Clustering with partial information*

Hans L. Bodlaender† Michael R. Fellows‡ Pinar Heggernes§ Federico Mancini§
Frances Rosamond‡ Charis Papadopoulos§

Draft, March 20, 2008

Abstract

The Correlation Clustering problem, also known as the Cluster Editing problem, seeks to edit a given graph by adding and deleting edges to obtain a collection of disconnected cliques, such that the editing cost is minimized. The Edge Clique Partition problem seeks to partition the edges of a given graph into edge disjoint cliques, such that the number of cliques is minimized. Both problems are naturally NP-hard, and they are well studied with respect to approximation and fixed parameter tractability. In this paper we study these two problems in a more general setting, where the input graphs miss some information, meaning that whether or not there is an edge between some pairs of vertices is not decided in advance. On such graphs the problems are previously studied only for approximation. For both problems, we give parameterized algorithms through kernelization. For the first problem, we show fixed parameter tractability when the parameters are the editing cost and the minimum number of vertices to cover the undecided edges. For the second problem, we show fixed parameter tractability when the parameters are the number of cliques and the minimum number of vertices to cover the undecided edges. For the second problem we also show that parameterizing with only the number of cliques is not enough, by proving that the problem remains NP-hard when the number of cliques that the graph is partitioned into is a fixed constant greater than 2.

1 Introduction

The Correlation Clustering problem was introduced and proved NP-hard by Bansal at al. in FOCS 2002 [1, 2]. Given a complete graph with labels $+$ or $-$ on each edge, the problem is to partition the vertices into clusters so that the number of $-$ edges inside each cluster plus the number of $+$ edges between the clusters, is minimized. Taking $+$ edges as edges and $-$ edges as non-edges, this problem is equivalent to the Cluster Editing problem, where we are given a non-complete graph and asked to add and delete the total minimum number of edges so that the resulting graph is a collection of disconnected cliques. The problem has been proven NP-hard several times, as it has been discovered and rediscovered in various applications areas, like hierarchical tree clustering [18], computational biology [3, 23], and phylogenetic trees [7].

*This work is supported by the Research Council of Norway.

†Department of Information and Computing Sciences, Utrecht University, The Netherlands. Email: hansb@cs.uu.nl

‡School of Electrical Engineering and Computer Science, University of Newcastle, Australia. Emails: mfellows@cs.newcastle.edu.au, frosamond@cs.newcastle.edu.au

§Department of Informatics, University of Bergen, 5020 Bergen, Norway. Emails: pinar@ii.uib.no, federico@ii.uib.no, charis@ii.uib.no

More general versions of Correlation Clustering have been defined and studied from the point of view of approximation [6, 8, 9, 11]. We study the version when the input graph is an arbitrary graph (not necessarily complete) with positive weights on the edges. In this setting the non-edges represent lack of information, and the weights represents how much two vertices “agree” or “disagree”, according to which label the edge between them has. The goal remains to find a partition into clusters that minimizes the disagreement. We introduce what we call *fuzzy* graphs as an alternative equivalent representation of the input graphs just described. In addition to edges and non-edges with positive editing costs, a fuzzy graph has also a set of *fuzzy edges* that can be turned into edges or non-edges without any editing cost. Hence Correlation Clustering on general weighted labeled graphs is equivalent to Weighted Cluster Editing on fuzzy graphs. In this paper we show that Weighted Fuzzy Cluster Editing is fixed parameter tractable when parameterized by the editing cost c and the minimum number r of vertices needed to cover the fuzzy edges. We give a kernel of size $O(c^2 + r)$ for this problem.

The fixed parameter tractability of the Cluster Editing problem has been shown with constantly improving results [5, 15, 22], and it has a linear kernel [13, 16]. The fixed parameter tractability of the Weighted Fuzzy Cluster Editing problem has been open until our results. An important open question is whether the problem is fixed parameter tractable when parameterized only by editing cost. Parameterized algorithms [10, 20] have gained much interest recently. A problem is fixed parameter tractable (FPT) if its input can be partitioned into a main part of size n and a parameter (usually an integer) k so that there is an algorithm that solves the problem in time $n^{O(1)}f(k)$, where f is a computable function. An important research direction within parameterized algorithms is the method of kernelization. A kernel is an instance of the problem smaller than the input, such that the problem has a solution on the input if and only if it has a solution on the kernel. It is well known that a problem is FPT if and only if a kernel of size $g(k)$ can be computed from the input in polynomial time, for a computable function g [10, 20]. Kernelization has also separately attracted a lot of attention recently, as it has been recognized as one of the most important aspects of parameterized complexity [12, 17, 20].

The second problem that we study in this paper is the Edge Clique Partition problem, which asks to partition the edges of a given graph into the smallest number of edge disjoint cliques. This problem is NP-hard [21], and it is recently shown to be FPT when parameterized by the number of cliques in the partition [19]. Again, we study this problem on fuzzy graphs. Given a fuzzy graph and an integer k, our problem asks whether the fuzzy edges can be turned into edges and non-edges so that the resulting set of edges can be partitioned into at most k edge disjoint cliques. In this problem no edge weights are involved, so we consider only unweighted fuzzy graphs. Notice also that no editing of the initial edges and non-edges of G is involved. For this problem we are able to give even a more complete set of results. First we prove that the problem remains NP-complete when k is a fixed constant and not a part of the input. Hence, parameterizing by only k is not enough to ensure fixed parameter tractability. Then, we show that when we use the minimum number r of vertices that cover the fuzzy edges as an additional parameter, the problem becomes FPT. We give a kernel of size $O(k^4 \cdot 3^r)$. The fixed parameter tractability of this problem on fuzzy graphs has been open until our results.

Note that the two problems studied in this paper are similar in the sense that in both we try to partition the graph into cliques, either vertex disjoint (and disconnected) or edge disjoint. The difference is in the objective function. In one case we minimize the “disagreement” between the cliques, and in the other the total number of cliques.

2 Notation and definitions

For an undirected graph $G = (V, E)$, we denote its vertex set by $V(G) = V$ and edge set by $E(G) = E$ with $n = |V|$. The set of *neighbors* of $v \in V$ is $N_G(v) = \{u \mid uv \in E\}$, and the *degree* of v is $d_G(v) = |N_G(v)|$. In addition, $N_G[v] = N_G(v) \cup \{v\}$. Analogously, for a set $S \subseteq V$, $N_G[S] = \cup_{x \in S} N_G[x]$ and $N_G(S) = N_G[S] \setminus S$. We omit subscripts when there is no ambiguity. An *induced subgraph* of G by $U \subseteq V$ is the graph $G[U] = (U, E_U)$, where $E_U = \{xv \in E \mid x, v \in U\}$. Given a vertex x of G, we denote the graph $G[V \setminus \{x\}]$ by $G - x$. In addition, for a set of edges $M \subset E$, we define $G(M) = (\{x \mid \exists u, xu \in M\},\ M)$.

A graph is *complete* if every pair of vertices are adjacent. If a subgraph is complete then it is called a *clique*. If $G[K]$ is a clique for $K \subset V$, we also say that K is a clique. If $G(M)$ is a clique for $M \subset E$, we also say that M is a clique. A vertex subset $S \subseteq V$ is a *vertex cover* if every edge of G has at least one endpoint in S. A *connected component* is a maximal connected subgraph.

We define a *fuzzy graph* $G = (V, E, F)$ to be a graph with two types of edges: E is the set of *real edges*, and F is the set of *fuzzy* edges. Between all other pair of vertices in the graph we say that we have *non-edges*. When we decide for each fuzzy edge whether it should become a real edge or a non-edge, we say that we *realize* the fuzzy edges. The resulting graph is called a *normalization* of the fuzzy graph. Formally we say that (R^+, R^-) with $F = R^+ \cup R^-$ is a *realization* of F into real edges R^+ and non-edges R^- such that $G' = (V,\ E \cup R^+)$ is the corresponding normalization of $G = (V, E, F)$. When we speak about the *connected components of a fuzzy graph*, we mean the connected components of the graph obtained by turning all fuzzy edges into non-edges. So a *connected fuzzy graph* is a fuzzy graph where between any two vertices there is a path of real edges.

3 Parameterized cluster editing with partial information

A *cluster graph* is a graph where each connected component is a clique. In this section we study the problem of editing a weighted fuzzy graph $G = (V, E, F)$ to obtain a cluster graph. *Editing* means turning some real edges into non-edges (*deleting*), turning some non-edges into real edges (*adding*), and turning all fuzzy edges into either real edges or non-edges. Each edge and non-edge is associated with a positive weight, whereas each fuzzy edge has weight 0. The *cost* of an editing is the sum of the weights of the deleted and added edges, and the goal is to minimize the cost. The problem is formally defined as follows.

Weighted Fuzzy Cluster Editing (WFCE)
Input: A fuzzy graph $G = (V, E, F)$, a weight function $w : V \times V \to \mathbb{N}$ such that $w(uv) = 0$ if $uv \in F$ and $w(uv) > 0$ if $uv \notin F$, and a natural number $c \geq 0$.
Question: Is there a set $M \subseteq V \times V$ such that: $G' = (V, (E \setminus M) \cup (M \setminus E))$ is a cluster graph and $\sum_{uv \in M} w(uv) \leq c$?

First we characterize the fuzzy graphs that can be turned into a cluster graph just by realizing the fuzzy edges, that is, without any editing cost. We show that they can be defined by a family of forbidden induced (fuzzy) subgraphs. The result was already noted in [11], but we restate it in a form more suitable for our framework and we give a constructive proof.

We define a *fuzzy path* $P_k^f = \{v_1, v_2, ..., v_k\}$, a path where for every $1 \leq i \leq k-1$ we have that $v_i v_{i+1}$ is a real edge, while $v_1 v_k$ is a non-edge, and all the other pairs of vertices are joined by fuzzy edges (see Figure 1).

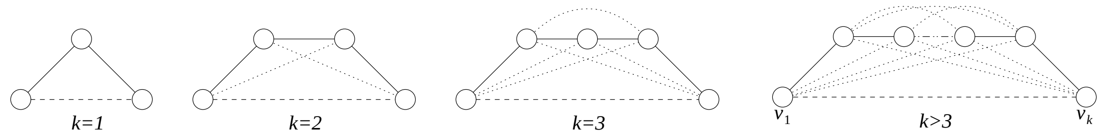

Figure 1: Example of fuzzy paths for $k = 1, 2, 3$ and the intuition for the general case. Real edges are represented by continuous lines, non-edges by dashed lines, and fuzzy edges by dotted lines.

Theorem 3.1 *Let G be a fuzzy graph. Then there exists a realization of the fuzzy edges that results in cluster graph without editing any real edge or non-edge if and only if G does not contain any induced subgraph isomorphic to P_k^f for $k \geq 3$.*

Proof. Let G contain a $P_k^f = \{v_1, v_2, ..., v_k\}$ as an induced subgraph, and assume for a contradiction that there is a normalization G' of G that is a cluster graph. We know that the vertices $v_1, v_2, ..., v_k$ must appear in the same clique of G' since they induce a connected graph no matter how we realize the fuzzy edges. However, in this case, the clique of G' would contain also the non-edge v_1v_k, that has not been edited, giving a contradiction.

If there are no fuzzy paths in G, then we show that every connected component of G is without non-edges. Once we prove this, it is easy to see that it is enough to turn all fuzzy edges in a connected component into real edges, and all fuzzy edges between connected components into non-edges to get a cluster graph G'. Assume there is a connected component containing a non-edge uv, then we can find a shortest path path $\{u, p_1, p_2, ..., v\}$ of real edges connecting u and v. However, if this is the case, we show that either the whole path $\{u, p_1, p_2, ..., v\}$ or one of its subpaths induces a fuzzy path. If $G[\{u, p_1, p_2, ..., v\}]$ is a fuzzy path we are done. Otherwise, since we took the shortest uv-path, we can deduce that in this subgraph there is some non-edge other than uv. Pick the non-edge p_ip_j such that p_i and p_j are closest on the path, and call P_{ij} the corresponding subpath that they define. Notice that one of p_i and p_j can be also u or v. As we took the closest such pair, there cannot be other non-edges in $G[P_{ij}]$, and no real edges other than the ones on the p_i, p_j-path, as this is a subpath of a shortest uv-path. It follows that all remaining edges are fuzzy, hence we have a fuzzy path, concluding the proof. ■

The c-Weighted Fuzzy Cluster Editing problem (c-WFCE) is the WFCE problem where we choose c of the problem definition to be the fixed parameter. The complexity of c-WFCE is open even for the unweighted case. The characterization given in Theorem 3.1 is through an infinite set of forbidden induced subgraphs, and hence an FPT algorithm for c-WFCE does not follow from the results of Cai [5].

In order to give an FPT algorithm, we introduce an additional parameter. We define a *fuzzy vertex cover* of a fuzzy graph to be a vertex subset S such that each fuzzy edge has an endpoint in S. The new parameter is $r = |S|$ where S is a smallest fuzzy vertex cover of G. We call the corresponding new problem the (c, r)-Weighted Fuzzy Cluster Editing, or (c, r)-WFCE, problem. Observe that checking whether G has a fuzzy vertex cover of size at most r is FPT when parameterized by r. To do this we create a non-fuzzy graph G' from $G(F)$ by turning all real edges of $G(F)$ into non-edges and all fuzzy edges into real edges. It is easy to see that G has a fuzzy vertex cover with at most r vertices if and only if G' has a vertex cover of at most r vertices. Since the r-vertex cover problem is well known to be FPT, our claim follows.

3.1 Kernel for the (c, r)-Weighted Fuzzy Cluster Editing problem

We show the fixed parameter tractability by giving a set of rules that either allow us answer NO, or produce a kernel of size $O(c^2 + r)$ in polynomial time, for the (c, r)-WFCE problem. First we give a general result to simplify some later proofs.

Observation 3.2 *Let G be a weighted fuzzy graph with connected components $C_1, \ldots, C_l$. Then G can be made into a cluster graph with editing cost at most c if and only if each connected component C_i can be made into a cluster graph with editing cost at most c_i, such that $\sum_{1 \le i \le l} c_i \le c$.*

Proof. If each connected component C_i can be made into a cluster graph with editing cost at most c_i such that $\sum_{1 \le i \le l} c_i \le c$, then G can be made into a cluster graph G' with editing cost at most c, since there are only non-edges and fuzzy edges between any two connected components of G, and the fuzzy edges can be turned into non-edges with no cost.

Assume now that G can be made into a cluster graph G' with editing cost at most c. If G' contains a cluster K that contains vertices from different connected components of G, then all edges in K between vertices of different connected components of G are either fuzzy edges or non-edges of G. Hence, if for every such edge we either keep the original non-edge or turn the fuzzy edge into a non-edge, we get a vertex disjoint union of smaller cliques rather than the cluster K. If we apply this transformation to each cluster of G' containing vertices from different connected components of G, we get a new graph G'' with the following properties: G'' is a cluster graph, the editing cost of turning G into G'' is at most c, and every cluster of G'' contains only vertices from the same connected component of G. Consequently, there must exist a value c_i such that $G[C_i]$ can be made into a cluster graph, namely $G''[C_i]$, with at most c_i as the editing cost, where $\sum_{1 \le i \le l} c_i \le c$, proving the theorem. ∎

Now we start presenting the rules, which will together give us the desired kernel and the resulting FPT algorithm for (c, r)-WFCE.

Rule 3.1.1 *If there is a connected component with no non-edges, remove it.*

Lemma 3.3 *Rule 3.1.1 is correct and can be applied in linear time.*

Proof. Since such a connected component can be made into a clique by only realizing fuzzy edges any edge, removing it will not affect the final result by Observation 3.2. Finding the connected components of a graph and checking its edges can be clearly done in linear time. ∎

Rule 3.1.2 *If Rule 3.1.1 does not apply and there are more than $c+1$ connected components, then answer* NO.

Lemma 3.4 *Rule 3.1.2 is correct and can be applied in linear time.*

Proof. If Rule 3.1.1 does not apply, then in every connected component there is at least one non-edge, and at least one path of real edges connecting the endpoints. As already shown in the proof of Theorem 3.1, this means that there exists a fuzzy path in each connected component, and each of them requires at least one editing to be destroyed. As there are more than c disjoint fuzzy paths, the result follows. Checking the number of connected components of a graph can clearly be obtained by a linear-time graph traversal. ∎

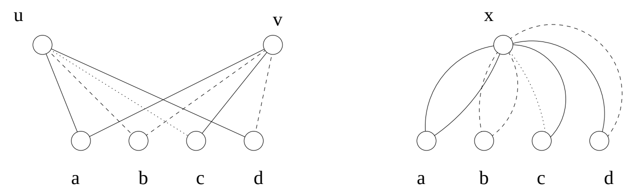

Figure 2: An example of contraction of two vertices u and v, with the resulting parallel edges. Real edges are represented by continuous lines, non-edges by dashed lines, and fuzzy edges by dotted lines.

For the following rule, note that a *minimum cut* between two vertices u and v is the minimum total weight of a collection of real edges that must be deleted so that u and v have no real paths between them. For this rule we also need some new definitions. When we *contract* two vertices u and v into one new vertex x, then u and v are deleted from the graph, x is added to the graph, and each previous pair of real edge, fuzzy edge, or non-edge uz and vz, appears now as two *parallel* edges between x and z. See Figure 2 for an example.

Rule 3.1.3 *If there are vertices u and v, with minimum cut value at least $c+1$, then contract u and v into one vertex x and do the following:*

1. *If uv was a non-edge decrease c by $w(uv)$.*
2. *If there are parallel edges with endpoint x and at least one of them is fuzzy, remove the fuzzy edge.*
3. *If there are parallel real edges (resp. non-edges) with endpoint x, replace them with one real edge (resp. non-edge) with weight equal to the sum of the weights of the parallel real edges (resp. non-edges).*
4. *If there is a real edge $e = ax$ in parallel with a non-edge $f = ax$ and:*
   (a) *If $w(e) > w(f)$, then subtract $w(f)$ from c and replace e and f with a real edge e' such that $w(e') = w(e) - w(f)$.*
   (b) *If $w(e) < w(f)$, then subtract $w(e)$ from c and replace e and f with a non-edge f' such that $w(f') = w(f) - w(e)$.*
   (c) *If $w(e) = w(f)$, then subtract $w(e)$ from c and replace e and f with a fuzzy edge g.*

If now $c < 0$, answer NO.

Lemma 3.5 *Rule 3.1.3 is correct and can be applied in polynomial time.*

Proof. Let $G = (V, E, F)$ be the current graph and $G' = ((V \setminus \{u, v\}) \cup \{x\}, E', F') = (V', E', F')$ and c' the new graph and parameter we obtain after applying the rule. Then we show that we can obtain a cluster graph $H = (V, E_H)$ from G with editing cost at most c if and only if we can obtain a cluster graph $H' = (V', E_{H'})$ from G' with editing cost at most c'.

Assume we can make a cluster graph $H = (V, E_H)$ from G with at most $c_T \leq c$ edge modifications. Since we cannot separate u from v by deleting edges of total weight less than $c+1$, these vertices must belong to the same cluster of H. Let us call such a cluster K. This means that for every other vertex $z \in V$, either $zv, zu \in E_H$ if z is in the same cluster as u and v in H, or $zv, zu \notin E_H$ otherwise. When we contract u and v into one new vertex x, we modify only the edges between x and the other vertices of the graph. That is, $G[V \setminus \{u,v\}] = G'[V' \setminus \{x\}]$. We show that if we edit the edges of G' so that a vertex in H' is adjacent to x if and only if it is adjacent to both u and v in H, and we set $H[V \setminus \{u,v\}] = H'[V' \setminus \{x\}]$, then we get a valid solution H' for G'.

The graph H' is clearly a cluster graph. In fact $H'[V' \setminus \{x\}]$ is a cluster graph by construction, and $N_{H'}[x] = N_H[v] \setminus \{u\} = N_H[u] \setminus \{v\}$, i.e., $H' = H - v = H - u$ if we do not consider weights. Let us check the editing cost of turning G' into H'. The cost of turning $G'[V' \setminus \{x\}]$ into $H'[V' \setminus \{x\}]$ is the same as that of turning $G[V \setminus \{u,v\}]$ into $H[V \setminus \{u,v\}]$. Let us call this cost c_1. Then, for every vertex $z \notin K$ that has real edges to both u and v, or one real edge and a fuzzy edge, the cost of replacing these edges with non-edges is $w(zu) + w(zv)$ both in G and G', by the construction given in case 2 and 3. The same is true for every vertex $z \in K$ with non-edges to both u and v, or one non-edge and one fuzzy edge. If both edges are fuzzy, there is no cost in either G or G'. Let us call the total cost of these edges c_2. Now, for every vertex z such that $zu \in E$, but $zu \notin E \cup F$ or vice versa, we have to edit either zu or zv in G to get any valid solution, including H. Hence, for each such vertex, we always have a cost of at least $\min\{w(zu), w(zv)\}$, no matter whether z will belong to K or not in the solution. We call the set of these vertices Z, and we partition it in 4 smaller sets according to which kind of editing operation we apply to each vertex $z \in Z$ in order to get H. So we have: $Z_u^+ = \{z \mid z \in K \wedge zv \in E \wedge zu \notin E \cup F\}$, $Z_v^+ = \{z \mid z \in K \wedge zu \in E \wedge zv \notin E \cup F\}$, $Z_u^- = \{z \mid z \notin K \wedge zu \in E \wedge zv \notin E \cup F\}$, and $Z_v^- = \{z \mid z \notin K \wedge zv \in E \wedge zu \notin E \cup F\}$. The "+" and "−" indicate whether we add or remove, respectively, an edge incident to z, and u and v indicate whether we edit zu or zv. By case 4a,4b and 4c of the rule, to disconnect or connect z and x in G', the cost is $w(zu) - \min\{w(zu), w(zv)\}$ or $w(zv) - \min\{w(zu), w(zv)\}$, according to which vertex is incident to the edited edge in G. We define the total cost of these editings in G' as $c_3 = \sum_{z \in Z_u^+ \cup Z_u^-} w(zu) - \min\{w(zu), w(zv)\} + \sum_{z \in Z_v^+ \cup Z_v^-} w(zv) - \min\{w(zu), w(zv)\}$. Notice at this point, that $c' = c - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$, where the $'' - w(uv)''$ is in parenthesis because it might be there or not, according to case 1 of the rule. In other words $c - c'$ is a lower bound on the editing cost of G. Hence if $c' < 0$, then the editing cost is at least $c - c' > c$ and there cannot be a solution for G. Therefore we can safely assume $c' \geq 0$ and $c_T \geq \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$, i.e. $c_T - \sum_{z \in Z} \min\{zu, zv\}(-w(uv)) \geq 0$. Now we are ready to show that $c'_T = c_1 + c_2 + c_3 \leq c'$. By construction $c' = c - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$ and being $c \geq c_T$, we have that $c' \geq c_T - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$. By our previous discussion $c_T = c_1 + c_2 + \sum_{z \in Z_u^+ \cup Z_u^-} w(zu) + \sum_{z \in Z_v^+ \cup Z_v^-} w(zv)(+w(uv))$. Putting the two things together we get $c' \geq c_1 + c_2 + \sum_{z \in Z_u^+ \cup Z_u^-} w(zu) + \sum_{z \in Z_v^+ \cup Z_v^-} w(zv)(+w(uv)) - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$. This translates into $c' \geq c_1 + c_2 + c_3 = c'_T$ as assumed, proving one direction of our main claim.

For the other direction, we use almost the same argument. Assume we can make a cluster graph $H' = (V, E_{H'})$ from G' using at most $c'_T \leq c'$ edge modifications. We show that if we edit the edges of G so that a vertex in H is adjacent to both u and v if and only if it is adjacent to x in H', we set $H[V \setminus \{u,v\}] = H'[V' \setminus \{x\}]$, and we add a real edge between u and v if there is not one, then we get a valid solution H for G. First notice that the graph H we obtain in this way is a cluster graph. By construction $H[V \setminus \{u,v\}]$ is a cluster graph, and $N_{H'}[x] = N_H[v] \setminus \{u\} = N_H[u] \setminus \{v\}$. If we do not consider the weights, the graph H is isomorphic to a graph obtained from H' adding a vertex incident to x and with the same adjacencies as x. This would clearly be a cluster graph. Now it is left to show that the editing cost to make G into H is $c_T \leq c$.

The argument is very similar to the one for the previous direction, so we use the same notation as well. The costs c'_1, c'_2 are defined as c_1 and c_2 in the previous part of the proof, and they are the same for both G and G'. We redefine $c'_3 = \sum_{z \in Z} w(zx)$ in G', and analyze the cost of the corresponding modifications in G. If $z \in Z$ and it costs $w(zx)$ to edit zx in G', then the equivalent editing cost in G to edit either zu or zv is $w(zx) + \min\{w(zu), w(zv)\}$ even if

$w(zx) = 0$, by construction of G'. Since $c = c' + \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$ and $c' \geq c'_T$, we get $c \geq c'_T + \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$. By definition $c'_T = c'_1 + c'_2 + c'_3$, and replacing this in the inequality, we obtain $c \geq c'_1 + c'_2 + \sum_{z \in Z} w(zx) + \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$, that is exactly c_T. Hence $c_T \leq c$, proving he correctness of the rule.

The running time is polynomial because the minimum cut of two vertices of a graph can be found in polynomial time [14], and G' can clearly be constructed in polynomial time as well. ∎

Theorem 3.6 *If Rules 3.1.1, 3.1.2 and 3.1.3 do not apply, and we have not answered* NO *yet, then either the current graph has at most $c^2 + 3c + r$ vertices, or the answer is* NO.

Proof. If Rule 3.1.1 and Rule 3.1.2 do not apply, it means that the graph has at most c connected components, and each of them must be edited. If Rule 3.1.3 does not apply, then there cannot be cliques of size greater than $c + 1$. Let us now consider a connected fuzzy graph $G = (V, E, F)$ with no clique of size greater than $c + 1$ and that can be made into a cluster graph by editing edges of total weight at least 1 and at most c. We prove that G cannot have more than $c^2 + 3c + r$ vertices. Let $X = V \setminus R$, where R is a minimum fuzzy vertex cover of G. The graph $G[X]$ does not contain any fuzzy edge, and since we can edit at most c edges, there are at most $2c$ vertices of X incident to edges involved in the editing. Let us call X' the set of these $2c$ vertices. Notice that $X' \neq \emptyset$ by assumption. Then, if there is a solution to the problem that edits at most c edges, the graph $G[X \setminus X']$ must be a union of disjoint cliques. First we claim that there cannot be more than $c + 1$ cliques in $G[X \setminus X']$. Since no vertex of these cliques is incident to an edited edge, and there are no fuzzy edges in between them, each of them must belong to a different clique in the solution. However, to create $c+1$ connected components from a connected graph, we need to remove at least c edges. Hence the claim is proved. From the previous argument, it also follows that all vertices connected to a clique in $G[X \setminus X']$ must end up in the same cluster in a solution. Assume they do not, then we should delete an edge incident to a vertex in $G[X \setminus X']$, contradicting that these vertices are not incident to an edited edge. Therefore every vertex in X' can be connected to at most one clique $G[X \setminus X']$, and furthermore it must be adjacent to all vertices in its clique. This means that every clique in $G[X \setminus X']$ has either size at most c, or it has size at most $c+1$ and has neighbors only in R. If we define N as the number of cliques in $G[X \setminus X']$ that have neighbors only in R, we can give the following bound on the number of vertices in G: $(c + 1) \cdot N + c \cdot (c + 1 - N) + (2c - N) + |R| = c^2 + 3c + r$. The first two terms give a bound on the number of vertices in $G[X \setminus X']$ according to the previous discussion, while the term $(2c - N)$ represents a tighter bound on $|X'|$. In fact, for every clique with neighbors only in R, there must be at least a distinct vertex in R incident to an edited edge. This because we need to disconnect the clique from the rest of the graph, but we cannot touch edges incident to its vertices. Besides at least one endpoint of the edge we have to remove will belong to the same cluster as the clique.

Consider now a fuzzy graph G with l connected components $C_1, \ldots, C_l$, where $1 \leq l \leq c$. By Observation 3.2 we know that there is a solution for G that edits at most c edges if and only if there is a solution for each $G[C_i]$ that edits at most c_i edges, such that $\sum_{i=1}^{l} c_i \leq c$. This means that, by what we just proved for connected fuzzy graphs, if there is a solution for G then $|V(G[C_i])| \leq c_i^2 + 3c_i + r_i$ for $1 \leq i \leq l$, where r_i is the size of a minimum vertex cover of $G[C_i]$. Hence $V(G) = \sum_{i=1}^{l} c_i^2 + 3c_i + r_i$, that is $\sum_{i=1}^{l} c_i^2 + 3\sum_{i=1}^{l} c_i + \sum_{i=1}^{l} r_i \leq (\sum_{i=1}^{l} c_i)^2 + 3c + r = c^2 + 3c + r$, completing the proof. ∎

It is easy to construct example where we have $(c + 1) \cdot N + c \cdot (c + 1 - N) + (c + 1 - N) + r = c^2 + 2c + 1 + r$ vertices in a *yes* instance for any given c (See Figure 3 for an example). Hence

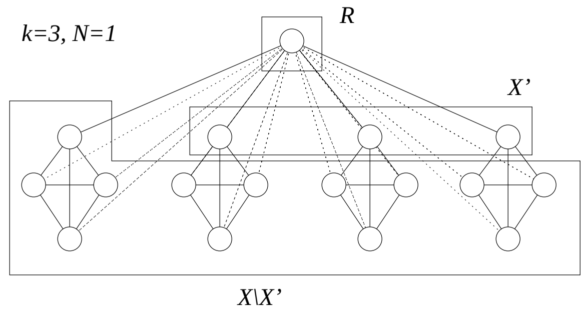

Figure 3: Example of a kernel with $c^2 + 2c + 1 + r$ vertices for $c = 3$. The non-edges are not drawn to keep the figure easier to observe.

Theorem 3.6 gives a quite tight bound on the size of a connected component, that is in any case $O(c^2 + r)$.

We have thus proved that the (c, r)-WFCE problem has a kernel of size $O(c^2 + r)$. We can now conclude the following.

Theorem 3.7 *The (c, r)-Weighted Fuzzy Cluster Editing problem can be solved in time $n^{O(1)} + O((c^2 + r)!)$.*

Proof. Run Rules 3.1.1, 3.1.2 and 3.1.3 on the input graph G until either we get a *no* as an answer or we get to an equivalent fuzzy graph that has size bounded by a function of c and r. Notice that the rules can be applied at most n times since, every time one of them is applied, either at least one vertex is removed from the graph, or we stop. Also, each of Rules 3.1.1, 3.1.2 and 3.1.3 can be applied in polynomial time by Lemmas 3.3, 3.4 and 3.5. This means that the total preprocessing can be performed in time polynomial in n. After the preprocessing we either return that there is no solution or the graph has with at most $c^2 + 3c + r$ vertices by Theorem 3.6 and we can just use brute force to solve the problem on this graph. The brute force approach consists in trying all possible vertex partition of the graph, and pick the one where the cost of making each set of the partition into a complete connected component is minimized. Since the number of possible partitions is factorial in the number of vertices, the result follows. ∎

4 Parameterized edge clique partitioning with partial information

In this section, we study the problem of partitioning the edges of a fuzzy graph $G = (V, E, F)$ into edge disjoint cliques. In this problem, no editing of the edges or non-edges of G is involved, but we have to decide for each fuzzy edge whether or not it should become a real edge or a non-edge. Below is a formal definition of the problem.

Fuzzy Edge Clique Partitioning (FECP)
Input: A fuzzy graph $G = (V, E, F)$ and an integer $k \geq 0$.
Question: Is there a realization (R^+, R^-) of the fuzzy edges such that the edges of $G' = (V, E \cup R^+)$ can be partitioned into at most k edge disjoint cliques?

Naturally, being a more general version of the problem on non-fuzzy graphs, the Fuzzy Edge Clique Partitioning problem is NP-hard as well. Interestingly, we show that it remains NP-hard also when k is a fixed constant and not a part of the input. Hence, in contrast to k-Edge Clique

Partitioning, k-Fuzzy Edge Clique Partitioning is NP-hard for every fixed $k \geq 3$. Then we show that, (k, r)-FECP, is FPT, where r is the minimum size of a fuzzy vertex cover of G.

4.1 k-Fuzzy Edge Clique Partitioning is NP-complete

Here we prove that is it NP-complete for every fixed $k \geq 3$ to decide whether the edges of a fuzzy graph can be partitioned into k edge disjoint cliques. The problem we reduce from is the classical k-Coloring problem. In this problem, the input is a graph $G = (V, E)$, and the problem is to decide whether the vertices of G can be colored with at most k colors, such that no two adjacent vertices have the same color. It is well known that this problem is NP-hard for every $k \geq 3$.

Theorem 4.1 *The k-Fuzzy Edge Clique Partitioning is NP-complete for fixed $k \geq 3$.*

Proof. Given a graph $G = (V, E)$ we build a fuzzy graph $G' = (V', E', F)$ as follows. For each vertex $v_i \in V$, create a new vertex u_i and call U the set of such vertices, so that $V' = V \cup U$. Now the only real edges in E' are the edges $v_i u_i$, because we replace each $v_i v_j \in E$ with a non-edge, and for every other pair of vertices we add a fuzzy edge. See Figure 4 for an example.

Now we claim that $G = (V, E)$ can be colored with at most k colors if and only if there exists a normalization of $G' = (V \cup U, E', F)$ whose edges can be partitioned into at most k edge disjoint cliques.

Assume there is a coloring of $G = (V, E)$ that uses only k colors, so that no two adjacent vertices have the same color. This is equivalent to partitioning V into k sets $A_1, \cdots, A_k$, such that $G[A_j]$ is an independent set for each $1 \leq j \leq k$. Let us partition also U into k sets $A'_1, \cdots, A'_k$ such that $u_i \in A'_j$ if and only if $v_i \in A_j$. Then, by construction and the fact that $G[A_j]$ contains only non-edges, we know that $G'[A_i \cup A'_i]$ contains only fuzzy edges and real edges for each $1 \leq j \leq k$. Hence, if we realize all fuzzy edges inside each $G'[A_i \cup A'_i]$ into real edges, and all the remaining fuzzy edges into non-edges, we get a non-fuzzy graph consisting of k vertex disjoint cliques, with no edges between them. Hence, we have a natural solution for the k-Clique Edges Partition problem on G', since vertex disjoint cliques are also edge disjoint and contain all edges.

Assume now that there exists a realization of the fuzzy edges of G', so that there is an edge partition S of the corresponding normalized graph H, such that $|S| \leq k$. By construction we know that the graph is not edgeless, so there cannot be a solution that uses zero cliques. First of all notice that if there is a solution, then there is also a solution where all cliques induced by the sets of S are vertex disjoint. Every vertex has exactly one real edge incident to it in G'. So, even if a vertex x is a gateway in the solution, and it is the in common to two or more cliques, there is only one clique $X \in S$ containing the real edge incident to x in G'. All the other edges incident to x in H that are not in X must, therefore, be either non-edges or fuzzy edges realized into either a non-edge or a real edge. Then we can just create a new solution realizing all the fuzzy edges incident to x that are not in X into non-edges instead. The new normalization we obtain, has clearly still a valid partition S', but x is not a gateway anymore. Iterating this process for all gateway vertices, we get a new normalized graph H' that is actually a disjoint union of at most k cliques. Let us now build a solution for the k-Coloring problem for G using S'. It is enough to give the same color to vertices of G that appear in the same clique of S', using a different color for each clique. This gives a legal k-coloring. First of all there are at most k cliques in S', so at most k colors are used. Besides we proved that each vertex and edge of G' belongs to only one clique of S', so we have no ambiguity and all vertices are colored with a unique color. Assume now that there is at least one edge $v_i v_j \in E$ such that the endpoints v_i and v_j got the same color. This implies that v_i and v_j

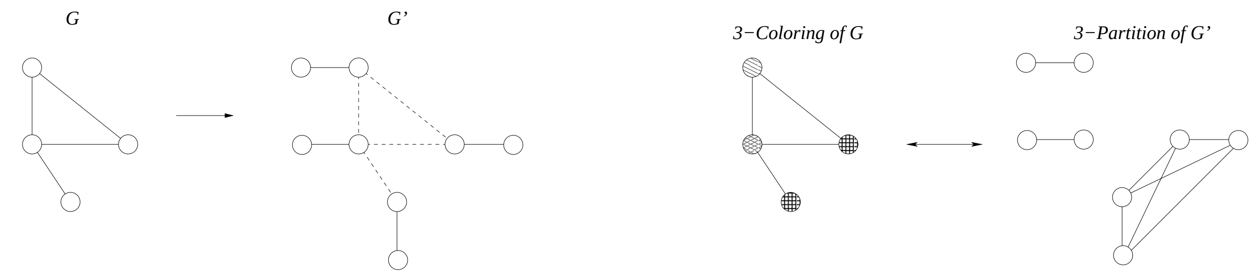

Figure 4: An example of the graph G' obtained from a graph G, and the equivalence of a solution of the 3-Coloring problem on G with a solution of the 3-Clique Partition problem on G'. In G' the fuzzy edges are not drawn to keep the figure clean, while in the corresponding clique partition the non edges are not drawn.

belongs to the same clique of S. However, by construction, there should be a non-edge between v_i and v_j in G'. Hence there cannot be any solution for G' where v_i and v_j belong to the same clique induced by a valid edge partition, giving a contradiction. This completes the proof of our claim.

Clearly, G' can be constructed from G in polynomial time; hence the theorem follows. ∎

4.2 The (k, r)-Fuzzy Edge Clique Partitioning is FPT

To obtain a kernel for (k, r)-FECP, we first give reduction rules that apply to k-Edge Clique Partitioning, hence the non-fuzzy version of our problem which simply asks whether the edges of a graph $G = (V, E)$ can be partitioned into at most k edge disjoint cliques.

First we give some observations on the structure of a partition $K = \{K_1, K_2, ..., K_k\}$ of E such that $G(K_i)$ is a clique for each i. For $K_i \in K$, we define $V(K_i)$ as the union of the endpoints of the edges in K_i, i.e. $V(G[K_i])$. We call the vertices that are in the intersection of some cliques in K *gateways*, while the vertices contained only in one clique are called *normal*. Two normal vertices in the same clique are said to be *co-normal*. We define a set $V' \subseteq V$ to be a *type* if there is at least one vertex v such that $N[v] = V'$. So we say that two vertices are of the same type if their closed neighborhood is identical, and that they are of different type otherwise. Finally notice that the intersection of two cliques in any solution cannot consist of more than one vertex, or there would be one edge covered by two cliques.

Theorem 4.2 ([4]) *Every edge clique partition of a complete graph on n vertices, except the trivial one of one clique, contains at least n cliques.*

Lemma 4.3 *If the answer to the k-Clique Partition problem for a graph $G = (V, E)$ is* YES*, then the answer is* YES *also for each induced subgraph of G.*

Proof. Assume there is a partition $K = \{K_1, \dots, K_l\}$ of E, such that $l \leq k$ and $G(K_i)$ is a clique for each $1 \leq i \leq l$. Then consider $V' \subset V$, and $G' = G[V \setminus V']$. We show that it is possible to modify K in order to obtain a valid solution $K' = \{K'_1, \cdots, K'_{l'}\}$ for G'. For each vertex $v \in V'$, remove all edges incident to v in G from the sets of K, and remove the sets the become empty during the process. Now for each set $K'_i \in K'$, we still have the property that $G'(K'_i)$ is a clique, because $G'(K'_i) = G'[V(K_i) \setminus V']$, and every induced subgraph of a clique is still a clique. The intersection of any two sets of K' is still empty, as it was for K, because we only deleted elements from them. Hence no edge is covered by more than one clique. Furthermore $\bigcup_{i=1}^{l'} K'_i = E(G')$, because $\bigcup_{i=1}^{l} K_i = E$ and we did not delete any edge between two vertices of $V \setminus V'$ by construction,

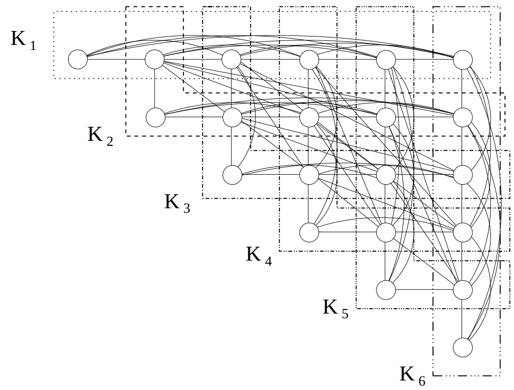

Figure 5: Example of a graph with $\binom{k}{2}$ gateway vertices. Every box represents a clique, and every two cliques intersect in exactly one distinct vertex.

but we removed all edges incident to the vertices of V'. Finally $|K'| \leq |K| \leq k$. This concludes the proof. ■

Lemma 4.3 implies that if there is even only one induced subgraph of G for which the answer is NO, then G itself is a NO instance. We will use this observation often.

Observation 4.4 *In any solution of k-Edge Clique Partition, there cannot be more than $\binom{k}{2}$ gateway vertices.*

Proof. Every two cliques can intersect in at most one vertex, or they would cover the same edge. Since there are at most $\binom{k}{2}$ possible intersection among k cliques, the result follows. ■

To show that this upper bound is tight, we provide a way to construct a graph G with $k(k-1)/2$ gateways, for any k. Let $K = \{K_1, K_2, \dots, K_k\}$ be a partition of the edges of a graph G into cliques, and let $|V(K_i)| = k$ for each $1 \leq i \leq k$. Now let v_i^j be the vertex i of $G(K_j)$. If we set $v_i^j = v_j^i$ for each $1 \leq i, j \leq k$, we get our graph, that we can see also in Figure 5. It is easy to see that every two cliques intersect in exactly one vertex, so that there is no edge covered by more than one clique. Furthermore, since for every pair of cliques, we have a different intersection, the bound we gave previously is tight.

For the next result, remember that the closed neighborhood of a vertex is the union of the cliques it belongs to in a feasible solution.

Observation 4.5 *If two vertices have the same type, then in any feasible solution either they are co-normal or they are both gateways.*

Proof. Take a partition K that is a valid solution for a graph $G = (V, E)$. We show that if two vertices are neither co-normal nor both gateways, they cannot have the same type.

Take two sets $K_i, K_j \in K$ and take two vertices that are normal but not co-normal, let us say $u \in V(K_i)$ and $v \in V(K_j)$. There cannot be an edge between them, or that edge should be covered by a third clique K_z. In this case we would have $u \in V(K_i) \cap V(K_z)$ and $v \in V(K_j) \cap V(K_z)$, meaning that they are both gateways, giving a contradiction. If there is no such edge, then their closed neighborhood cannot be identical, since u would not appear in $N[v]$ and vice versa. This settles the first case.

Assume now that u is normal with respect to some clique K_i and that v is a gateway. Since by definition a vertex is a gateway if it is in the intersection of some cliques, the vertex v must appear in at least two cliques K_j and K_z. Then $N[v] \subseteq V(K_j) \cup V(K_z)$. Since $N[u] = V(K_i)$, if $v \notin V(K_i)$, then $N[u] \cap N[v] = \emptyset$. However, even if we could choose K_j or K_z to be equal to K_i, let us say $K_j = K_i$, we would still get that $N[u] = V(K_i) \subset N[v] \subseteq V(K_u) \cup V(K_z)$, because every clique in K contains at least one edge, i.e, two vertices, and being u a normal vertex, it cannot have neighbors other than v in $V(K_z)$. Thus u and v cannot have the same type. ∎

Observation 4.6 *If there are more than $k + \binom{k}{2}$ vertices of different type, then the answer to k-Edge Clique Partition is* NO*.*

Proof. By Observation 4.5 we know that all vertices that can belong to the same clique in a solution must either have the same type or be gateways. Since by Observation 4.4 there can be at most $\binom{k}{2}$ gateway vertices, and, if the problem has a solution, at most k cliques, we can conclude that there cannot be more than $k + \binom{k}{2}$ different types of vertices. ∎

Now we show that a simple generalization of the rules and observations given until now, gives a kernelization of (k, r)-Fuzzy Edge Clique Partitioning, where r is the size of a minimum fuzzy vertex cover of the input graph $G = (V, E, F)$.

First we need to introduce a generalization of the type of a vertex for fuzzy graphs. The *fuzzy neighborhood* of a vertex v is the set of the vertices w such that $vw \in F$. We say that two vertices are of the same *absolute type* if their closed and fuzzy neighborhoods are equal.

Consider a fuzzy graph $G = (V, E, F)$, and let $S \subset V$ be a minimum fuzzy vertex cover of G, such that $|S| \leq r$. Then for each vertex in $X = V \setminus S$, there can be at most 3^r possible ways to have adjacencies in S. So we can classify the vertices of X into 3^r categories, so that the vertices in the same category have the same absolute type with respect to the vertices in S. Since $G[X]$ is a non-fuzzy graph, if there is no solution to k-Edge Clique Partitioning for $G[X]$, then there is no solution to (k, r)-FECP on G no matter how we realize the fuzzy edges, due to Lemma 4.3.

Rule 4.2.1 *If there are more than $(k + \binom{k}{2}) \cdot 3^r$ vertices with different absolute type in X, then the answer is* NO*.*

Lemma 4.7 *Rule 4.2.1 is correct and can be executed in polynomial time.*

Proof. If there are more then $(k + \binom{k}{2}) \cdot 3^r$ absolute types of vertices, then $G[X]$ must have more than $(k + \binom{k}{2})$ vertices of different types. Hence by Observation 4.6, there is no solution for $G[X]$. By Lemma 4.3, this implies that there is no solution for G as well, proving the first part of the statement.

The rule can be easily executed in polynomial time by listing the absolute closed neighborhoods of the vertices of G, and checking whether there are more than $(k + \binom{k}{2}) \cdot 3^r$ different ones. Since k and r are constants, the result follows. ∎

Rule 4.2.2 *If Rule 4.2.1 does not apply and there are more than $\binom{k}{2} + 1$ vertices of the same absolute type in X, then remove one.*

Lemma 4.8 *Rule 4.2.2 is correct and can be executed in polynomial time.*

Proof. Let us u be the vertex we remove. Then we show that there is a solution for G if and only if there is a solution for $G - u$. Assume there is a normalization H of $G - u$ that admits a valid edge partition K', i.e., with at most k elements and each element induces a clique in H. Since in

$G-u$ there are at least $\binom{k}{2}+1$ of the same absolute type as u, we know that at least one of them, let us say v, is a normal vertex for exactly one clique induced by some set of K'. By Rule 4.4, in fact, there cannot be more than $\binom{k}{2}$ gateways in any solution. This means that if we realize the fuzzy edges of u as the fuzzy edges of v in H, we can add u to the clique where v belongs in H, so that u and v are co-normal. This new graph is a normalization of G with a valid edge partition K. In fact K can be obtained from K' just adding all edges incident to u to the same set of K' containing all edges incident to v.

On the other hand, by Lemma 4.3, if there is a normalization G' of G with a valid edge partition, then there is also a normalization of $G-u$ that has a valid edge partition, namely $G'-u$. ∎

Lemma 4.9 *If Rules 4.2.1 and 4.2.2 do not apply, then the graph has at most* $(\binom{k}{2}+1)\cdot((k+\binom{k}{2})\cdot 3^r)+r$ *vertices.*

Proof. It follows directly by the fact that Rule 4.2.1 and Rule 4.2.2 do not apply. ∎

Theorem 4.10 (k,r)*-FECP is FPT with a kernel of size* $O(k^4\cdot 3^r)$.

Proof. Since Rules 4.2.1 and 4.2.2 can be applied in polynomial time, and at most a polynomial number of times, by Lemma 4.9 we showed that it is possible to produce a kernel of size $O(k^4\cdot 3^r)$ for the (k,r)-FECP problem. ∎

5 Concluding remarks

The main open problem about the parameterized complexity of the k-Fuzzy Cluster Editing problem still unsolved. In the meanwhile it would be interesting to try and improve the quadratic kernel for the (c,r)-Weighted Fuzzy Cluster Editing problem, to a linear one.

For (k,r)-Fuzzy Edge Clique Partitioning our kernel is exponential in r, so an open problem is whether a polynomial kernel exists.

References

[1] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. In *Proceedings of FOCS 2002 - 43rd Symposium on Foundations of Computer Science*, pages 238–, 2002.

[2] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

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

[4] N. J. De Bruijin and P. Erdos. On a combinatorial problem. *Ind. Math.*, 10:421–423, 1948.

[5] L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.

[6] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. *J. Comput. Syst. Sci.*, 71(3):360–383, 2005.

[7] Z. Chen, T. Jiang, and G. Lin. Computing phylogenetic roots with bounded degrees and errors. *SIAM J. Comput.*, 32(4):864–879, 2003.

[8] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2-3):172–187, 2006.

[9] E. D. Demaine and N. Immorlica. Correlation clustering with partial information. In *Proceedings of RANDOM-APPROX 2003 - 7th International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 1–13, 2003.

[10] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[11] D. Emanuel and A. Fiat. Correlation clustering - minimizing disagreements on arbitrary weighted graphs. In *Proceedings of ESA 2003 - 11th Annual European Symposium on Algorithms*, pages 208–220, 2003.

[12] M. R. Fellows. The lost continent of polynomial time: Preprocessing and kernelization. In *Proceedings of IWPEC 2006 - Parameterized and Exact Computation, Second International Workshop*, pages 276–277, 2006.

[13] M. R. Fellows, M. A. Langston, F. A. Rosamond, and P. Shaw. Efficient parameterized preprocessing for cluster editing. In *Proceedings of FCT 2007 - Fundamentals of Computation Theory, 16th International Symposium*, pages 312–321, 2007.

[14] A. V. Goldberg. Recent developments in maximum flow algorithms (invited lecture). In *Proceedings of SWAT '98 - 6th Scandinavian Workshop on Algorithm Theory*, volume 1432, pages 1–10, 1998.

[15] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Graph-modeled data clustering: Exact algorithms for clique generation. *Theory Comput. Syst.*, 38(4):373–392, 2005.

[16] J. Guo. A more effective linear kernelization for cluster editing. In *Proceedings of ESCAPE 2007 - Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, First International Symposium*, pages 36–47, 2007.

[17] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[18] M. Křivánek and J. Morávek. NP -hard problems in hierarchical-tree clustering. *Acta Inf.*, 23(3):311–323, 1986.

[19] E. Mujuni and F. A. Rosamond. Parameterized complexity of the clique partition problem. In *Proceedings of ACiD 2007 - 2nd Workshop of Algorithms and Complexity in Durham*, 2007.

[20] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[21] J. Orlin. Contentment in graph theory: Covering graphs with cliques. *Indagationes Math.*, 39:406–424, 1977.

[22] F. Protti, M. D. da Silva, and J. L. Szwarcfiter. Applying modular decomposition to parameterized bicluster editing. In *Proceedings of IWPEC 2006 - Parameterized and Exact Computation, Second International Workshop*, pages 1–12, 2006.

[23] R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics*, 144(1-2):173–182, 2004.