REPORTS
IN
INFORMATICS

Clustering with partial information

Hans L. Bodlaender, Michael R. Fellows,
Pinar Heggernes, Federico Mancini,
Charis Papadopoulos, Frances Rosamond

Department of Informatics

UNIVERSITY OF BERGEN
Bergen, Norway

Clustering with partial information[*]

Hans L. Bodlaender[†]   Michael R. Fellows[‡]   Pinar Heggernes[§]   Federico Mancini[§]

Charis Papadopoulos[§]   Frances Rosamond[‡]

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 vertex-disjoint cliques, such that the editing cost is minimized. The Edge Clique Partitioning 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 known to be NP-hard, and they have been previously studied with respect to approximation and fixed parameter tractability. In this paper we study these two problems in a more general setting that we term *fuzzy graphs*, where the input graphs may have missing information, meaning that whether or not there is an edge between some pairs of vertices of the input graph can be *undecided*.

For fuzzy graphs the Correlation Clustering and Edge Clique Partitioning problems have previously been studied only with respect to approximation. Here we give parameterized algorithms based on kernelization for both problems. We prove that the Correlation Clustering problem is fixed-parameter tractable on fuzzy graphs when parameterized by (k, r), where k is the editing cost and r is the minimum number of vertices required to cover the undecided edges. In particular we show that it has a polynomial-time reduction to a problem kernel on $O(k^2 + r)$ vertices. We provide an analogous result for the Edge Clique Partitioning problem on fuzzy graphs. Using (k, r) as parameters, where k bounds the size of the partition, and r is the minimum number of vertices required to cover the undecided edges, we describe a polynomial-time kernelization to a problem kernel on $O(k^4 \cdot 3^r)$ vertices. This implies fixed-parameter tractability for this parameterization. Furthermore we also show that parameterizing only by the number of cliques k, is not enough to obtain fixed-parameter tractability. The problem remains, in fact, NP-hard for each fixed $k > 2$.

1   Introduction

The Correlation Clustering problem for general (ordinary) graphs was introduced and proved NP-hard by Bansal *et al.* [2, 3]. Given a complete graph with labels $\langle + \rangle$ or $\langle - \rangle$ on each edge, the problem is to partition the vertices into clusters so that the number of $\langle - \rangle$ edges inside each cluster plus the number of $\langle + \rangle$ edges between the clusters, is minimized. Taking $\langle + \rangle$ edges as *edges* and $\langle - \rangle$ edges as *non-edges*, this problem is equivalent to the Cluster Editing problem, where we are given an ordinary graph graph and asked to add and delete the total minimum number of edges so that the resulting graph is a collection of disconnected (i.e., vertex-disjoint) cliques. The Correlation Clustering problem has been proven NP-hard several times, as it has been discovered and rediscovered in various applications areas, such as hierarchical tree clustering [23], computational biology [4, 30], and phylogenetic trees [8]. General versions of the Correlation Clustering problem have been defined and studied from the point of view of approximation [7, 10, 11, 13]. The second problem that we study in this paper is the Edge Clique Partitioning problem, which asks to partition the edges of a given graph into the minimum number of edge-disjoint cliques. This problem is NP-hard [28] for general graphs, but also for K_4-free and even chordal graphs [24].

[†]Department of Information and Computing Sciences, Utrecht University, The Netherlands. `hansb@cs.uu.nl`

[‡]PCRU, Office of DVC (Research), University of Newcastle, Australia. `michael.fellows@newcastle.edu.au`, `frances.rosamond@newcastle.edu.au`

[§]Department of Informatics, University of Bergen, Norway. `pinar@ii.uib.no`, `federico@ii.uib.no`, `charis@ii.uib.no`

In a general way, one can view the problems we consider here, the CORRELATION CLUSTERING (equivalent to CLUSTER EDITING) problem, and the EDGE CLIQUE PARTITIONING problem, as belonging to a loose class of problems, having to do with "clique-structuring" of graphs by means of editing or covering operations. For rhetorical convenience, we will refer to this loose class of problems as *GRAPH CLUSTERING PROBLEMS*. This class of problems, in which we would also include VERTEX COVER, CLIQUE COVER and many others, has proved to be a highly productive source of practical applications for parameterized algorithms [1, 9, 20].

A key point of what we offer here is to expand the investigation of *GRAPH CLUSTERING PROBLEMS* to inputs consisting of *fuzzy graphs*, where some pairs of vertices of the input may have an *undetermined*, *unknown* or *undecided* relation. For many applications, this clearly adds to the realism of the modeling in an important way. To mention one application area where a similar idea has been considered before, in bioinformatics, the notion of "sandwich graph problems" has played a useful role [18].

NP-hard problems remain hard also on fuzzy graphs, as they are a generalization of ordinary graphs. Hence we investigate their tractability from a parameterized complexity point of view, and we try to understand which structural parameters are more suitable to attack problems on fuzzy graphs.

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 $O(n^c \cdot f(k))$, where f is a computable function and c is a fixed constant [12]. 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 [12, 27].

The fixed-parameter tractability of the CLUSTER EDITING problem (for ordinary non-fuzzy graphs) has been shown, with a series of improvements in [6, 19, 29], when using the editing cost k as parameter. The problem has also been shown to admit a linear kernelization [15, 21]. On fuzzy graphs, it is not known whether using only k as parameter, ensures fixed parameter tractability. Here we introduce a new parameter r, that represents the minimum number of vertices required to cover the undecided edges. By parameterizing the CLUSTER EDITING problem by (k, r), we show that the problem admits a quadratic kernel, specifically on $O(k^2 + r)$ vertices, and therefore FPT also for fuzzy graphs. Furthermore the results hold also when the fuzzy graph is weighted.

The EDGE CLIQUE PARTITIONING problem has been recently shown to be FPT in [26], when parameterized by the number k of cliques that the edges can be partitioned into. In their work the authors give a quadratic kernel for it. The corresponding parameterization on fuzzy graphs asks, given a fixed k, 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. We prove that the problem becomes hard when the input is a fuzzy graphs, namely NP-complete for any fixed $k \geq 3$. Parameterizing only by k is thus not enough to ensure fixed-parameter tractability. However, if we parameterize by (k, r), where r is again the minimum number of vertices required to cover the undecided edges of the fuzzy graph, then the problem becomes FPT, and admits a polynomial time kernelization to a kernel on $O(k^4 \cdot 3^r)$ vertices.

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 pairs 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 edit 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)
Instance: 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 $k \geq 0$.
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 $\sum_{uv \in M} w(uv) \leq k$?

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 [13], but we restate it in a form more suitable for our framework and we give a constructive proof.

We define a *fuzzy path* $P_l^f = \{v_1, v_2, ..., v_l\}$ to be a fuzzy graph where for every $1 \leq i \leq l-1$, $v_i v_{i+1}$ is a real edge, $v_1 v_l$ is a non-edge, and all the other pairs of vertices are joined by fuzzy edges (see Figure 1).
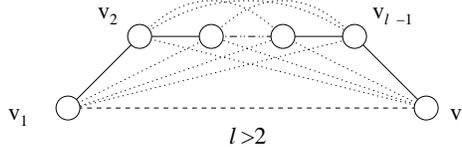

Figure 1: The family of fuzzy paths. Real edges are represented by continuous lines, non-edges by dashed lines, and fuzzy edges by dotted lines.

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

Proof. Let G contain a $P_l^f = \{v_1, v_2, ..., v_l\}$ 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 $v_1, v_2, ..., v_l$ 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 $v_1 v_l$, 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 $\{u, p_1, p_2, ..., v\}$ 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 a 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 $G[P_{ij}]$ is a fuzzy path, concluding the proof. ∎

The k-Weighted Fuzzy Cluster Editing problem (k-WFCE) is the WFCE problem where we choose k of the problem instance to be the parameter. The complexity of k-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 k-WFCE does not follow from the results of Cai [6].

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 (k, r)-Weighted Fuzzy Cluster Editing, or (k, 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 (k, r)-Weighted Fuzzy Cluster Editing problem

We show fixed-parameter tractability by giving a set of rules that either enable us to answer NO, or produce a kernel of size $O(k^2 + r)$ in polynomial time, for the (k, 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 k if and only if each connected component C_i can be made into a cluster graph with editing cost at most k_i, such that $\sum_{1 \le i \le l} k_i \le k$.*

Proof. If each connected component C_i can be made into a cluster graph with editing cost at most k_i such that $\sum_{1 \le i \le l} k_i \le k$, then G can be made into a cluster graph G' with editing cost at most k, 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 k. 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 k, and every cluster of G'' contains only vertices from the same connected component of G. Consequently, there must exist a value k_i such that $G[C_i]$ can be made into a cluster graph, namely $G''[C_i]$, with at most k_i as the editing cost, where $\sum_{1 \le i \le l} k_i \le k$, proving the theorem. ∎

Now we start presenting the rules, that are mostly self-explanatory. We will not give sharp bounds on the running time of each rule, but we will limit the explanation to why they can be executed in polynomial time. Our main goal is to prove that there exists a quadratic kernel that can be computed in polynomial time, and therefore the (k, r)-WFCE problem is FPT.

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

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, 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 $k+1$ 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 k disjoint fuzzy paths, the result follows. Checking the number of connected components of a graph can be obtained by a linear-time graph traversal. ∎

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. The idea is that, if two vertices cannot be disconnected deleting edges of total weight at most k, then they must belong to the same cluster in every solution, if any exists. 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.

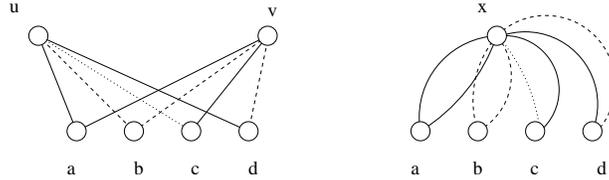

Figure 2: An example of contraction of two vertices u and v, with the resulting parallel edges. Real edges are represented by continuous lines, non-edges by dashed lines, and fuzzy edges by dotted lines.

Rule 3.1.3 *If there are vertices u and v such that the value of a minimum cut between them is at least $k+1$, then contract u and v into one vertex x and do all of the following:*

1. *If uv was a non-edge then let $k = k - w(uv)$.*

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 $f = ax$ then:*

   (a) *If $w(e) > w(f)$, then subtract let $k = k - w(f)$ 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 let $k = k - w(e)$ 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 let $k = k - w(e)$ and replace e and f with a fuzzy edge g.*

If now $k < 0$, 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 let $G' = ((V \setminus \{u, v\}) \cup \{x\}, E', F') = (V', E', F')$ and k' be the new graph and the 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 k if and only if we can obtain a cluster graph $H' = (V', E_{H'})$ from G' with editing cost at most k'.

5

Assume we can make a cluster graph $H = (V, E_H)$ from G with at most $k_T \leq k$ edge modifications. Since we cannot separate u from v by deleting edges of total weight less than $k + 1$, 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 k_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 k_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 $k_3 = \sum_{z \in Z_u^+ \cup Z_u^-} w(zu) - \min\{w(zu), w(zv)\} + \sum_{z \in Z_v^+ \cup Z_v^-} w(zv) - \min\{w(zu), w(zv)\}$. Notice at this point, that $k' = k - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$, where the $"-w(uv)"$ is in parenthesis because it might be there or not, according to case 1 of the rule. In other words $k - k'$ is a lower bound on the editing cost of G. Hence if $k' < 0$, then the editing cost is at least $k - k' > k$ and there cannot be a solution for G. Therefore we can safely assume $k' \geq 0$ and $k_T \geq \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$, i.e. $k_T - \sum_{z \in Z} \min\{zu, zv\}(-w(uv)) \geq 0$. Now we are ready to show that $k'_T = k_1 + k_2 + k_3 \leq k'$. By construction $k' = k - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$ and being $k \geq k_T$, we have that $k' \geq k_T - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$. By our previous discussion $k_T = k_1 + k_2 + \sum_{z \in Z_u^+ \cup Z_u^-} w(zu) + \sum_{z \in Z_v^+ \cup Z_v^-} w(zv)(+w(uv))$. Putting the two things together we get $k' \geq k_1 + k_2 + \sum_{z \in Z_u^+ \cup Z_u^-} w(zu) + \sum_{z \in Z_v^+ \cup Z_v^-} w(zv)(+w(uv)) - \sum_{z \in Z} \min\{zu, zv\}(-w(uv))$. This translates into $k' \geq k_1 + k_2 + k_3 = k'_T$ 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 $k'_T \leq k'$ 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 $k_T \leq k$.

The argument is very similar to the one for the previous direction, so we use the same notation as well. The costs k'_1, k'_2 are defined as k_1 and k_2 in the previous part of the proof, and they are the same for both G and G'. We redefine $k'_3 = \sum_{z \in Z} w(zx)$ 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 $k = k' + \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$ and $k' \geq k'_T$, we get $k \geq k'_T + \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$. By definition $k'_T = k'_1 + k'_2 + k'_3$, and replacing this in the inequality, we obtain $k \geq k'_1 + k'_2 + \sum_{z \in Z} w(zx) + \sum_{z \in Z} \min\{zu, zv\}(+w(uv))$, that is exactly k_T. Hence $k_T \leq k$, 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 [17], 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 $k^2 + 3k + r$ 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 k 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 $k + 1$.

Let us now consider a connected fuzzy graph $G = (V, E, F)$ with no clique of size greater than $k + 1$ and that can be made into a cluster graph by editing a set of real edges and non-edges M of total weight at least 1 and at most k. Then we show that G cannot have more than $k^2 + 3k + r$ vertices. Let us define: $S \subseteq V$ the set of vertices that are incident to some edited edge or non-edge in M; R a minimum fuzzy vertex cover of G; and $X = V \setminus R$, so that $G[X]$ does not contain any fuzzy edge. We define also $X' = S \cap X$. It is easy to see that $|X'| \leq |S| \leq 2k$. Let us focus on the graph $G[X \setminus X']$. It does not contain fuzzy edges, and none of its vertices is incident to an edited real edge or non-edge. We can conclude that it must be a union of disjoint cliques. In particular we show that it must be the union of at most $k + 1$ disjoint cliques, and that each of them has specific neighbors in the rest of the graph. 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 cluster in the solution. However, to create more than $k + 1$ connected components from a connected graph, we need to remove at least $k + 1$ edges. Hence the first part of the claim is proved. From the previous argument, it also follows that all vertices in X' connected to a clique in $G[X \setminus X']$, must end up in the same cluster of the solution as the clique they are adjacent to. Assume they do not, then we should delete an edge incident to a vertex in $G[X \setminus X']$, getting a contradiction. Therefore every vertex in X' can be connected to at most one clique in $G[X \setminus X']$, and furthermore it must be adjacent to all vertices of this clique. This means that every clique in $G[X \setminus X']$ has either some neighbors in X' and size at most k, or it has neighbors only in R and size at most $k + 1$. Going back to G, 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: $|V| \leq (k+1) \cdot N + k \cdot (k+1-N) + (2k-N) + |R| = k^2 + 3k + r$. 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 $(2k - N)$ represents a tighter bound on $|X'|$. In fact, for every clique with neighbors only in R, there must be at least one 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 $1 \leq l \leq k$. By Observation 3.2 we know that there is a solution for G that edits at most k edges if and only if there is a solution for each $G[C_i]$ that edits at most k_i edges, such that $\sum_{i=1}^{l} k_i \leq k$. This means that, by what we just proved for connected fuzzy graphs, if there is a solution for G then $|V(G[C_i])| \leq k_i^2 + 3k_i + r_i$ for $1 \leq i \leq l$, where r_i is the size of a minimum vertex cover of $G[C_i]$. Hence $|V(G)| = \sum_{i=1}^{l} k_i^2 + 3k_i + r_i$, that is $\sum_{i=1}^{l} k_i^2 + 3 \sum_{i=1}^{l} k_i + \sum_{i=1}^{l} r_i \leq (\sum_{i=1}^{l} k_i)^2 + 3k + r = k^2 + 3k + r$, completing the proof. ∎

It is easy to construct examples where we have $(k+1) \cdot N + k \cdot (k+1-N) + (k+1-N) + r = k^2 + 2k + 1 + r$ vertices in a *yes* instance for any given k; see Figure 3 for an example. Hence Theorem 3.6 gives a quite tight bound on the size of the kernel, that is in any case $O(k^2 + r)$.

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

Theorem 3.7 *The (k,r)-*WEIGHTED FUZZY CLUSTER EDITING *problem can be solved in time* $n^{O(1)} + O((k^2 + r)^{2k})$.

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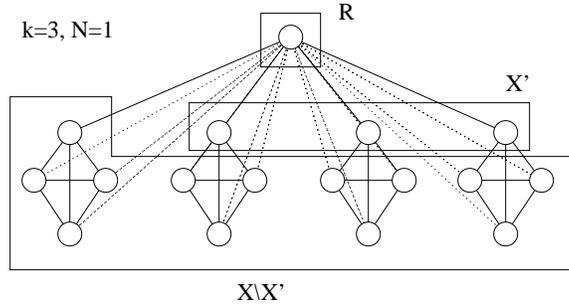
Figure 3: Example of a kernel with $k^2 + 2k + 1 + r$ vertices for $k = 3$. The non-edges are not drawn to keep the figure easier to observe.

only in n. After the preprocessing we either return that there is no solution or the current fuzzy graph $G = (V, E, F)$, has with at most $k^2 + 3k + r$ 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 subsets M of the set $(V \times V) \setminus F$ of total weight at most k. For each set M, we build a graph $G' = (V, (E \setminus M) \cup (M \setminus E), F)$. Since no more editing of real edges or non-edges is allowed for this graph, then this is a valid solution if and only if realizing the fuzzy edges inside each connected component into real edges, and the rest into non-edges, produces a cluster graph. If for some set M we get a cluster graph, we answer YES, else there is no solution. Since $|(V \times V) \setminus F| \leq \binom{k^2 + 3k + r}{2}$ and $|M| \leq k$, the result follows. ∎

4 Parameterized edge clique partition 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)
Instance: 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, for every $k \geq 3$. Recall that, in contrast, the Edge Clique Partitioning problem is FPT when parameterized by k. We show the FECP problem parameterized by both k and r, where r is again the size of a minimum fuzzy vertex cover, is FPT. We call this version of the problem (k, r)-FECP.

4.1 The k-Fuzzy Edge Clique Partitioning problem is NP-complete

Here we prove that deciding whether the edges of a fuzzy graph can be partitioned into at most k edge-disjoint cliques, is NP-complete for every fixed $k \geq 3$, and polynomial otherwise. The problem we reduce from, is the classical k-Coloring problem. In this problem, the input is a graph $G = (V, E)$ and a fixed $k > 0$, and the question is 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-complete for every fixed $k \geq 3$.

Theorem 4.1 *The k-*Fuzzy Edge Clique Partitioning *problem 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_iu_i, because we replace each $v_iv_j \in E$ 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 A_j is an independent set for each $1 \le j \le k$. 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 $1 \le j \le k$. 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-Edge Clique Partitioning problem on G', since vertex-disjoint cliques are also edge-disjoint and their union contains all edges.

Assume now that there exists a normalization H of G', that admits a partition $K = \{K_1, \ldots, K_l\}$ of its edges, so that $H(K_i)$ is a clique for each $1 \le i \le l$ and $l \le k$. By construction we know that the graph is not edgeless, so that $l \le 1$. Let us now build a solution for the k-Coloring problem for G using K. We associate a different color c_i to each element $K_i \in K$, and give to a vertex $v_j \in V$ the color c_i if the edge $v_ju_j \in K_i$. We claim that this gives a legal k-coloring. First of all there are at most k cliques in K, so at most k colors are used. Besides each edge v_ju_j is contained in a unique clique of K, so we have no ambiguity and all vertices of G are colored with a unique color. Assume now that there is at least one edge $v_iv_j \in E$ such that the endpoints v_i and v_j got the same color. This implies that v_iu_i and v_ju_j belong to the same clique of K. However, by construction, there should be a non-edge between v_i and v_j in G'. Hence there cannot be any normalization of G' where v_iu_i and v_ju_j belong to the same clique, and therefore to the same element of a valid edge clique partition. This proves our claim.

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

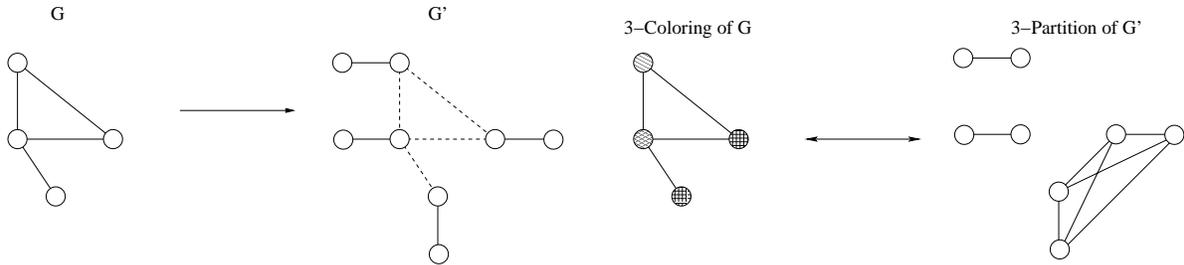Figure 4: An example of the graph G' obtained from a graph G, and the equivalence of a solution of the 3-Coloring problem on G with a solution of the 3-Edge Clique Partitioning problem on G'. In G' the fuzzy edges are not drawn to keep the figure clean, while in the corresponding clique partition the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

As a complemantary result, we prove next that the k-FECP problem is solvable in polynomial time when k is 0, 1, or 2. For the following results, we define an *isolated vertex of a fuzzy graph* to be a vertex that is not incident to any real edge. Similarly, we define a *universal vertex of a fuzzy graph* to be a vertex that is not incident to any non-edge.

Theorem 4.2 *The 0-FECP and the 1-FECP problems are solvable in polynomial time.*

Proof. The 0-FECP problem is solvable if and only if there is a normalization of the input fuzzy graph that is edgeless. In other words if the input graph has no real edges.

The 1-FECP problem is solvable if and only if there is a normalization of the input where the set of all edges induces a complete graph. This is equivalent to checking whether there is any non-edge in the input graph after we remove all isolated vertices. If there is, then there is no normalization satisfying the desired conditions. In fact there would be two vertices incident to some real edges, that cannot be in a clique together, i.e., the endpoints of the non-edge. If there is not any non-edge, instead, we can turn all fuzzy

edges into real edges and get a complete graph, while we turn all fuzzy edges incident to isolated vertices into non-edges, so that they are still isolated in the normalization.

In both cases the conditions can clearly be checked in polynomial time. ∎

For the 2-FECP problem, we need a few more details, and we start with an observation to formalize some simple facts.

Observation 4.3 *Let $G = (V, E, F)$ be a fuzzy graph with a normalization H that admits a minimum edge clique partition $K = \{K_1, K_2\}$, and let $V_1 = V(H(K_1))$ and $V_2 = V(H(K_2))$. Then either $V_1 \cap V_2 = \emptyset$ or $V_1 \cap V_2 = \{x\}$ for some $x \in V$. In the latter case, $H - x$ is a normalization of $G - x$ with an edge clique partition K' such that $|K'| \leq 2$, and if $K' = \{K_1', K_2'\}$ then $K_1' = K_1 \setminus \{xy \mid y \in K_1\}$, $K_2' = K_2 \setminus \{xy \mid y \in K_2\}$, and $V(H(K_1')) \cap V(H(K_2')) = \emptyset$.*

Proof. The edges of $H - x$ can be partitioned in at most 2 cliques. In fact, when removing x, and therefore all edges incident to it, the remaining edges in the graph are completely contained in $V_1 \setminus \{x\}$ and $V_2 \setminus \{x\}$. Furthermore, since V_1 and V_2 are cliques, $V_1 \setminus \{x\}$ and $V_2 \setminus \{x\}$ are also cliques. What might happen is that $V_1 \setminus \{x\}$ or $V_2 \setminus \{x\}$ (or both) consists of a single isolated vertex. Therefore the edges of $H - x$ might be partitioned into 0, 1, or 2 cliques. Finally, as $V(H(K_1')) = V_1 \setminus \{x\}$ and $V(H(K_2')) = V_2 \setminus \{x\}$, it is obvious that $V(H(K_1')) \cap V(H(K_2')) = \emptyset$. ∎

Theorem 4.4 *The 2-FECP problem can be solved in polynomial time.*

Proof. We describe a polynomial-time algorithm to solve the problem. Let $G = (V, E, F)$ be the fuzzy input graph.

First we describe how to solve a special case of the 2-FECP, namely the problem of checking whether there is a normalization of G whose vertices and edges can be partitioned into at most 2 vertex-disjoint cliques. We call this problem 2-Fuzzy Edge-Vertex Clique Partitioning problem, or 2-FEVCP. Checking whether such a normalization exists is equivalent to checking whether the connected components of G can be partitioned in at most two sets C_1 and C_2 such that $G[C_1]$ and $G[C_2]$ do not contain any non-edge. This can be done as follows. First check whether any connected component of G contains non-edges. If so, answer NO. This is a necessary condition for the problem to have a solution, but not sufficient. If no connected component contains non-edges, create a graph G' that has a vertex for each connected component of G, and an edge between two vertices if and only if there is at least one non-edge between the c orresponding connected components. It is easy to see that our problem is the same as asking whether G' is 2-colorable. Thus the 2-FEVCP is solvable in polynomial time.

Now let G^* be our input graph without isolated vertices. There is a solution to the 2-FECP problem for G if and only if there is for G^*. If there is a solution for G^*, i.e., a normalization H^* whose edges can be partitioned in at most 2 cliques, then there is a solution for G, just turning all fuzzy edges incident to isolated vertices into non-edges, and adding them as isolated vertices in H^*. Also, if there is a solution for G, then removing all isolated vertices of G from H, yields a solution for G^*.

We show how to solve the 2-FECP problem on G^*, by using the algorithm for the 2-FEVCP problem as a subroutine. First we run the algorithm for 2-FEVCP on G^*. Notice that a solution to the 2-FEVCP, is also a solution for the 2-FECP, but not vice versa. In particular, if we get a NO answer to the 2-FEVCP, we know that if there is a solution to the 2-FECP problem on G^*, it consists of at least two cliques, intersecting each other in one vertex for some normalization H^* of G^*. Furthermore, this vertex would be universal in G^*. Thus, by Observation 4.3, G^* has a solution if and only if there exists a vertex $v \in V(G^*)$ such that $G^* - v$ is a YES instance for the 2-FEVCP problem and v is universal in G. Notice that when we run the 2-FEVCP algorithm on $G^* - v$, we do not remove isolated vertices, because they were not isolated in G^*. Hence they must be part of the vertices that must be partitioned in $G^* - v$, or we might get false positives.

Since the algorithm for the 2-FECP problem essentially consists of calling at most $n + 1$ times the 2-FEVCP algorithm as a subroutine, it runs in polynomial time, proving the theorem. ∎

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

To obtain a kernel for (k, r)-FECP, we first give some observations that apply to any valid solution of the problem on non-fuzzy graphs, i.e., the k-EDGE CLIQUE PARTITIONING problem.

For a non-fuzzy graph $G = (V, E)$, and a fixed $k \geq 0$, we call a *feasible solution* a partition $K = \{K_1, K_2, ..., K_l\}$ of E such that $G(K_i)$ is a clique for each i, and $l \leq k$. 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 *gateways* the vertices that are in the intersection of some cliques defined by elements of K, 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 4.5 ([5]) *Every edge clique partition of a complete graph on n vertices, except the trivial one of a single clique, contains at least n cliques.*

Lemma 4.6 *If the answer to the k-EDGE CLIQUE PARTITIONING 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 $K = \{K_1, \ldots, K_l\}$ 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 feasible 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, but we removed all edges incident to the vertices of V'. Finally $|K'| \leq |K| \leq k$. This concludes the proof. ∎

Lemma 4.6 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 4.7 *In any solution of k-EDGE CLIQUE PARTITIONING, there cannot be more than $\binom{k}{2}$ gateway vertices.*

Proof. Every two cliques defined by a feasible solution 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 $K = \{K_1, K_2, \ldots, K_k\}$ be a partition of the edges of a graph G into k 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 4.8 *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 feasible solution $K = \{K_1, K_2, ..., K_l\}$ of 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,
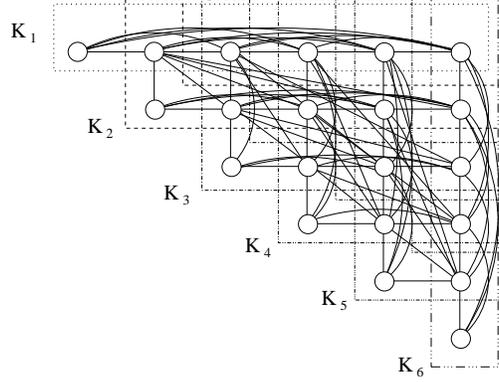
11

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.

where z can be also equal to i or j. In this case we would have $u \in V(K_i) \cap V(K_z)$ or $v \in V(K_j) \cap V(K_z)$, or both, meaning that at least one of them is a gateway vertex, 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 $V(K_z)$. ∎

Observation 4.9 *If there are more than $k + \binom{k}{2}$ vertices of pairwise different type, then the answer to k-EDGE CLIQUE PARTITIONING is* NO.

Proof. By Observation 4.8 we know that all vertices that can belong to the same clique in a feasible solution, must either have the same type or be gateways. By Observation 4.7 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 observations given until now, can be used as rules to give a polynomial time kernelization for the (k, r)-FECP problem. From now on we assume a fuzzy 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 4.6.

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

12

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 4.9, there is no solution for $G[X]$. By Lemma 4.6, 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 4.11 *Rule 4.2.2 is correct and can be executed in polynomial time.*

Proof. Let 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 feasible solution K'. Since in $G - u$ there are at least $\binom{k}{2} + 1$ vertices 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 4.7, in fact, there cannot be more than $\binom{k}{2}$ gateways in any solution. This means that if we realize the fuzzy edges of $G[V \setminus \{u\}]$ as in H and the fuzzy edges incident to u as the fuzzy edges incident to v in $G - u$, we get a normalization H' of G where u and v have the same type. Hence a feasible solution for H' 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 4.6, if there is a normalization G' of G with a feasible solution, then there is also a normalization of $G - u$ that has a feasible solution, namely $G' - u$. ∎

Lemma 4.12 *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 4.13 (k, r)-Fuzzy Edge Clique Partitioning *is FPT with a kernel of size $O(k^4 \cdot 3^r)$.*

Proof. The size of the kernel follows from Lemma 4.12, so we only need to show that the preprocessing can be performed in polynomial time. Rules 4.2.1 and 4.2.2 can be applied in polynomial time by Lemma 4.10 and 4.11. Besides they are applied at most a polynomial number of times. In fact, every time we apply Rule 4.2.2 we either remove one vertex or stop, and Rule 4.2.1 needs to be applied only once before every application of Rule 4.2.2. In total we can have at most $2n$ application of the rules, hence the theorem follows. ∎

5 Concluding remarks

In this paper we have studied the parameterized complexity of two important examples of *GRAPH CLUS- TERING PROBLEMS* on inputs consisting of fuzzy graphs: graphs that represent incomplete information about relationships. We believe that the investigation of "problems on fuzzy graphs" is extremely well-motivated by applications, particularly in areas such as machine learning and bioinformatics, where complete information about the graphs modeling various computational objectives is often not available. In this general context, much more remains to be done.

We have described two FPT algorithms, respectively, for the Weighted Fuzzy Cluster Editing problem, and the Fuzzy Edge Clique Partitioning problem, where both are parameterized by the compound parameter (k, r), and where k is a *cost parameter*: respectively, the total cost of the editing in the case of Weighted Fuzzy Cluster Editing, and the number of cliques in the partition in the case of Fuzzy Edge Clique Partitioning; and where r is a *structural parameter*: the minimum number of vertices required to cover the undecided edges of the fuzzy graph taken as input. This structural parameter

could be well-motivated by applications where only a small number of "trouble-maker" vertices are the "cause" of the uncertain information about the input.

We have also shown that in the case of FUZZY EDGE CLIQUE PARTITIONING, it is not possible to extend the above positive outcome to a parameterization only by k.

In the case of the FUZZY CLUSTER EDITING problem, the analogous question remains open, and this is in fact a prominent concrete open problem in parameterized complexity. Apart from the important machine learning applications noted in [2, 3], it has recently been shown that for the special case where all weights are 1, the FUZZY CLUSTER EDITING problem (parameterized only by k) is FPT-equivalent [16, 13], to the MINIMUM TERMINAL EDGE SEPARATION problem left open by Marx in [25].

Another area of open problems concerning this work is that of *improving kernelization bounds*. Because FPT kernelization is of great practical significance due to the general connection to efficient pre-processing (see [14, 22, 27] for background and discussion of this point), it is an outstanding open problem as to whether FUZZY EDGE CLIQUE PARTITIONING admits a $Poly(k, r)$ kernelization.

References

[1] F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *Proceedings of ALENEX/ANALC'04 - 6th Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithmics and Combinatorics*, pages 62–69, 2004.

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

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

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

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

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

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

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

[9] F. K. H. A. Dehne, M. A. Langston, X. Luo, S. Pitre, P. Shaw, and Y. Zhang. The cluster editing problem: Implementations and experiments. In *Proceedings of IWPEC'06 - 2nd International Workshop on Parameterized and Exact Computation*, pages 13–24. Springer, 2006.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

[16] M. R. Fellows, M. Mnich, F. Rosamond, and S. Saurabh. Manuscript. 2008.

[17] 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.

[18] M. C. Golumbic, H. Kaplan, and R. Shamir. Graph sandwich problems. *J. Algorithms*, 19(3):449–473, 1995.

[19] 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.

[20] J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Data reduction, exact, and heuristic algorithms for clique cover. In *Proceedings of the 8th ACM-SIAM ALENEX*, pages 86–94. ACM-SIAM, 2006.

[21] 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.

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

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

[24] S. H. Ma, W. D. Wallis, and J. L. Wu. The complexity of the clique partition number problem. *Congr. Numer.*, 67:59–66, 1988.

[25] D. Marx. Parameterized graph separation problems. In *Proceedings of IWPEC'04 - 1st International Workshop on Parameterized and Exact Computation*, pages 71–82. Springer, 2004.

[26] 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.

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

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

[29] 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.

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