| - 1)$ and $d_A(a) < \gamma(|A| - 1)$, thus $d_B(a) > \gamma|B|$ and similarly $d_A(b) > \gamma|A|$.

Rule 1.4: If there are $e, e' \in S' \cap (A \times B)$, $e = \{a, b\}$ and $e' = \{a', b'\}$ such that $\{a, a'\}$ and $\{b, b'\}$ are not in $E \cup S'$. Then, we can remove the edges $\{a, b\}, \{a', b'\}$ from S' and add instead the edges $\{a, a'\}, \{b, b'\}$ as shown in Figure 4.

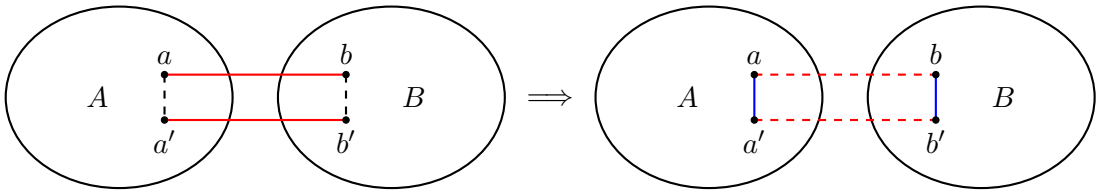


Figure 4: Application of Rule 1.4.

Rule 1.5: If there are a, b, b' such that $e = \{a, b\} \in S'$, $e' = \{a, b'\} \in S'$, $\{b, b'\} \notin E \cup S'$.

First note that all neighbors of b' in A except a itself are necessarily neighbors of a. Indeed, let $a' \in A$ be a neighbor of b' i.e. $\{a', b'\} \in S'$, if $\{a, a'\} \notin E \cup S'$ then Rule 1.4 could be applied to the edges $\{a, b\}, \{a', b'\}$.

So, $N_A(b') \setminus \{a\} \subseteq N_A(a)$. It holds that: $|N_A(b')| > \gamma|A|$. Hence, $|N_A(a)| \geq |N_A(b') \setminus \{a\}| > \gamma|A| - 1$. Since we want that $d_A(a) \geq \gamma(|A| - 1)$, it holds that a is only missing one neighbor in A. Then, instead of adding the edges $\{a, b\}$ and $\{a, b'\}$, we can add the edge $\{a, a'\}$ for some a' non-neighbor of a and add the edge $\{b, b'\}$ as done in Figure 5.

Rule 1.6: If a similar situation happens for b we proceed similarly.

We assume now that if there are a, b, b' (resp. a, a', b) such that $\{a, b\}, \{a, b'\} \in S'$ (resp. $\{a, b\}, \{a', b\} \in S'$) then $\{b, b'\} \in E \cup S'$ (resp. $\{a, a'\} \in E \cup S'$).
So for all $e = \{a, b\} \in S' \cap (A \times B)$ it holds that a has at least $\gamma|B|$ neighbors in B and they all are neighbors of b, except b itself. Hence b has at least $\gamma|B| - 1$ neighbors in B. Similarly, a has at least $\gamma|A| - 1$ neighbors in A. Also, since we assumed that a has strictly less than $\gamma(|A| - 1)$ neighbors in A, it holds that $N_A(a) \cup \{a\} = N_A(b)$. Similarly, $N_B(b) \cup \{b\} = N_B(a)$. In fact, a and b are true twins.

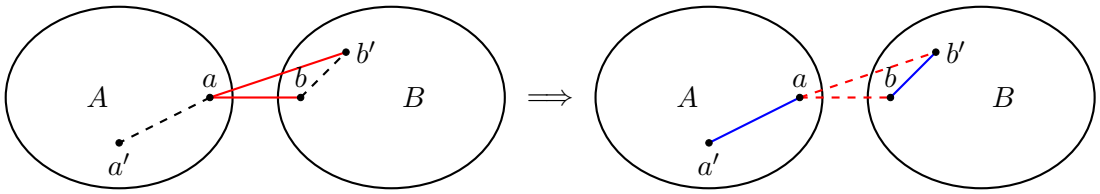


Figure 5: Application of Rule 1.5.

Rule 1.7: Finally for any edge $e = \{a, b\} \in S' \cap (A \times B)$, let a', b' be two neighbors of both a and b. Such neighbors always exist since $d_B(a) \geq \gamma|B| > 1$ and $d_A(b) \geq \gamma|A| > 1$. It holds that $\{a, a', b, b'\}$ is a clique and a, a' are only missing one neighbor in A, i.e. $d_A(a) = d_A(a') = \lceil \gamma(|A| - 1) \rceil - 1$ and b, b' are only missing one neighbor in B. Hence, we can delete the edges $\{a, b\}, \{a', b\}, \{a, b'\}, \{a', b'\}$ from S' and add instead edges as shown in Figure 6. Note that a, a' (resp. b, b') always have at least one non-neighbor in A (resp. B) to add an edge with. Indeed, their degrees in A (resp. in B) are strictly lower than $|A| - 1$.

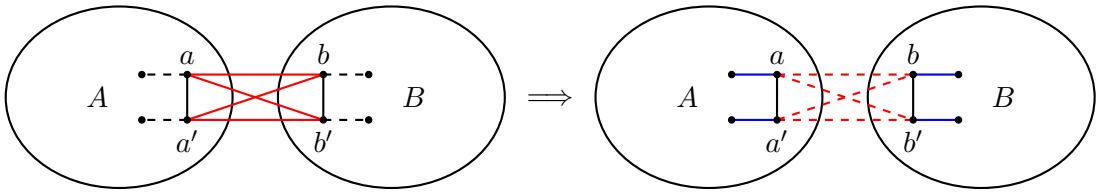


Figure 6: Application of Rule 1.7.

After applying all the rules, $S' \cap (A \times B) = \emptyset$. Also, for each rule the number of edge deleted is always greater or equal to the number of edges added, thus $|S'| \leq |S|$. Finally, for $a \in A$, after applying any rule either $d(a)$ increases or stays the same and thus stays greater than its original value $d_{G \cup S}(a) \geq \gamma(|A| + |B| - 1)$, or $d_A(a) \geq \gamma(|A| - 1)$ and a similar property holds for $b \in B$. So, $G \cup S'$ is a γ-cluster, S' is an optimal solution to the γ-Cluster Completion problem on G and S' does not add any edge between A and B.

To conclude the proof for all possible A and B, note that if, for instance, $|A| = 1$ then the only $a \in A$ has already more than $\gamma(|A| - 1) = 0$ neighbors in A and Rule 1.1 or 1.2 could be applied as long as there are edges between A and B. So the results holds even if $|A| = 1$ or $|B| = 1$. □

More generally the previous lemma tells us that there exists an optimal solution of the γ-Cluster Completion problem which does not add any edge between the connected components of the input graph and thus we can solve independently on the connected components.

Corollary 11. *Given a graph $G = (V, E)$, let $CC_1, ..., CC_r$ be the connected components of G. There exists S a solution of the γ-Cluster Completion problem on G such that S does not add any edge between the CC_is.*

Proof. (Corollary 11) We provide a proof by induction on r.

- If $r = 1$: the result holds trivially.
- If $r = 2$: the result is exactly Lemma 10.
- $r - 1 \implies r$: Let S be an optimal solution of the γ-Cluster Completion problem on G. We assume that S contains an edge between two connected components of G let's consider $e \in S$ such that $e \in CC_{r-1} \times CC_r$.

By optimality of S, it holds that $S \setminus \{e\}$ is an optimal solution to the γ-Cluster Completion problem on $G \cup \{e\} = (V, E \cup \{e\})$.

By induction and since $G \cup \{e\}$ has exactly $r-1$ connected components: CC_1, ..., CC_{r-2}, $CC_{r-1} \cup CC_r$, it holds that there is S' an optimal solution which does not add any edge between the connected components of $G \cup \{e\}$. Since $|S'| = |S \setminus \{e\}|$, it holds that $|S' \cup \{e\}| = |S|$ and $S' \cup \{e\}$ is an optimal solution to the γ-Cluster Completion problem on G. Also, $S' \cup \{e\}$ does not add any edges between $CC_{r-1} \cup CC_r$ and the rest of the graph. Let $S' \cup \{e\} = S'_1 \cup S'_2$ where $S'_1 = (S' \cup \{e\}) \cap \binom{CC_{r-1} \cup CC_r}{2}$ is the restriction of this solution to $G[CC_{r-1} \cup CC_r]$ and S'_2 contains the other edges of $S' \cup \{e\}$. It holds that S'_1 is an optimal solution to the γ-Cluster Completion problem on $G[CC_{r-1} \cup CC_r]$. By applying Lemma 10 to $G[CC_{r-1} \cup CC_r]$ it holds that there is S''_1 an optimal solution on $G[CC_{r-1} \cup CC_r]$ which does not add any edge between CC_{r-1} and CC_r. Thus, it holds that $S'' = S''_1 \cup S'_2$ is an optimal solution to the γ-Cluster Completion problem on G which does not add any edge between the connected components of G.

The corollary holds for any $r \geq 1$. □

Moreover this corollary also holds for the γ-Cluster Editing problem.

Corollary 12. *Given a graph $G = (V, E)$, let $CC_1, ..., CC_r$ be the connected components of G. There exists S a solution of the γ-Cluster Editing problem on G such that S does not add any edge between the CC_is.*

Proof. (Corollary 12) Let S be an optimal solution of the γ-Cluster Editing problem on G. We denote by S^- the set of edge deletions, i.e. $S^- = S \cap E$ and by S^+ the set of edge additions, i.e. $S^+ = S \setminus S^-$. Let $G' = (V, E \setminus S^-)$, it holds that S^+ is an optimal solution of the γ-Cluster Editing problem on G'. Since it only adds edges, it is also an optimal solution of the γ-Cluster Completion problem on G'.

Thanks to Corollary 11 we know that there is S_2^+ an optimal solution of the γ-Cluster Completion problem on G' such that S_2^+ does not add any edge between the connected components of G'. Note that the connected components of G' are always contained in the connected components of G, since edges have only been deleted, thus S_2^+ does not add any edges between the connected components of G.

Finally, $S_2 = S^- \cup S_2^+$ is an optimal solution of the γ-Cluster Editing problem on G which does not add any edge between its connected components. □

Going back to the γ-Cluster Completion problem, it holds that if the input graph is connected then the solution graph contains only one γ-quasi-clique. Hence, the γ-Cluster Completion problem on a connected graph can be reformulated as follows.

Problem 13 (connected γ-Cluster Completion)**.** Given $G = (V, E)$ a connected graph and $k \in \mathbb{Z}_{\geq 0}$, does it exist a set $S \subseteq \binom{V}{2}$ such that $|S| \leq k$ and, with $G \cup S = (V, E \cup S)$, $d_{G \cup S}(v) \geq \gamma(|V| - 1)$ for any vertex v ?

Lemma 14. *The γ − Cluster Completion problem is solvable in polynomial time on connected graphs.*

Proof. (Lemma 14) This result is a consequence of [18]. They introduced the following problem:

Problem 15. Given $G = (V, E)$, $n = |V|$, $V = \{v_1, ..., v_n\}$, $b_1, ..., b_n \in \mathbb{Z}_{\geq 0}$, find $E' \subseteq E$ such that E' is maximal and, with $G' = (V, E')$, $d_{G'}(v_i) \leq b_i$ for all $1 \leq i \leq n$.

In [18], the authors proved that this problem is solvable in polynomial time. Solving the γ − Cluster Completion problem on a connected graph G is equivalent to solving this problem with $b_i = n - 1 - \lceil \gamma(n-1) \rceil$ and the complement graph of G : $\overline{G} = (V, \overline{E})$, with $\overline{E} = \binom{V}{2} \setminus E$. Then, an optimal S can be deduced from an optimal E' by : $S = \overline{E} \setminus E'$. □

Remark 16. Note that, in [18], the fact that G is connected is never used in the proof. In fact we obtain a polynomial algorithm able to find, given a graph $G = (V, E)$, a minimal set $S \subseteq \binom{V}{2}$ such that $G \cup S = (V, E \cup S)$ is a γ-quasi-clique.

Theorem 9 is a direct consequence of Corollary 11 and Lemma 14. Also, thanks to this theorem the following corollary holds. □

Corollary 17. *Given a graph $G = (V, E)$ and a partition of V: $C_1, ..., C_r$, it is possible to compute in polynomial time a minimal set $S \subseteq \binom{V}{2}$, such that the connected components of $G' = (V, E \Delta S)$ are $C_1, ..., C_r$ and are γ-quasi-cliques.*

For $G' = (V, E \setminus S)$ and $G' = (V, E \cup S)$ such an S may not exists. The existence of such S can also be decided in polynomial time and if it exists, an optimal S can be computed in polynomial time.

Proof. (Corollary 17) For $G' = (V, E \Delta S)$, to compute an optimal S, one must first remove all edges between C_i and C_j for any $i < j$, then one can use Remark 16 on the C_is. For $G' = (V, E \setminus S)$ only the first step is necessary, then one have to check whether the C_is are γ-quasi-cliques. For $G' = (V, E \cup S)$ only the second step is necessary, then one have to check whether the C_is are disconnected from one another. □

3.2 γ-Cluster Deletion and γ-Cluster Editing

In this subsection we prove the following theorem.

Theorem 18. *Given $\gamma \in]\frac{1}{2}, 1[$, γ-Cluster Editing is NP-hard.*

We provide a reduction from Clique restricted to regular graphs which is NP-complete [8]. Let $(G = (V, E), k)$ be an instance of the Clique problem such that G is d-regular. Note that finding a clique in G of size k is equivalent to finding $S \subseteq E$ such that when removed, S separates G into a clique of size k and a set of $n - k$ vertices.

The detailed construction of the equivalent instance is described later in this section. We want to give first an intuitive approach of the proof. The equivalent instance $(G' = (V', E'), k')$ of the $\gamma-$Cluster Editing problem is constructed as follows: we add two sets of new vertices X and Y as described in Figure 7.

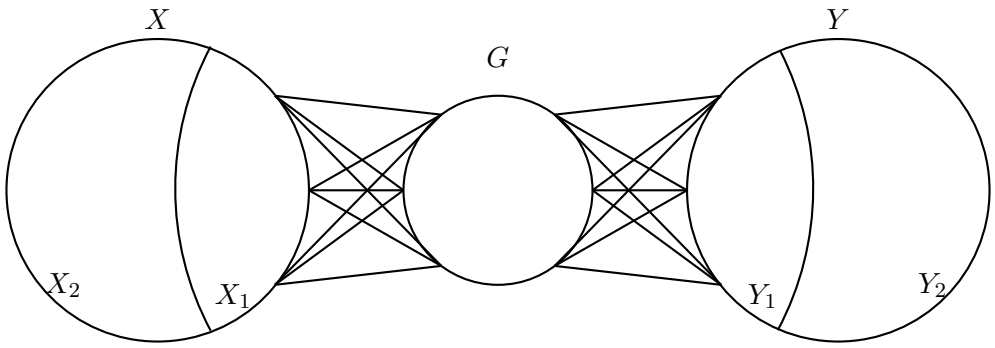


Figure 7: Construction of G'.

The idea of the proof being that X will "attract" a clique of size k in G and Y will attract the rest of the vertices of G. We first want to show here how to construct such X and Y. The

number of authorized editions in the equivalent γ-Cluster Editing instance is:

$$k' = n_{X_1}(n-k) + k(d-k+1) + n_{Y_1}k.$$

Here $n_{Y_1}k$ represents the number of edges between Y and K the clique of G of size k. $n_{X_1}(n-k)$ represents the number of edges between X and $V \setminus K$ and $k(d-k+1)$ represents the number of edges between K and $V \setminus K$. The following lemma explains why we choose to restrict the Clique problem on regular graphs.

Lemma 19. *Given $G = (V, E)$ a d-regular graph and $K \subseteq V$ of size $k \leq d$. It holds that $|E(K, V \setminus K)| \leq k(d-k+1)$ if, and only if, K is a clique of G. In that case, $|E(K, V \setminus K)| = k(d-k+1)$.*

Proof. (Lemma 19) The vertices in K have d neighbors in total and at most $k-1$ of them are also in K, so: $|E(K, V \setminus K)| \geq k(d-k+1)$. The equality holds if, and only if, every vertex in K has exactly $k-1$ neighbors in K, i.e. if K is a clique. □

Let's start the formal construction of $G' = (V', E')$. For some widgets we use a lot of regular graphs. Since given two odds numbers n and d we cannot construct a d-regular graph on n vertices, we instead use an almost d-regular graph, i.e. a graph where every vertex has degree d except one of them whose degree is $d+1$. One can obtain a d-regular graph or an almost d-regular graph via the following construction.

Construction 20. Given n and $d < n-1$, let's see how to construct a d-regular graph, or an almost d-regular. We start from an empty graph on n vertices and we label them $v_0, ..., v_{n-1}$ and we proceed as follows. See Figure 8 for an illustration.

1. We add an edge between each v_i and the $\lfloor d/2 \rfloor$ previous and next vertices in the cyclic order $v_0, ..., v_{n-1}$.

   Now G is a d-regular graph if d is even, and $(d-1)$-regular if d is odd.

2. If d is odd. For i in $[\![0, \lfloor \frac{n}{2} \rfloor]\!]$ we add an edge between v_i and $v_{i+\lfloor \frac{n}{2} \rfloor}$.

   If n is even, G is a d-regular graph.

   If n is odd, every vertex has degree d except v_{n-1} whose degree is $d-1$.

3. If both d and n are odd, we now add an edge between v_{n-1} and a random vertex.

   Now each vertex in V has degree d except one of them whose degree is $d+1$.

We call a graph construct this way an (n, d)-*dreamcatcher* graph.

Remark 21. Given $\gamma \in]\frac{1}{2}, 1[$ for all integer n, at least one of the following holds:

- $\lceil \gamma n \rceil < \lceil \gamma(n+1) \rceil$.
- $\lceil \gamma(n+1) \rceil < \lceil \gamma(n+2) \rceil$.

We present here a general widget used in the construction of G', allowing us to construct the sets X and Y such that X attracts necessarily a clique of size K, and Y attracts the other vertices of G.

Lemma 22. *Given a graph $G = (V, E)$ three integers k, d_K and N one can construct a set X of new vertices, i.e. $X \cap V = \emptyset$, such that: $|X| > N$, for $K \subseteq V$, $X \cup K$ is a γ-quasi-clique $\implies |K| \leq k$ and if $|K| = k$, $\delta(G[K]) \geq d_K$.*

Proof. (Lemma 22) Let G, k, d_K and N be as in the Lemma. We note $n = |V(G)|$. Let n_X be such that, $n_X > N$, $(1-\gamma)n_X > n$ and $\lceil \gamma(n_X + k - 1) \rceil < \lceil \gamma(n_X + k) \rceil$ (thanks to the Remark 21 such an n_X always exist). Let $d_X = \lceil \gamma(n_X + k - 1) \rceil$, $n_{X_1} = d_X - d_K$ and $n_{X_2} = n_X - n_{X_1}$.

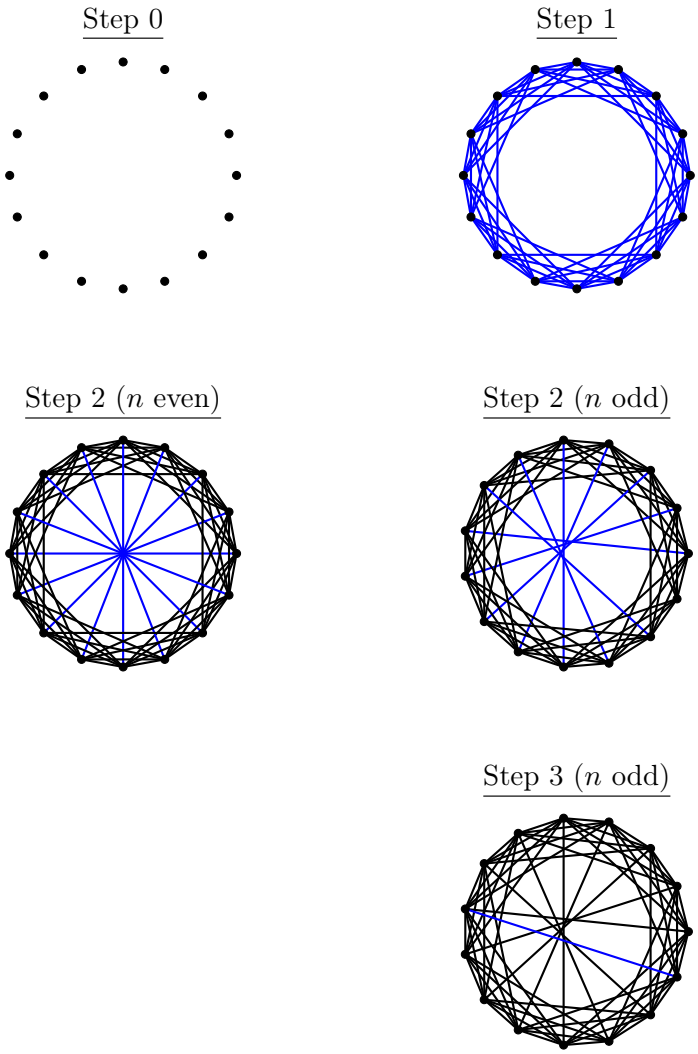


Figure 8: Steps of the construction of an (n, d)-dreamcatcher graph.

First we construct X, an (n_X, d_X)-dreamcatcher we note $X = \{x_1, ..., x_n\}$. Let $X_1 = \{x_1, ..., x_{n_{X_1}}\}$ and $X_2 = X \setminus X_1$. Second we add all possible edges between X_1 and V as done in Figure 9.

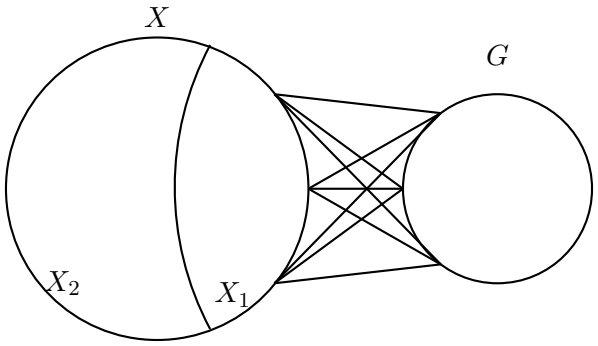


Figure 9: Construction of the widget.

Let's $K \subseteq V$ such that $X \cup K$ is a γ-quasi-clique. First, the vertices in X_2 (or all of them except one) have exactly $d_X = \lceil \gamma(n_X + k - 1) \rceil$ neighbors in $X \cup K$, since $d_X < \gamma(n_X + k)$ and $X \cup K$ is a γ-quasi-clique it holds that $|X \cup K| \leq n_X + k$ and thus, $|K| \leq k$. Also, if $|K| = k$, then the vertices in K have exactly $n_{X_1} = \lceil \gamma(n_X + k - 1) \rceil - d_K$ neighbors in X, hence they must have at least d_K neighbors in K. □

This construction will later be used in the reduction so it is important to note that n_X can be chosen such that $n_X < \max(N, \frac{1}{1-\gamma} n) + 2$, and thus is polynomial in N, n.

Remark 23. Note that for $K \subseteq V$, $|K| > k$, to transform $X \cup K$ into a γ-quasi-clique one must add at least one neighbor to any element of X_2 (except for maybe one of them). Hence, one must add at least $\frac{n_{X_2} - 1}{2}$ edges.

Remark 24. Finally, if n_{X_1} is a given value and is large enough, i.e. such that $n_{X_1} > \gamma(N + k) - d_K$. Then one can construct the same set of vertices with exactly n_{X_1} vertices in X_1 by choosing $n_X = \lfloor \frac{1}{\gamma} (n_{X_1} + d_K) \rfloor - k + 1$, or this value plus one, so that n_X verifies $\lceil \gamma(n_X + k - 1) \rceil < \lceil \gamma(n_X + k) \rceil$.

We want to ensure that there is a clique of size k in G if, and only if, there is $S \subseteq E'$ such that $|S| \leq k'$ and the connected components of $(V', E' \Delta S)$ are γ-quasi-cliques. Actually we ensure that there is a clique K of size k in G if, and only if, $X \cup K$ and $Y \cup (V \setminus K)$ are both γ-quasi-cliques and there are at most k' edges between them.

Let's start the formal construction of G'.

Construction 25 (G', k')**.** We start with $G' = (V', E')$ where $V' = V$ and $E' = E$ (a copy of G). We note $n = |V|$. Let

$$N = \left\lceil \frac{2}{1 - \gamma} \max \left((n+1)^2, \frac{1}{\gamma - \frac{1}{2}} n \right) \right\rceil + 2n$$

and $N' = N + \lceil \frac{1}{\gamma} n \rceil + n$.

The purpose of N and N' is described later. The intuitive idea is that N and N' are polynomial in n and $N >> n$, so any inequality involving N and n will always favor N.

1. Using Lemma 22 with $G, k, d_K = k - 1$ and N', we create a set of n_X vertices such that $n_X > N'$, and for $K \subseteq V$, $X \cup K$ is a γ-quasi-clique implies that $|K| \leq k$ and if $|K| = k$ then $\delta(G[K]) \geq k - 1$. We use the notations $X_1, X_2, d_X, n_{X_1}, n_{X_2}$ given in the Lemma.

2. Using Lemma 22 and Remark 24 with $G, n-k, d_K = 0$ and n_{X_1} given, we create a set of n_Y vertices such that $n_{Y_1} = n_{X_1}$ and for $K' \subseteq V$, $Y \cup K'$ is a γ-quasi-clique implies that $|K'| \leq n-k$. We also use the notations $Y_1, Y_2, d_Y, n_{Y_1}, n_{Y_2}$ for respectively, the vertices of Y linked to V, the other vertices of Y, i.e. $Y \setminus Y_1$, the degree inside Y of the vertices in Y (except maybe one of them), the number of vertices in Y_1 and the number Y_2.

3. The number of authorized editions is

$$\begin{aligned} k' &= n_{X_1}k + n_{Y_1}(n-k) + k(d-k+1) \\ &= n_{X_1}n + k(d-k+1). \end{aligned}$$

See Figure 7 for an illustration of the construction of G'.

To ensure that the edge editions are properly placed we need a few inequalities involving the values of n, N, N' and k'. These inequalities are actually what motivated the values of N and N'.

Lemma 26. *It holds that:*

1. $d_X > 2(n+1)^2$.
2. $n_{X_1} > 2(n+1)^2$.
3. $n_{X_2} > 2(n+1)^2$.
4. $n_X > 4(n+1)^2$.
5. $\frac{1}{2}(\gamma - \frac{1}{2})n_X > n$.
6. *Similar points hold for* Y.
7. $\frac{1}{2}\left(\gamma(n_X + n_Y - 2n - 1) - (d_X + n)\right)(n_X - n) > k'$.

Proof. (Lemma 26)
Note first that:

- $\gamma > \frac{1}{2} > 1 - \gamma$.
- $d_X \geq \gamma(n_X + k - 1) \geq \gamma N$.
- $n_{X_1} \geq \gamma n_X - n \geq \gamma N - n$.
- $n_{X_2} \geq (1-\gamma)n_X \geq (1-\gamma)N$.

1. $N > \frac{1}{\gamma}2(n+1)^2$, so $d_X > 2(n+1)^2$.
2. $N > \frac{1}{\gamma}2(n+1)^2 + n$, so $n_{X_1} > 2(n+1)^2$.
3. $N > \frac{1}{1-\gamma}2(n+1)^2$, so $n_{X_2} > 2(n+1)^2$.
4. $n_X = n_{X_1} + n_{X_2} > 4(n+1)^2$.
5. $N > \frac{2}{\gamma - \frac{1}{2}}n$, so $\frac{1}{2}(\gamma - \frac{1}{2})n_X > n$.
6. The previous points are all consequences of $n_X > N$. By construction, $n_Y \geq \lfloor \frac{1}{\gamma}(n_{X_1}) \rfloor - n + k + 1$. Since $n_{X_1} \geq \gamma N' - n \geq \gamma(N+n)$, it holds that $n_Y \geq N$ and the same properties holds for n_Y, d_Y, n_{Y_1} and n_{Y_2}.

7. $\gamma(n_X + n_Y - 2n - 1) - (d_X + n) \geq \frac{1}{2}\gamma n_Y - 3n - 1 > n + 1$ and $(n+1)(n_X - n) = (n+1)n_X - n(n+1)$. So,

$$\begin{aligned}\frac{1}{2}\left(\gamma(n_X + n_Y - 2n - 1) - (d_X + n)\right)(n_X - n) - k' \\ \geq \quad (n+1)n_X - (n+1)^2 - nn_{X_1} - (n+1)^2 \\ \geq \quad n_X - 2(n+1)^2 > 0.\end{aligned}$$

□

We are now ready to prove Theorem 18.

Proof. (Theorem 18) Since N' is polynomial in n, so is the size of G'. Hence, in order to prove that the γ-Cluster Editing problem is NP-hard we only have to prove that:

G has a clique of size $k \iff (G', k')$ is a yes-instance of the γ-Cluster Editing problem.

$\Longrightarrow$ Let K be a clique of G of size k. Let's consider: $Q_X = X \cup K$ and $Q_Y = Y \cup (V \setminus K)$.

- Q_X is a γ-quasi-clique. Indeed, $|Q_X| = n_X + k$, for all $x \in X$, $d_{Q_X}(x) \geq d_X \geq \gamma(n_X + k - 1)$ and for all $v \in K$, $d_{Q_X}(v) = d_X(v) + d_K(v) = \gamma(n_X + k - 1) - k + 1 + k - 1$.
- Similarly Q_Y is a γ-quasi-clique.
- There are $n_{X_1}(n-k)$ edges between X and $(V \setminus K)$. Thanks to Lemma 19 there are $k(d - k + 1)$ edges between K and $(V \setminus K)$. There are $n_{Y_1}k$ edges between Y and K. Hence there are excatly $n_{X_1}n + k(d - k + 1) = k'$ edges between Q_X and Q_Y.

Hence, Q_X and Q_Y are γ-quasi-cliques and can be disconnected by removing k' edges. Thus, (G', k') is a yes-instance of the γ-Cluster Editing problem.

$\Longleftarrow$ Let $S \subseteq \binom{V'}{2}$, such that $|S| \leq k'$ and the connected components of $G'' = (V', E'\Delta S)$ are γ-quasi-cliques.

We decompose this part of the proof into several claims. We postpone the proofs of these claims after the last one of them.

Claim 27. *At least $n_X - n$ (resp. $n_Y - n$) elements of X (resp. Y) are connected to each other in G''.*

We note X_A (resp. Y_A) the set of at least $n_X - n$ (resp. $n_Y - n$) vertices of X (resp. Y) connected to each other in G'' and $X_B = X \setminus X_A$ (resp. $Y_B = Y \setminus Y_A$).

Claim 28. *Let Q be a connected component of G'', Q contains at most one of the following sets: X_A, Y_A.*

Claim 29. *For all $v \in V$, v is either connected to X_A or to Y_A in G''.*

Claim 30. *X (resp. Y) is connected in G''.*

We note Q_X (resp. Q_Y) the connected of G'' containing X (resp. Y).

Claim 31. *Q_X contains X and at most k other vertices. Q_Y contains Y and at most $n - k$ other vertices.*

Claim 32. *$K = Q_X \cap V$ is a clique of G of size k.*

Proof. (Claim 27) Let's assume that $G''[X]$ is not connected. Let (A, B) be a partition of X such that $|A| \geq \frac{1}{2}n_X \geq |B|$, and $E_{G''}(A, B) = \emptyset$. Since there are no edges between A and B in G'', S contain all edges of $E_{G'}(A, B)$. Also, since $d_X > n_X/2 \geq |B|$, each vertex of B is linked to at least $d_X - |B|$ elements of A in G'. Thus there are at least $(d_X - |B|)|B|$ edges between A and B in G'.

Let $f : x \in [0, \frac{n_X}{2}] \mapsto (d_X - x)x$, it holds that for all $x \in [n+1, \frac{n_X}{2}]$, $f(x) > k'$.

Indeed, f increases on $[0, \frac{d_X}{2}]$ and $n+1 \leq \frac{d_X}{2}$ (cf. Lemma 26.1) and $f(n+1) = (d_X - n - 1)(n+1)$.

$$\begin{aligned} f(n+1) - k' &= d_X(n+1) - (n+1)^2 - n_{X_1}n - k(d-k+1) \\ &\geq (n_{X_1} + k - 1)(n+1) - (n+1)^2 - n_{X_1}n - k(d-k+1) \\ &\geq n_{X_1} + (k-1)(n+1) - (n+1)^2 - k(d-k+1) \\ &\geq n_{X_1} - (n+1)^2 - k(d-k+1) \\ &> n_{X_1} - 2(n+1)^2 > 0. \qquad \text{(cf. Lemma 26.2)} \end{aligned}$$

So $f(n+1) > k'$.

And f decreases on $[\frac{d_X}{2}, \frac{n_X}{2}]$ and $f(\frac{n_X}{2}) = (d_X - \frac{n_X}{2})\frac{n_X}{2}$.

$$\begin{aligned} f(\frac{n_X}{2}) - k' &= (d_X - \frac{n_X}{2})\frac{n_X}{2} - k' \\ &\geq (\gamma(n_X + k - 1) - \frac{n_X}{2})\frac{n_X}{2} - k' \\ &\geq \frac{1}{2}(\gamma - \frac{1}{2})n_X^2 - k' \\ &\geq n \cdot n_X - n \cdot n_{X_1} - k(d-k+1) \\ &> n \cdot n_{X_2} - n^2 > 0. \qquad \text{(cf. Lemma 26.3)} \end{aligned}$$

So $f(\frac{n_X}{2}) > k'$.

Since $|S| \leq k'$, it holds that $|B| \leq n$.

Let's prove that there is a connected component of $G''[X]$ containing at least half of the vertices of X. Assuming otherwise let $C_1, ..., C_r$ be the connected components of $G''[X]$ (we assume that $|C_1| \geq ... \geq |C_r|$).

- If $|C_1| \leq \frac{n_X}{4}$. Let $B = C_1 \cup ... \cup C_i$ such that $|B| \leq \frac{n_X}{2}$ and $|B \cup C_{i+1}| \geq \frac{n_X}{2}$. Let $A = X \setminus B$. Then it holds that (A, B) is a partition of X, with $|A| \geq \frac{1}{2}n_X \geq |B|$, no edges between A and B in G'', and $|B| \geq \frac{n_X}{4} \geq n+1$ (cf. Lemma 26.4) which is impossible.
- If $\frac{n_X}{4} \leq |C_1| \leq \frac{n_X}{2}$. Let $B = C_1$ and $A = X \setminus B$. This case is impossible for the same reasons.
- Hence $|C_1| \geq \frac{n_X}{2}$.

Hence it holds that for (A, B) a partition of X such that $E_{G''}(A, B) = \emptyset$ and $|A| \geq n_X/2 \geq |B|$: A is connected and $|B| \leq n$. □

Proof. (Claim 28) Let's assume that there is Q a connected component of G'' containing X_A and Y_A. Let $x \in X_A$, x has at most $d_X + n$ neighbors in Q when considering only the edges of E'. Hence when editing with S we must at least add $\gamma(|X_A| + |Y_A| - 1) - (d_X + n)$ neighbors to x. Hence we must add at least $\frac{1}{2}\left(\gamma(n_X + n_Y - 2n - 1) - (d_X + n)\right)(n_X - n)$ edges to G' which is impossible since this number is stricly greater than k' (cf. Lemma 26.7).

Hence Q cannot contain both X_A and Y_A. □

Proof. (Claim 29) Let $v \in V$, v cannot be connected to both X_A and Y_A in G'' otherwise they would also be connected. Hence either all edges between v and X_A are removed when editing with S, or all edges between v and Y_A are removed. In both cases at least $n_{X_1} - n$ edges are removed, leading to a total of $(n_{X_1} - n)n$ removed edges when taking into account all vertices of V.

Also, if there is $v \in V$ which is not connected to X_A nor to Y_A in G'' then an extra $n_{X_1} - n$ edges must be deleted. Thus the total number of edges deleted is at least:

$$(n_{X_1} - n)\,(n+1) > k'.$$

Hence, any $v \in V$ is either connected to X_A or to Y_A in G''. □

Proof. (Claim 30) Thanks to the previous claim we know that at least $(n_{X_1} - n)\,n$ edges are removed between the vertices in V and vertices in X_A or Y_A. Hence, there are at most $k' - n_{X_1} \cdot n + n^2$ editions left in S. For all $i \in [\![1, n_X]\!]$ it holds that x_i and x_{i+1} are linked in G' and have at least $d_X - 2$ common neighbors in G'. Hence to disconnect x_i and x_{i+1} we need to remove at least $d_X - 1$ edges. Since $d_X - 1 > k' - n_{X_1} \cdot n + n^2$ (cf. Lemma 26.1) it holds that x_i and x_{i+1} are connected in G'' and thus X remains connected in G''. A similar result holds for Y. □

Proof. (Claim 31) Thanks to the two previous claims we know that for each $v \in V$ we must remove either all the edges between v and X or all the edges between v and Y. Thus, there are at least $n_{X_1} \cdot n$ edges removed this way. Note that vertices of X_2 (except for maybe one of them) have originally only $d_X < \gamma(n_X + k)$ neighbors. Hence if Q_X contains X and more than $k+1$ other vertices, it holds that each vertex in X_2 (except for maybe one of them) is originally missing at least one neighbor. Thus, when editing with S we must add at least $\frac{1}{2}(n_{X_2} - 1)$ edges which is impossible since $\frac{1}{2}(n_{X_2} - 1) > k' - n_{X_1} \cdot n$ (cf. Lemma 26.3). A similar result holds for Q_Y. □

Proof. (Claim 32) Thanks to the previous claims we know that each $v \in V$ remains connected to exactly one of the two sets: X or Y. So, exactly $n_{X_1} n$ edges have already been deleted. In order to disconnect Q_X and Q_Y we must remove all edges between $Q_X \cap V$ and $Q_Y \cap V$. Let's note $K = Q_X \cap V$, it holds that $|K| = k$. Also $Q_Y \cap V = (V \setminus K)$. There are only $k(d - k + 1)$ editions left and thanks to Lemma 19, it holds that K is a clique of G, otherwise Q_X and Q_Y would be impossible to disconnect with this number of editions. □

This concludes the proof of Theorem 18. □

It is important to note that with this construction any optimal solution only involve edge deletions. Thus, using the exact same proof, one can find that the γ-Cluster Deletion problem is also NP-hard.

Theorem 33. *Given* $\gamma \in \left]\frac{1}{2}, 1\right[$, γ-Cluster Deletion *is* NP*-hard.*

3.3 Fixed number of clusters

In this section we study a variant of the γ-clustering problems where the final number of clusters is fixed to a given constant p. We call these variants the (γ, p)-clustering problems.

First, note that the (γ, p)-clustering problems are all solvable in polynomial time when $p = 1$. Indeed, with only one cluster we can apply Corollary 17 with $r = 1$ and $C_1 = V$. Then, we already prove that the (γ, p)-Cluster Deletion and γ-Cluster Editing are

NP-complete for every $p \geq 2$. Indeed, in the proof of Theorem 18 it is clear that there are always exactly two clusters Q_X and Q_Y which proves the NP-completeness for $p = 2$. For $p \geq 2$ one only have to add $p - 2$ isolated vertices. Finally, only the complexity of the (γ, p)-Cluster Completion problem for $p \geq 2$ is still open. We prove here that these problems are all solvable in polynomial time.

Theorem 34. *For any $p \geq 1$, the (γ, p)-Cluster Completion problem can be solved in polynomial time.*

Proof. (Theorem 34) Again, since edges can only be added to the graph, a connected component always remain connected, hence if the input graph has less than p connected components there are no solutions. Thus, we assume that the input graph always has more than p connected components. As opposed to the γ-Cluster Completion problem and in order to have exactly p clusters, an optimal solution to the (γ, p)-Cluster Completion problem sometimes involve adding edges between connected components. Hence, to solve the (γ, p)-Cluster Completion problem we need to find a p-partition of the connected components of the input graph G such that the number of edge additions required to transform each part into a γ-quasi-clique is minimal.

This problem is solved using a dynamic programming algorithm. Before, describing the table we want to highlight by Lemma 35 that "most" of the connected components, i.e. except a finite number of them, can be completed into cliques.

Given $\gamma \in]\frac{1}{2}, 1[$ we denote ε_γ a non-negative real such that for all $0 \leq \varepsilon \leq \varepsilon_\gamma$ it holds that:

$$\left(\frac{\gamma\varepsilon}{1-\gamma-\varepsilon} + (1-\gamma) + \varepsilon\right) < \gamma.$$

Such an ε_γ always exists because $\gamma > \frac{1}{2}$ and $1 - \gamma < \gamma$.

First we prove in the following lemma that we can always add all possible edges inside the "small" sets of vertices, i.e. of relative size lower than ε_γ.

Lemma 35. *Let $G = (V, E)$, let $A, B \subseteq V$ such that $E(A, B) = \emptyset$, and $|B| \leq \varepsilon_\gamma |V|$. There exists $S \subseteq \binom{V}{2}$ of minimal size such that $G \cup S = (V, E \cup S)$ is a γ-quasi-clique, verifying that B is a clique of $G \cup S$.*

Proof. (Lemma 35) Let $G = (V, E), A, B, S \subseteq \binom{V}{2}$ of minimal size such that $G \cup S = (V, E \cup S)$ is a γ-quasi-clique and $d_\gamma = \lceil \gamma(|V| - 1) \rceil$ the required degree. Let $\varepsilon = \frac{|B|}{|V|} \leq \varepsilon_\gamma$. Let's construct S' another solution such that B is a clique of $G \cup S' = (V, E \cup S')$.

We start with $S' = S$ and we apply Rules 1.2 and 1.4 from the proof of Lemma 10 labeled here Rule 2.1 and Rule 2.2. We also assume that Rule 2.2 is applied only if Rule 2.1 cannot be applied anywhere. Again these rules only concerns edges of S' between A and B. These edges are denoted $e = \{a, b\}$ with $a \in A$ and $b \in B$, if either a or b already exists in the context we use instead $a', a'', ...$ and $b', b'', ...$ The degrees and neighbors are also always consider with respect to the graph $G \cup S' = (V, E \cup S')$.

Rule 2.1: If there are a, b, b' such that $\{a, b\} \in S'$ and $\{b, b'\} \notin E \cup S'$ and $d(a) \geq d_\gamma + 1$ then we can replace $\{a, b\}$ by $\{b, b'\}$ as in Figure 10.

Rule 2.2: If there are a, a' and b, b' such that $\{a, b\} \in S'$, $\{a', b'\} \in S'$, $\{a, a'\} \notin E \cup S'$, $\{b, b'\} \notin E \cup S'$. Then we can replace $\{a, b\}$ and $\{a', b'\}$ by $\{a, a'\}$ and $\{b, b'\}$ as in Figure 11.

Before applying the rules, since we start with $S' = S$, it holds that $G \cup S'$ is a γ-quasi-clique and thus the degree of vertices are all greater than d_γ. When applying the rules the degree

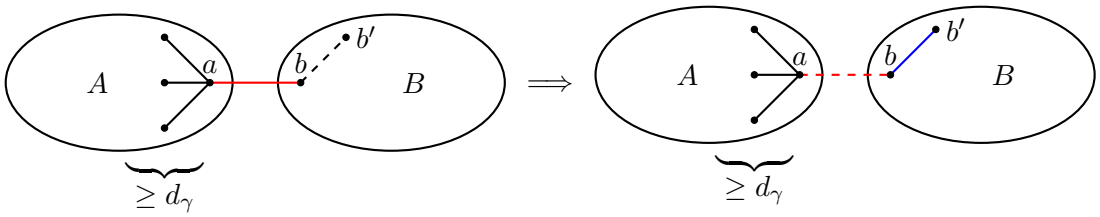


Figure 10: Application of Rule 2.1.

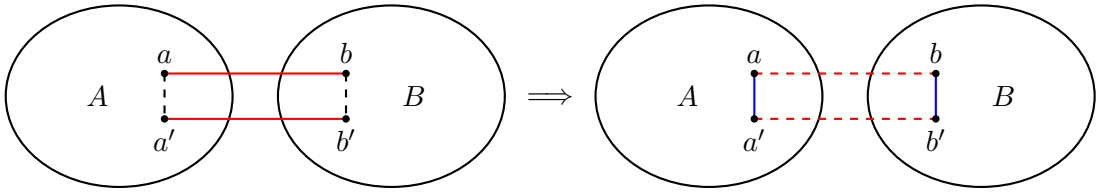


Figure 11: Application of Rule 2.2.

of each vertex either increases, stays constant or stays greater than d_γ. Hence, after applying the rules, each vertex has a degree still greater than d_γ and $G \cup S'$ is still a γ-quasi-clique.

Let's now prove that, if these rules cannot be applied then B is a clique of $G \cup S'$. Let assume the opposite, let b, b' be two vertices in B such that $\{b, b'\} \notin E \cup S'$, let $A_1 = N_A(b) \cup N_A(b')$ and $A_2 = A \setminus A_1$. We note $n_A = |A|, n_B = |B|, n_{A_1} = |A_1|$ and $n_{A_2} = |A_2|$.

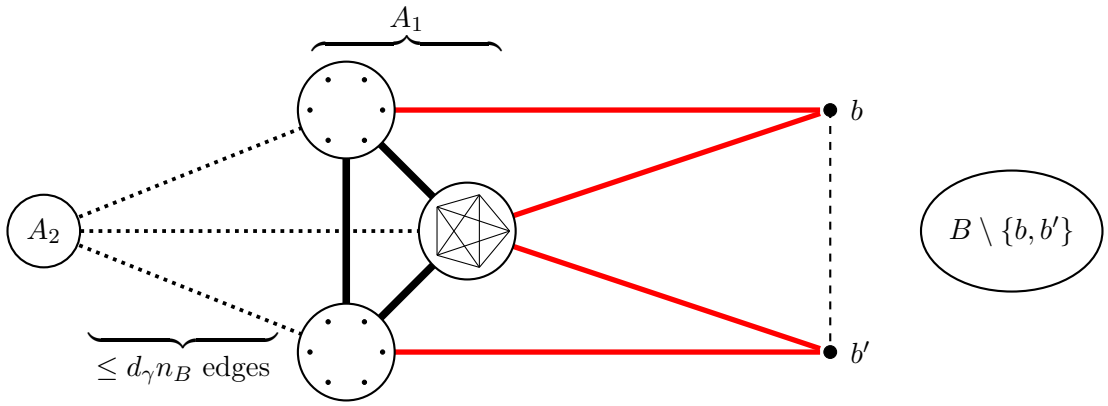


Figure 12: State of the graph after applying the rules exhaustively in the case where B is not a clique.

We present here a few useful properties that describe the state of the graph $G \cup S'$ after applying Rules 2.1 and 2.2. Figure 12 presents an overview of these properties.

- $N_A(b) \cap N_A(b') \neq \emptyset$. Indeed, thanks to Lemma 7, the number of common neighbors of b and b' is at least $(2\gamma - 1)(|V| - 1) > n_B$, so they share at least a neighbor in A.
- Let $a \in N_A(b) \cap N_A(b')$, a is linked to every other element of A_1 otherwise Rule 2 could be applied.
- $|A_1| < d_\gamma$. Indeed, if for $a \in N_A(b) \cap N_A(b')$ it holds that a has at least $|A_1| - 1 + 2$ neighbors: all A_1 except a itself, b and b'. Also, $d(a) = d_\gamma$ otherwise Rule 2.1 could be applied.
- Let $a \in A_1$, $d_{A_1}(a) \geq d_\gamma - n_B$. Indeed, if $a \in N_A(b)$, then a must be link to all $N_A(b')$ otherwise Rule 2.2 could be applied. Also, $|N_A(b')| = d(b') - d_B(b') \geq d_\gamma - n_B + 2$.

- The elements of A_1 have exactly d_γ neighbors in total otherwise Rule 2.1 could be applied. Hence they have at most n_B neighbors in A_2. Also, $|A_1| \leq d_\gamma$. Thus, there are at most $d_\gamma n_B$ edges between A_1 and A_2.
- For an element of A_2, the average number of neighbors in A_1 is lower than: $\frac{d_\gamma n_B}{n_{A_2}}$. Let $a \in A_2$ such that: $d_{A_1}(a) \leq \frac{d_\gamma n_B}{n_{A_2}}$. It holds that a has strictly less than d_γ neighbors in total. Indeed, note that:
  - $n_A = (1-\varepsilon)n$,
  - $n_{A_1} \geq d_\gamma - n_B \geq \gamma(n-1) - \varepsilon n$,
  - $n_{A_1} \leq \gamma(n-1)$,
  - $n_{A_2} = |A| - n_{A_1} \leq (1-\gamma)n + \gamma$
  - $n_{A_2} \geq (1-\gamma-\varepsilon)n$.

  So,

$$\begin{aligned} d(a) &\leq d_{A_1}(a) + d_{A_2}(a) + d_B(a) \\ &\leq \frac{d_\gamma n_B}{n_{A_2}} + n_{A_2} - 1 + n_B \\ &\leq n\left(\frac{\gamma\varepsilon}{1-\gamma-\varepsilon} + (1-\gamma) + \varepsilon\right) \\ &< \gamma n \leq d_\gamma. \end{aligned}$$

This contradicts the fact that $G \cup S'$ is a γ-quasi-clique. Hence, B must be a clique of $G \cup S'$. □

Second, using the pigeonhole principle one can see that there is a finite number of disjoint "regular" sets of vertices, i.e. not "small".

Remark 36. Given a set X, there are at most $\frac{1}{\varepsilon_\gamma}$ disjoints sets $X_1, ..., X_{\lfloor \frac{1}{\varepsilon_\gamma} \rfloor}$ such that: $X_i \subseteq X$ and $|X_i| > \varepsilon_\gamma |X|$.

Let's note $CC_1, ..., CC_r$ the connected components of G. The final γ-quasi-cliques are denoted $Q_1, ..., Q_p$, as previously said the Q_is form a partition of the CC_ℓs. Let $p_\gamma = \lfloor \frac{1}{\varepsilon_\gamma} \rfloor$.

The idea is to represent each cluster using $2p_\gamma + 1$ sets of vertices which we call bags: two for each "regular" set and one for the rest of the vertices. We want the bags to be always "small" so that they can be completed into cliques and thus be represented only by their number of vertices. For the "regular" sets of vertices we use two bags so that they can both be "small" and form a "regular" set of vertices. In total we have to consider $p(2p_\gamma + 1)$ bags labeled respectively $(1,1)$ for bag 1 of cluster 1 up to $(p, 2p_\gamma + 1)$ for bag $2p_\gamma + 1$ of cluster p.

We decompose the procedure into three algorithms. The first one is a branching algorithm used to process the connected components of relative size greater than ε_γ. It sometimes "guesses" values meaning in practice that it will create a new branch for each possibility. The second one is a dynamic programming algorithm used to process the connected components of relative size smaller or equal to ε_γ. The third one finalizes the computation.

Branching algorithm:

1. We start by guessing $|Q_1|, ..., |Q_p|$ the size of the final γ-quasi-cliques.
2. For each final γ-quasi-cliques Q_i we guess at most p_γ connected components of G contained in Q_i and of size greater than $\varepsilon_\gamma |Q_i|$.

3. For each component CC which has not been guessed during the previous step, we add (for this branch only) all possible edges inside CC. We note κ_0 the total number of edges added this way.

Now, for each leaf of the branching tree, we design a dynamic programming algorithm. This algorithm uses a table T with $p \times (2p_\gamma + 1)$ cells representing the bags. The values of the cells vary from 0 to n. Let $\mathcal{T} = [\![0, n]\!]^{p\times(2p_\gamma+1)}$ be the set of all possible tables. Let's describe $\omega : \mathcal{T} \to [\![0, n^2]\!] \cup \{\perp\}$ the weight function associating a table T with the corresponding number of edges already added to the graph.

Dynamic programming algorithm:

1. We start with $\omega^{(0)}(0_{p\times(2p_\gamma+1)}) = \kappa_0$ and $\omega^{(0)}(T) = \perp$ for all $T \in \mathcal{T} \setminus \{0_{p\times(2p_\gamma+1)}\}$, where $0_{p\times(2p_\gamma+1)}$ represents the table containing only zeros.

2. For $\ell = 1$ to r we create $\omega^{(\ell)}$ using $\omega^{(\ell-1)}$ and the following rules.

   2.1 If $CC_{(\ell)}$ is one of the components guessed at step 2 of the branch algorithm then $\omega^\ell = \omega^{\ell-1}$.

   2.2 Otherwise, for all (i, j) and all $T \in \mathcal{T}$. Let $T^{(i,j)}$ be a copy of T except for cell (i, j) which verifies $T^{(i,j)}[i, j] = T[i, j] - |CC_\ell|$. Let $\kappa_\ell^{(i,j)} = |CC_\ell| \times T^{(i,j)}[i, j]$ be the number of edge additions required to transform the union of two cliques of size $|CC_\ell|$ and $T^{(i,j)}[i, j]$ into a single clique.
   We set $\omega^{(\ell)}(T) = \min_{\substack{1\le i\le p\\ 1\le j\le 2p_\gamma+1}} \left(\omega^{(\ell-1)}(T^{(i,j)}) + \kappa_\ell^{(i,j)}\right)$.
   For the sake of clarity we only described here the general case. If some values are out of bounds: for instance $T^{(i,j)}[i, j] < 0$ or $\omega^{(\ell-1)}(T^{(i,j)}) + \kappa_\ell^{(i,j)} > n^2$, we replace the corresponding value with $\perp$. Also, when computing the minimum, we assume that $\forall q \in \mathbb{Z}_{\ge 0}, \perp > q$.

Let's now see how to compute the optimal number of edge addition given $\omega = \omega^r$.

Final checking algorithm:

1. For all table T, we must check first if T corresponds to a solution verifying the first guesses. Hence, if for some $1 \le i \le p$, $T[i, j] > \varepsilon|Q_i|$, or if $T[i, 1] + ... + T[i, 2p_\gamma + 1] \neq |Q_i| - t_i$, where t_i is the total size of the components guess for Q_i at step 2 of the Branching algorithm. Then, we update ω: $\omega(T) = \perp$.

2. Now, for all T such that $\omega(T) \neq \perp$. We create a graph G_T corresponding to the table T and the guesses, i.e. a union of the components guessed at step 2 of the Branching algorithm and a clique of size $T[i, j]$ for each cell (i, j). Using Corollary 17 we compute κ' the optimal number of edge additions required to transform the Q_is into γ-quasi-cliques. The total number of edge additions corresponding to T and the current branch is $\omega(T) + \kappa'$.

3. The optimal number of edge additions required to transform G into a γ-cluster graph with exactly p connected components is the minimal value on all the branches.

Let's now evaluate the total time-complexity of these algorithms. First, there are less than $n^p \cdot n^{(p+1)p_\gamma} = n^{O(1)}$ possibilities for the guesses. Hence, the Branching algorithm creates $n^{O(1)}$ branches. Then, the Dynamic programming algorithm processes r times $|\mathcal{T}|$ tables. Hence, this algorithm runs in time $r \times (n + 1)^{p\times(2p_\gamma+1)} = n^{O(1)}$. Finally, the Checking algorithm uses $|\mathcal{T}|$ times Corollary 17, hence it also runs in $n^{O(1)}$. Thus, the total complexity of the process is polynomial in n. □

4 Parameterized Complexity

Thanks to the previous section we know that the γ-Cluster Deletion and the γ-Cluster Editing problems are NP-complete. In order to provide an efficient algorithm for these problems, we consider them with regard to the parameterized complexity point of view. We choose k, the size of the solution as parameter. This parameter is usually the most natural one when studying optimization problems, also such parameterization has already given interesting results on the classical Cluster Deletion and Cluster Editing problems with several FPT algorithms and kernalizations [5, 10]. When parameterized by k, it is trivial that these problems are XP. Indeed the brute-force algorithm which tries every $S \subseteq \binom{V}{2}$ of size k runs in time $n^{2k+O(1)}$. In this section we provide FPT algorithms solving the γ-Cluster Deletion problem and the γ-Cluster Editing problem.

Note that when solving the γ-Cluster Deletion problem, edges are never added and thus disconnected vertices always remain disconnected. Hence, the problem can be solved independently on the connected components. The same result holds for the γ-Cluster Editing problem and is a consequence of Corollary 12.

The idea behind the algorithms for the γ-Cluster Deletion problem and the γ-Cluster Editing problem is based on Lemmas 7 and 8 presented earlier in Section 2. First using Lemma 7 we show how to restrain ourselves to graph with connected components of diameter 2. Then, using the definition of a γ-quasi-clique and Lemma 8 we show how to handle connected components of size unbounded by $f(k)$ for some function f. Since connected components of size $\leq f(k)$ can be solved in FPT time by a brute-force algorithm, all the possible cases are covered and we obtain a general FPT algorithm.

4.1 γ-Cluster Deletion

Theorem 37. *The γ-Cluster Deletion problem can be solved in time $2^{k\log(2k)} \cdot n^{O(1)}$.*

Proof. (Theorem 37) Without loss of generality we assume that $k \geq 2$. The algorithm is a search tree algorithm based on the following rules. When applying the rules we sometimes "try" to remove edges meaning that the algorithm creates a new branch for each possibility, removing the corresponding edges and updating k accordingly. Let $G = (V, E)$ be a graph and k be an integer representing the number of edge deletions.

Rule 3.1: If there exist two vertices $u, x \in V$ such that $\text{dist}_G(u, x) = 3$. Then, for any solution S of the γ-Cluster Deletion problem on G, it holds that u and x are not contained in the same γ-quasi-clique of $G \setminus S = (V, E \setminus S)$. Hence, we can try to delete each one of the three edges on the shortest path between u and x, as shown in Figure 13, and update the value of k: $k \leftarrow k - 1$.

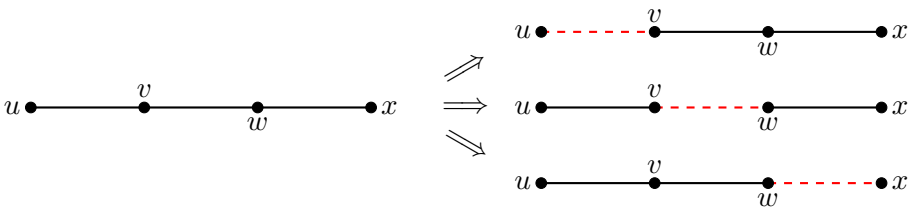


Figure 13: Application of Rule 3.1.

Note that if there exist two vertices $u, x \in V$, in the same connected component of G, such that $\text{dist}_G(u, x) > 3$. Then, on the shortest path between u and x one can find a vertex y such that $\text{dist}_G(u, y) = 3$ and Rule 3.1 could be applied. Hence, if Rule 3.1 cannot be applied then, the diameter of the connected components of G is always at most 2.

Rule 3.2: If there is $CC \subseteq V$ a connected component of G such that CC is a γ-quasi-clique then, we can remove the vertices of CC and solve the problem on $G[V \setminus CC]$.

If CC a connected component of G is not a γ-quasi-clique. Then, it holds that u, the vertex of minimal degree in CC, has a degree strictly lower than $\gamma(|CC|-1)$. Conversely, if CC is a connected component of G and u is the vertex of minimal degree in CC and $|CC| \geq \lfloor \frac{1}{\gamma} d(u) \rfloor + 2$, then CC is not a γ-quasi-clique (cf. Remark 6).

Rule 3.3: If there is CC a connected component of G and u the vertex of minimal degree in CC such that $d(u) \leq k$. Then, let $C \subseteq CC$ of size $|C| = \lfloor \frac{1}{\gamma} d(u) \rfloor + 2$ such that $u \in C$ and $G[C]$ is connected. For any solution, it holds that u cannot remain connected to all C. Hence, for T a spanning tree of $G[C]$ we can try to delete any edge of T, as shown in Figure 14, and update k: $k \leftarrow k-1$. Note that $|C| < 2k+2$, since $d(u) \leq k$ and $\gamma > \frac{1}{2}$.

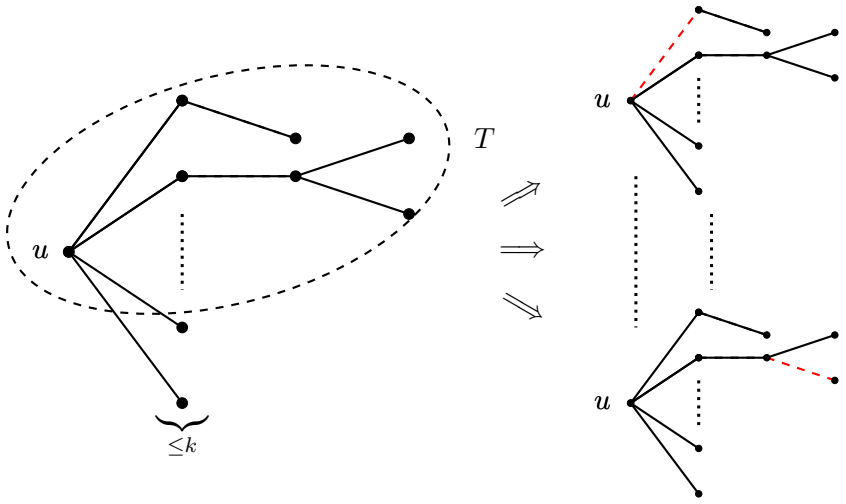


Figure 14: Application of Rule 3.3.

Let's assume that G and k are such that Rules 3.1, 3.2 and 3.3 cannot be applied. Let CC be a connected component of G. It holds that $\text{diam}(G[CC]) \leq 2$, $G[CC]$ is not a γ-quasi-clique and $\delta(G[CC]) \geq k+1$. Hence, $G[CC]$ is not a γ-quasi-clique and is $(k+1)$-edge connected thanks to Lemma 8. So even after removing k edges $G[CC]$ is connected, thus the γ-Cluster Deletion problem on G with k deletions has no solution.

Let's now evaluate the running time of this process. Assuming Rule 3.1 is executed i times and Rule 3.3 is executed j times. Since each spanning tree T has at most $2k+1$ nodes and $2k$ edges, the overall complexity can be expressed as : $3^i \cdot (2k)^j \cdot n^{O(1)}$. Also each time Rule 3.1 or 3.3 is executed an edge is deleted, hence $i+j \leq k$. Thus, this process can be executed in $2^{k \log(2k)} \cdot n^{O(1)}$. □

4.2 γ-Cluster Editing

Theorem 38. *The* $\gamma-$ Cluster Editing *problem can be solved in time* $2^{O(k \log k)} \cdot n^{O(1)}$.

Proof. (Theorem 38) The FPT algorithm used for this problem is actually very similar to the one used for the γ-Cluster Deletion problem. Again, trying to remove edges means that we create a new branch for every possibility. However, since edges can be added to G, the restriction to connected components of diameter lower than 2 cannot be done as easily and require the full use of Lemma 7.

Let $\varepsilon = \gamma - \frac{1}{2} > 0$. For any given graph G and any given Q, γ-quasi-clique of G, Lemma 7 ensures that:

$$\forall u, v \in Q, \qquad |N_Q(u) \cap N_Q(v)| \geq 2\varepsilon \ (|Q| - 1).$$

Let $G = (V, E)$ be a graph and k be an integer. The algorithm proceeds as follows.

Rule 4.1: If there exist two vertices $u, v \in V$ such that $\text{dist}_G(u, v) = 3$. Let CC be the connected component of G containing u and v.

(a) If $|CC| \leq \frac{1}{2\varepsilon}k + 1$ we can solved this component with the brute-force algorithm in $(\frac{1}{2\varepsilon}k)^{2k}$ and then removed from G.

Assuming that (a) cannot be applied, $|CC| \geq \lceil \frac{1}{2\varepsilon}k \rceil + 2$. Let S be an optimal solution.

(b) If u, v are not in the same γ-quasi-clique of $G\Delta S = (V, E\Delta S)$, then S contains one of the three edges on the shortest path between u and v.

(c) Else, u, v are in the same γ-quasi-clique of $G\Delta S$, let's note it Q. It holds that $|N_{(G\Delta S)[Q]}(u) \cap N_{(G\Delta S)[Q]}(v)| \geq 2\varepsilon \ (|Q| - 1)$ and since $\text{dist}_G(u, v) = 3$, $|N_G(u) \cap N_G(v)| = 0$. So S contains at least $2\varepsilon \ (|Q| - 1)$ edges with an endpoint being u or v. Hence, $k \geq |S| \geq 2\varepsilon \ (|Q| - 1)$ and $|Q| \leq \frac{1}{2\varepsilon}k + 1$. In this case, let $C \subseteq CC$ be a set of vertices such that $G[C]$ is connected, C contains u and $|C| = \lceil \frac{1}{2\varepsilon}k \rceil + 2$. Let T be a spanning tree of $G[C]$. C cannot remain connected in $G\Delta S$, thus S contains at least an edge of T.

We combine cases (b) and (c) by trying to remove successively one of the following edges: $\{u, w\}$, $\{w, w'\}$, $\{w', v\}$ or one of the edges of $E(T)$, where (u, w, w', v) is the shortest path between u and v. Also we update k: $k \leftarrow k - 1$.

As previously, if Rule 4.1 cannot be applied then the connected components of G all have a diameter lower than 2.

Rule 4.2: If there exists CC a connected component of G such that $|CC| \leq 4k$ then this component can be solved using the brute-force algorithm in $(4k)^{2k}$ and then removed from G.

Rule 4.3: If there exists CC a connected component of G such that $|CC| \geq 4k + 1$ and u the vertex of minimal degree in CC verifies $d_G(u) \leq k$. Then, for any optimal solution S it holds that $d_{G\Delta S}(u) \leq 2k$. So, with respect to the solution S, Q, the γ-quasi-clique of $G\Delta S$ containing u, contains at most $\frac{1}{\gamma}2k + 1 < 4k + 1$ vertices. Again, for $C \subseteq CC$ such that $G[C]$ is connected, $|C| = 4k + 1$, C contains u and T a spanning tree of $G[C]$, we can try to remove every edge of T and update k: $k \leftarrow k - 1$.

Rule 4.4: If there exists CC a connected component of G such that u the vertex of minimal degree in CC verifies $d_G(u) \geq k + 1$. Then, CC is $(k + 1)$-edge-connected, thus it is pointless to remove edges inside CC. So CC is a γ-quasi-clique of $G\Delta S$ and is only missing edges in G. Thus, we process CC by solving the γ-Cluster Completion problem on $G[CC]$. We then update k and remove CC from G.

Let's now evaluate the running time of this process. Each Rule can be applied in time $2^{O(k \log k)} \cdot n^{O(1)}$ and/or create a number of branches lower than $O(k)$. Thus, the total time complexity can be expressed as : $2^{O(k \log k)} \cdot n^{O(1)}$. □

4.3 Fixed number of clusters

In the section we provide FPT algorithms for both the (γ, p)-Cluster Deletion and the (γ, p)-Cluster Editing problems. These algorithms are actually really similar to the ones previously presented and use most of the same rules.

Theorem 39. *For $p \geq 2$ the (γ, p)-Cluster Deletion problem can be solved in time $2^{O(k \log k)} \cdot n^{O(1)}$.*

Proof. (Theorem 39) Let $G = (V, E)$ be a graph and k be an integer. We apply Rules 3.1 and 3.3 from the proof of Theorem 37 as much as possible. After applying these rules let's note $CC_1, ..., CC_r$ the connected components of G. We know that $\text{diam}(G[CC_i]) \leq 2$ and the CC_i either is a γ-quasi-clique or is not and is $(k+1)$-edge connected. Several cases are possible.

- If there exist CC_i such that CC_i is not a γ-quasi-clique and is $(k+1)$-edge-connected then (G, k) is a no-instance.

Hence, let's assume that all connected components are γ-quasi-cliques. Note that if $|CC_i| \geq \frac{1}{\gamma}(k+1)+1$ then $\delta(G[CC_i]) \geq \gamma(|CC_i|-1) \geq k+1$ and thus, CC_i is $(k+1)$-edge-connected. Let's also assume without loss of generality that the first r' ones are not $(k+1)$-edge connected and are of size $\leq \frac{1}{\gamma}(k+1)+1$.

- If $r > p$ then (G, k) is a no-instance. Indeed, when deleting edges the number of connected components can only increase.
- If the previous cases do not hold then $r' \leq p$ and $|CC_1 \cup \cdots \cup CC_{r'}| \leq \frac{1}{\gamma}(k+1)p \leq (2k+2)p$. So, each possible solution S of size at most k on $G[CC_1 \cup \cdots \cup CC_{r'}]$ can be tried in time $((2k+2)p)^{2k} \cdot n^{O(1)}$ using the brute-force algorithm. If one of these solutions verifies that the solution graph $G \setminus S$ has exactly p connected components which are γ-quasi-clique, then (G, k) is a yes-instance otherwise it is a no-instance.

The total complexity of this process is $2^{O(k \log k)} \cdot n^{O(1)}$.

□

Theorem 40. *For $p \geq 2$ the (γ, p)-Cluster Editing problem can be solved in time $2^{O(k \log k)} \cdot n^{O(1)}$.*

Proof. (Theorem 40) Let $G = (V, E)$ be a graph and k be an integer. We apply Rule 4.1.b, Rule 4.3.c and Rule 3 from the proof of Theorem 38 as much as possible. After applying these rules let's note $CC_1, ..., CC_r$ the connected components of G. Note that if $r > k + p$, then there are no solution. Indeed, adding an edge can only decrease the number of connected components by one. Let $A = \max(\frac{1}{2\varepsilon}, 4)$. We know that each connected component CC_i verifies:

- $|CC_i| \leq \frac{1}{2\varepsilon}k + 1$ or $\text{diam}(G[CC_i]) \leq 2$,
- $|CC_i| \leq 4k$ or $\delta(G[CC_i]) \geq k+1$.

Hence, for CC_i either $|CC_i| \leq Ak + 1$ or CC_i is $(k+1)$-edge connected. Without loss of generality we assume that the first r' connected components are not $(k+1)$-edge-connected and the following ones are. Note that it is pointless to remove edges inside a $(k+1)$-edge-connected set of vertices. Also, $|CC_1 \cup \cdots \cup CC_{r'}| \leq (Ak+1)(k+p)$. Hence, we can try every possible edge deletion set S_1 of size at most k on $G[CC_1 \cup \cdots \cup CC_{r'}]$ using a brute-force algorithm in time $2^{O(k \log k)}$. Then, we solve the (γ, p)-Cluster Completion problem on $G \setminus S_1$ in polynomial time.

The total complexity of this process is $2^{O(k \log k)} \cdot n^{O(1)}$. □

5 Conclusion

In this paper, we introduce and study a relaxation of the clustering problems which we call the γ-clustering problems. In these problems the relaxation is made on the clusters where the clique constraint is replaced with γ-quasi-cliques. For $\gamma > \frac{1}{2}$, we prove that the γ-Cluster Deletion and γ-Cluster Editing problems are NP-complete while the γ-Cluster Completion problem is solvable in polynomial time. We also provide FPT algorithms parameterized by k, the size of the solution, for the two NP-complete problems. Finally, we also study variants of the γ-clustering problems, called the (γ, p)-clustering problems, where the number of final clusters must be a fixed constant p. We obtain similar complexity and parameterized complexity with these problems.

Future works on the γ-clustering problems could look at other approaches such as approximations or betterment of our results such as kernalizations. Finally, the study of similar problems with other relaxations of cliques such as density-based quasi-cliques or s-plexes could be interesting, revealing new similarities and differences between these concepts.

Acknowledgment: This work is partially funded by Agence Nationale de la Recherche under grant ANR-20-CE23-0002.

References

[1] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. pages 238 – 247, 2002.

[2] Ambroise Baril, Riccardo Dondi, and Mohammad Mehdi Hosseinzadeh. Hardness and tractability of the γ-complete subgraph problem. *Information Processing Letters*, page 106105, 2021.

[3] Amir Ben-Dor, Ron Shamir, and Zohar Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, pages 281 – 297, 1999.

[4] Mauro Brunato, Holger H. Hoos, and Roberto Battiti. On effectively finding maximal quasi-cliques in graphs. In *Learning and Intelligent Optimization*, pages 41 – 55, 2008.

[5] Yixin Cao and Jianer Chen. Cluster editing: Kernelization based on edge cuts. *Algorithmica*, pages 152 – 169, 2012.

[6] Moses Charikar, Venkatesan Guruswami, and Anthony Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences*, pages 360 – 383, 2005.

[7] Jeffrey Dean and Monika R Henzinger. Finding related pages in the world wide web. *Computer Networks*, pages 1467 – 1479, 1999.

[8] Michael R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified np-complete graph problems. *Theoretical Computer Science*, pages 237 – 267, 1976.

[9] Michelle Girvan and Mark E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, pages 7821 – 7826, 2002.

[10] Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier. Graph-modeled data clustering: Fixed-parameter algorithms for clique generation. In *Algorithms and Complexity*, pages 108 – 119, 2003.

[11] Pinar Heggernes, Daniel Lokshtanov, Jesper Nederlof, Christophe Paul, and Jan Arne Telle. Generalized graph clustering: Recognizing (p, q)-cluster graphs. In *Graph Theoretic Concepts in Computer Science*, pages 171 – 183, 2010.

[12] Christian Komusiewicz. Multivariate algorithmics for finding cohesive subnetworks. *Algorithms*, page 21, 2016.

[13] Daniel Lokshtanov and Dániel Marx. Clustering with local restrictions. *Information and Computation*, pages 278 – 292, 2017.

[14] Grigory Pastukhov, Alexander Veremyev, Vladimir Boginski, and Oleg Prokopyev. On maximum degree-based γ-quasi-clique problem: Complexity and exact approaches. *Networks*, pages 244 – 257, 2017.

[15] Jeffrey Pattillo, Alexander Veremyev, Sergiy Butenko, and Vladimir Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, pages 244 – 257, 2013.

[16] Seyed-Vahid Sanei-Mehri, Apurba Das, and Srikanta Tirthapura. Enumerating top-k quasi-cliques. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 1107 – 1112, 2018.

[17] Ron Shamir, Roded Sharan, and Dekel Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, pages 173 – 182, 2004.

[18] Yossi Shiloach. Another look at the degree constrained subgraph problem. *Information Processing Letters*, pages 89 – 92, 1981.

[19] Victor Spirin and Leonid A. Mirny. Protein complexes and functional modules in molecular networks. In *Proceedings of the national Academy of sciences*, pages 12123 – 12128, 2003.

[20] Takeaki Uno. An efficient algorithm for enumerating pseudo cliques. In *Proceedings of the 18th International Conference on Algorithms and Computation*, pages 402 – 414, 2007.