Graph-Based Data Clustering with Overlaps

Michael R. Fellows[1⋆], Jiong Guo[2], Christian Komusiewicz[2⋆⋆], Rolf Niedermeier[2], and Johannes Uhlmann[2⋆⋆⋆]

[1] PC Research Unit, Office of DVC (Research), University of Newcastle, Callaghan, NSW 2308, Australia.
michael.fellows@newcastle.edu.au
[2] Institut für Informatik, Friedrich-Schiller-Universität Jena Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{jiong.guo,c.komus,rolf.niedermeier,johannes.uhlmann}@uni-jena.de

Abstract. We introduce overlap cluster graph modification problems where, other than in most previous work, the clusters of the target graph may overlap. More precisely, the studied graph problems ask for a minimum number of edge modifications such that the resulting graph consists of clusters (maximal cliques) that may overlap up to a certain amount specified by the overlap number s. In the case of *s-vertex overlap*, each vertex may be part of at most s maximal cliques; *s-edge overlap* is analogously defined in terms of edges. We provide a complete complexity dichotomy (polynomial-time solvable vs NP-complete) for the underlying edge modification problems, develop forbidden subgraph characterizations of "cluster graphs with overlaps", and study the parameterized complexity in terms of the number of allowed edge modifications, achieving fixed-parameter tractability results (in case of constant s-values) and parameterized hardness (in case of unbounded s-values).

1 Introduction

Graph-based data clustering is an important tool in exploratory data analysis [21,25]. The applications range from bioinformatics [2,22] to image processing [24]. The formulation as a graph-theoretic problem relies on the notion of a *similarity graph*, where vertices represent data items and an edge between two vertices expresses high similarity between the corresponding data items. Then, the computational task is to group vertices into clusters, where a *cluster* is nothing but a dense subgraph (typically, a clique). Following Ben-Dor et al. [2], Shamir et al. [21] initiated a study of graph-based data clustering in terms of *graph modification* problems. More specifically, here the task is to modify (add or delete) as few edges of an input graph as possible to obtain a *cluster graph*,

⋆ Supported by the Australian Research Council. Work done while staying in Jena as a recipient of a Humboldt Research Award of the Alexander von Humboldt Foundation, Bonn, Germany.
⋆⋆ Supported by a PhD fellowship of the Carl-Zeiss-Stiftung.
⋆⋆⋆ Supported by the DFG, research project PABI, NI 369/7.

that is, a *vertex-disjoint* union of cliques. Numerous recent publications build on this concept of cluster graphs, e.g., [4,7,9,11,13,20]. To uncover the *overlapping* community structure of complex networks in nature and society [18], however, the concept of cluster graphs so far fails to model that clusters may overlap, and it has been criticized explicitly for this lack of overlaps [7]. In this work we introduce a graph-theoretic relaxation of the concept of cluster graphs by allowing, to a certain degree, overlaps between the clusters (which are cliques). We distinguish between "vertex overlaps" and "edge overlaps" and provide a thorough study of the corresponding cluster graph modification problems.

The two core concepts we introduce are *s-vertex overlap* and *s-edge overlap*, where in the first case we demand that every vertex in the cluster graph is contained in at most s maximal cliques and in the second case we demand that every edge is contained in at most s maximal cliques. Clearly, 1-vertex overlap actually means that there is no overlap between the cliques (clusters). Based on these definitions, we study a number of edge modification problems (addition, deletion, editing) in terms of the two overlap concepts, generalizing and extending previous work that focussed on non-overlapping clusters.

Previous work. Perhaps the best studied cluster graph modification problem is the NP-hard Cluster Editing, where one asks for a minimum number of edges to add or delete in order to transform the input graph into a disjoint union of cliques. Cluster Editing has been intensively studied from a theoretical [1,3,9,11,13,20] as well as a practical side [4,7]. The major part of this work deals with the parameterized complexity of Cluster Editing, having led to efficient search-tree based [3,11] and polynomial-time kernelization [9,13,20] algorithms. One motivation of our work is drawn from these intensive studies, motivated by the practical relevance of Cluster Editing and related problems. As discussed before, however, Cluster Editing forces a sometimes too strict notion of cluster graphs by disallowing any overlap. To the best of our knowledge, relaxed versions of Cluster Editing have been largely unexplored. The only approach studying overlapping cliques in the context of Cluster Editing that we are aware of has been presented by Damaschke [6]. He investigated the Twin Graph Editing problem, where the goal is to obtain a so-called *twin graph* (with a further parameter t specified as part of the input) with a minimum number k of edge modifications. A t-twin graph is a graph whose "critical clique graph" has at most t edges, where the critical clique graph is the representation of a graph that is obtained by keeping for each set of vertices with identical closed neighborhoods exactly one vertex. Roughly speaking, our model expresses a more local property of the target graph. The main result of Damaschke [6] is fixed-parameter tractability with respect to the combined parameter (t, k). We note that already for $s = 2$ our s-vertex overlap model includes graphs whose twin graphs can have an unbounded number t of edges. Hence, s is not a function of t. Moreover, we expect that for many real-world graphs the number k of necessary edge modifications is much smaller in our model than in the one of Damaschke.

Our results. We provide a thorough study of the computational complexity of clustering with vertex and edge overlaps, significantly extending previous work on Cluster Editing and closely related problems. In particular, in terms of the overlap number s, we provide a complete complexity dichotomy (polynomial-time solvable versus NP-complete) of the corresponding edge modification problems, most of them turning out to be NP-complete (see Table 1 in Section 3). For instance, somewhat surprisingly, whereas Cluster Editing restricted to only allowing edge additions (also known as Cluster Addition or 1-Vertex Overlap Addition) is trivially solvable in polynomial time, 2-Vertex-Overlap Addition turns out to be NP-complete. We also study the parameterized complexity of clustering with overlaps. On the negative side, we show W[1]-hardness results with respect to the parameter "number of edge modifications" in case of unbounded overlap number s. On the positive side, we prove that the problems become fixed-parameter tractable for every constant s. This result is based on forbidden subgraph characterizations of the underlying overlap cluster graphs, which may be of independent graph-theoretic interest. Indeed, it turns out that the "1-edge overlap cluster graphs" are exactly the diamond-free graphs. Finally, we develop polynomial-time data reduction rules for two special cases. More precisely, we show an $O(k^4)$-vertex problem kernel for 1-Edge Overlap Deletion and an $O(k^3)$-vertex problem kernel for 2-Vertex Overlap Deletion, where both times k denotes the number of allowed edge modifications. We conclude with a number of open problems.

Preliminaries. Given a graph $G = (V, E)$, we use $V(G)$ to denote the vertex set of G and $E(G)$ to denote the edge set of G. Let $n := |V|$ and $m := |E|$. The (open) neighborhood $N(v)$ of a vertex v is the set of vertices that are adjacent to v, and the closed neighborhood $N[v] := N(v) \cup \{v\}$. We use $G[V']$ to denote the subgraph of G induced by $V' \subseteq V$, that is, $G[V'] := (V', \{\{u, v\} \mid u, v \in V', \{u, v\} \in E\})$. Moreover, $G - v := G[V \setminus \{v\}]$ for a vertex $v \in V$ and $G - e := (V, E \setminus \{u, v\})$ for an edge $e = \{u, v\}$. For two sets E and F let $E \Delta F := (E \setminus F) \cup (F \setminus E)$ (symmetric difference). For a set X of vertices let $E_X := \{\{u, v\} \mid u, v \in X, u \neq v\}$ denote the set of all possible edges on X. Furthermore, for a graph $G = (V, E)$ and a set $S \subseteq E_V$ let $G \Delta S := (V, E \Delta S)$ denote the graph that results by modifying G according to S. A set of pairwise adjacent vertices is called *clique*. A clique K is a *critical clique* if all its vertices have the same neighborhood and K is maximal. A *graph property* is defined as a nonempty proper subset of the set of graphs closed under graph isomorphism. A *hereditary* graph property is a property closed under taking induced subgraphs.

For a graph property π, the π Editing problem is defined as follows.

Input: A graph $G = (V, E)$ and an integer $k \geq 1$.
Question: Does there exist a set $S \subseteq V \times V$ with $|S| \leq k$ such that $G \Delta S$ has property π?

In this paper, we focus attention on π being either the s-vertex overlap property or the s-edge overlap property (see Definition 1 in Section 2). The set S is called a solution. Moreover, we say that the vertices that are incident to an

edge in S are *affected* by S and that all other vertices are *non-affected*. In the corresponding π Deletion (or π Addition) problem, only edge deletion (or addition) is allowed.

Parameterized complexity is a two-dimensional framework for studying the computational complexity of problems [8,10,17]. One dimension is the input size n (as in classical complexity theory), and the other one is the *parameter* k (usually a positive integer). A problem is called *fixed-parameter tractable* (fpt) if it can be solved in $f(k) \cdot n^{O(1)}$ time, where f is a computable function only depending on k. This means that when solving a combinatorial problem that is fpt, the combinatorial explosion can be confined to the parameter. A core tool in the development of fixed-parameter algorithms is polynomial-time preprocessing by *data reduction*. Here, the goal is for a given problem instance x with parameter k to transform it into a new instance x' with parameter k' such that the size of x' is upper-bounded by some function only depending on k, the instance (x, k) is a yes-instance iff (x', k') is a yes-instance, and $k' \leq k$. The reduced instance, which must be computable in polynomial time, is called a *problem kernel*, and the whole process is called *reduction to a problem kernel* or simply *kernelization*.

Downey and Fellows [8] developed a formal framework to show *fixed-parameter intractability* by means of *parameterized reductions*. A parameterized reduction from a parameterized language L to another parameterized language L' is a function that, given an instance (x, k), computes in $f(k) \cdot n^{O(1)}$ time an instance (x', k') (with k' only depending on k) such that $(x, k) \in L \Leftrightarrow (x', k') \in L'$. The basic complexity class for fixed-parameter intractability is called $W[1]$ and there is good reason to believe that $W[1]$-hard problems are not fpt [8,10,17].

Due to the lack of space, most proofs are deferred to the full version of this article.

2 Forbidden Subgraph Characterization

In this section, we first introduce the two graph properties considered in this work. Then, we present induced forbidden subgraph characterizations for graphs with these properties.

Definition 1 (s-vertex-overlap property and s-edge-overlap property). *A graph $G = (V, E)$ has the* s-vertex-overlap property *(or* s-edge-overlap property*) if every vertex (or edge) of G is contained in at most s maximal cliques.*

Clearly, a graph having the 1-vertex-overlap property consists of a vertex-disjoint union of cliques. See Fig. 1 for a graph fulfilling the 2-vertex-overlap and the 1-edge-overlap property.

Given a graph and a non-negative integer s, we can decide in polynomial time whether G fulfills the s-vertex-overlap property using a clique enumeration algorithm with polynomial delay. For each $v \in V$, we enumerate the maximal cliques in $G[N[v]]$. We abort the enumeration if we have found $s + 1$ maximal cliques. Using for example a polynomial delay enumeration algorithm by Makino and Uno [16] that relies on matrix multiplication and enumerates cliques with

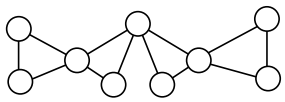

Fig. 1. An example with the 2-vertex overlap and 1-edge-overlap properties.

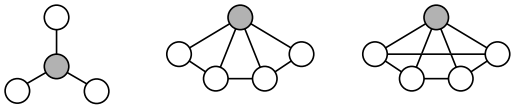

Fig. 2. Forbidden induced subgraphs for the 2-vertex-overlap property. In every graph, the gray vertex is contained in at least three maximal cliques.

delay $O(n^{2.376})$, the overall running time of this algorithm is $O(s \cdot n^{3.376})$. For the edge case, a similar approach applies. The only difference is that, here, we consider the common neighborhood of the endpoints of every edge, that is, $N[u] \cap N[v]$ for an edge $\{u, v\}$.

Theorem 1. *Given a graph G and a non-negative integer s, it can be decided in time $O(s \cdot n^{3.376})$ (or $O(s \cdot m \cdot n^{2.376})$) time whether G has the s-vertex-overlap (or s-edge-overlap) property.*

The next lemma proves the existence of induced forbidden subgraph characterizations for graphs having the s-vertex-overlap or the s-edge-overlap property.

Lemma 1. *The s-vertex-overlap property and the s-edge-overlap property are hereditary.*

Hereditary graph properties can be characterized by a finite or infinite set of forbidden subgraphs [12]. Thus, by Lemma 1, such a characterization must exist. Here, we show that the forbidden subgraphs have size $O(s^2)$ and, hence, that for fixed s the number of forbidden induced subgraphs is finite. Furthermore, we can find a forbidden induced subgraph in polynomial time.

Theorem 2. *Given a graph G that violates the s-vertex-overlap (or s-edge-overlap) property, one can find in time $O(s \cdot n^{3.376} + s^2 \cdot n)$ (or $O(s \cdot m \cdot n^{2.376} + s^2 \cdot n)$) time a forbidden induced subgraph of size $O(s^2)$.*

See Fig. 2 for the induced forbidden subgraphs for graphs with the 2-vertex-overlap property. Observe that many important graph classes are contained in the class of graphs with the s-overlap property. In particular, it is easy to see that the diamond-free graphs are equivalent to graphs with the 1-edge-overlap property, as stated in Lemma 2. A diamond is the graph that results by deleting one edge from a four-vertex clique. Diamond-free graphs, that is, graphs that contain no diamond as an induced subgraph, form a graph class studied for its own sake [23].

Lemma 2. *A graph G has the 1-edge-overlap property iff G is diamond-free.*

Table 1. Classical computational complexity of graph-based data clustering with overlaps. Herein, "NPC" means that the respective problem is NP-complete and "P" means that the problem can be solved in polynomial time.

	s-vertex-overlap	s-edge-overlap
Editing	NPC for $s \geq 1$	NPC for $s \geq 1$
Deletion	NPC for $s \geq 1$	NPC for $s \geq 1$
Addition	P for $s = 1$, NPC for $s \geq 2$	P for $s = 1$, NPC for $s \geq 2$

3 A Complexity Dichotomy with Respect to s

This section provides a complete picture of the computational complexity of the introduced problems. The results are summarized in Table 1.

Lemma 3 shows that if one of the problems is NP-hard for some $s \geq 1$, then it is NP-hard for every $s' \geq s$.

Lemma 3. *For $s \geq 1$, there is a polynomial-time many-one reduction from s-*Property Operation *to $(s+1)$-*Property Operation*, where* Property $\in$ { Vertex-Overlap, Edge-Overlap } *and* Operation $\in$ { Editing, Deletion, Addition }*.*

Since Cluster Editing and Cluster Deletion (equivalent to 1-Vertex-Overlap Editing and 1-Vertex-Overlap Deletion) are known to be NP-complete [15,21], we directly arrive at the following theorem.

Theorem 3. *s-*Vertex-Overlap Editing *and s-*Vertex-Overlap Deletion *are NP-complete for $s \geq 1$.*

1-Vertex-Overlap Addition is trivially polynomial-time solvable: one has to transform every connected component into a clique by adding the missing edges. In contrast, for $s \geq 2$, s-Vertex-Overlap Addition becomes NP-complete.

Theorem 4. *s-*Vertex-Overlap Addition *is NP-complete for $s \geq 2$.*

Proof. (Sketch) We present a polynomial-time many-one reduction from the NP-complete Maximum Edge Biclique problem [19] to 2-Vertex-Overlap Addition (2-VOA). Then, for $s \geq 2$, the NP-hardness follows directly from Lemma 3. The decision version of Maximum Edge Biclique is defined as follows: Given a bipartite graph $H = (U, W, F)$ and an integer $l \geq 0$, does H contain a biclique with at least l edges? A biclique is a bipartite graph with all possible edges.

The reduction from Maximum Edge Biclique to 2-VOA works as follows: Given a bipartite graph $H = (U, W, F)$, we construct a graph $G = (V, E)$, where $V := U \cup W \cup \{r\}$ and $E := E_{\overline{F}} \cup E_r \cup E_U \cup E_W$. Herein,

- $E_{\overline{F}} := \{\{u, w\} \mid u \in U, w \in W\} \setminus F$,

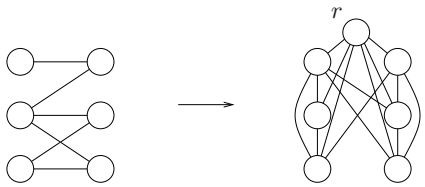

Fig. 3. Example for the reduction from Maximum Edge Biclique (left graph) to 2-Vertex-Overlap Addition (right graph).

- $E_X := \{\{x, x'\} \mid x, x' \in X, x' \neq x\}$ for $X \in \{U, W\}$, and
- $E_r := \{\{r, x\} \mid x \in U \cup V\}$.

That is, the graph $(U, V, E_{\overline{F}})$ is the bipartite complement of H, in G both U and V are cliques, and r is adjacent to all other vertices in G. See Fig. 3 for an illustration of this construction. The correctness proof is deferred to the full version of this article. □

Next, we consider the edge overlap case. First, observe that the reduction given in the proof of Theorem 4 can be easily modified to show the NP-hardness of 2-Edge-Overlap Addition: Simply replace the introduced vertex r by an edge e and connect both endpoints of e to all vertices in the given bipartite graph of the Maximum Edge Biclique. The correspondence between the solutions of both instances can be shown in complete analogy with the vertex overlap case. Note that 1-Edge-Overlap Addition is trivially solvable in polynomial time, since there exists only one possibility to destroy a diamond by adding edges; by Lemma 2, diamonds are the only forbidden subgraph of graphs having the 1-edge-overlap property.

Theorem 5. *s-Edge-Overlap Addition is NP-complete for $s \geq 2$.*

Finally, we can show that 1-Edge-Overlap Editing and Deletion are NP-complete by a reduction from Vertex Cover in cubic graphs. For $s > 1$, the NP-hardness follows directly by using Lemma 3.

Theorem 6. *s-Edge-Overlap Editing and s-Edge-Overlap Deletion are NP-complete for $s \geq 1$.*

4 Parameterized Complexity

Here, we consider the parameterized complexity of our overlap clustering problems. First, due to Theorem 2, we have a set of forbidden subgraphs for both properties whose size only depends on s. Thus, using a result of Cai [5], we can conclude that all three problems with overlap properties are fixed-parameter tractable with respect to the combined parameter (s, k).

Theorem 7. *π Editing, π Addition, and π Deletion problems with $\pi \in$ { s-Vertex-Overlap, s-Edge-Overlap } are fixed-parameter tractable with respect to the combined parameter (s, k).*

Next, we consider the parameterization with only k as the parameter. This means that s can have an unbounded value. To show W[1]-hardness, we develop a parameterized reduction from the W[1]-complete Set Packing problem [8].

Theorem 8. *s-Vertex(Edge)-Overlap Deletion(Editing) is W[1]-hard with respect to the parameter k in the case of unbounded s.*

5 Two Kernelization Results for Edge Deletion

Not surprisingly, all nontrivial overlap clustering problems we study here seem to become significantly more demanding than clustering without overlaps. Hence, to start with, we subsequently present two kernelization results for the two most basic NP-hard clustering problems with nontrivial overlaps. We defer the correctness proofs of the considered data reduction rules to the full version of this article. It is easy to see that they can be executed in polynomial time.

First, we present a kernelization for 1-edge Overlap Deletion, which, by Lemma 2, is equivalent to the problem of destroying *diamonds* by at most k edge deletions. We introduce four data reduction rules for this problem and show that an instance that is reduced with respect to all these rules has $O(k^4)$ vertices.

Rule 1. If there is a maximal clique C containing only edges which are not in any other maximal clique, then remove all edges of C.
Rule 2. If there is a matching of size greater than k in the complement graph of the graph that is induced by the common neighbors of the two endpoints of an edge e, then remove e, add e to the solution, and decrease the parameter k by one.
Rule 3. Remove all vertices that are not in any diamond.
Rule 4. If there is a critical clique with more than $k + 2$ vertices, then remove vertices until only $k + 2$ vertices remain.

Theorem 9. *1-Edge Overlap Deletion admits a problem kernel with $O(k^4)$ vertices which can be found in $O(m^3\sqrt{n} + m^2n^2)$ time.*

Proof. Let G denote an input graph reduced with respect to the four reduction rules. Partition the vertices of the graph G', resulting by applying a solution S to the input graph G, into two subsets, one set X containing the vertices that are endpoints of edges deleted by S and $Y := V \setminus X$. Further, construct for each edge $e \in S$ a set Y_e containing the vertices in Y that occur together with e in some diamonds. By Rule 3, $Y = \bigcup_{e \in S} Y_e$.

First, we show that for every maximal clique C in $G'[Y]$ it holds that $C \subseteq Y_e$ for an edge $e \in S$: In G, there is a maximal clique C' containing C and, by Rule 1, C' has an edge $e = \{u, v\}$ which is in two maximal cliques. If $e \in S$, then every vertex in C should be in Y_e; otherwise, there must be a vertex $w \notin C'$

that is adjacent to both u and v. One of the edges $\{u, w\}$ and $\{v, w\}$ must then be in S. Thus, every vertex in C must build a diamond with this deleted edge and C is contained in either $Y_{\{u,w\}}$ or $Y_{\{v,w\}}$.

Next, we prove that, for every edge $e = \{u, v\} \in S$, at most $4k$ maximal cliques of $G'[Y]$ are subsets of Y_e. Clearly, all vertices in Y_e must be adjacent to one of u and v. Let $N_{u,v}$ denote the common neighbors of u and v in Y_e. Obviously, $N_{u,v}$ is an independent set and, by Rule 2, $|N_{u,v}| \leq 2k$. Let $N_v := (N(v) \setminus N(u)) \cap Y_e$ and $N_u := (N(u) \setminus N(v)) \cap Y_e$. Since $N_{u,v}$ is an independent set, no vertex from $N_v \cup N_u$ can be adjacent to two vertices in $N_{u,v}$. Then, we can partition the vertices in $N_u \cup N_v$ into at most $4k$ subsets according to their adjacency to the vertices from $N_{u,v} = \{x_1, \ldots, x_l\}$ with $l \leq 2k$, every subset N_{u,x_i} (or $N(v, x_i)$) containing the vertices in $N(u) \cap N(x_i)$ (or $N(v) \cap N(x_i)$). It is easy to see that each of these subsets is a clique, since, otherwise, we would have some undestroyed diamond. With the same argument, there cannot be an edge between N_{u,x_i} and N_{u,x_j} with $i \neq j$. Moreover, the edges between N_{u,x_i} and N_{v,x_j}, if there are any, do not belong to the maximal cliques that are contained in Y_e. The reason is that the two endpoints of such an edge cannot have common neighbors in Y_e; otherwise, there would be some undestroyed diamond. Thus, we have at most $4k$ maximal cliques in $G'[Y]$ which are entirely contained in Y_e.

Finally, we show that if two vertices $u, v \in Y$ are contained in exactly the same sets of maximal cliques in Y, then they have the same neighborhood in G. Assume that this is not true. Then, u and v must have different neighborhoods in X. Let $w \in X$ be a neighbor of u but not of v. Since every two maximal cliques in Y can intersect in at most one vertex (due to the 1-edge overlap property), there can be only one maximal clique in Y containing both u and v. Assume that this maximal clique is contained in Y_e for an edge $e \in S$. Moreover, there must be another clique C in G containing w and u, but not v. By Rule 1, C must contain an edge which is part of two maximal cliques. This implies that the vertices w and u have to be in $Y_{e'}$ for an edge $e' \in S$ and $e \neq e'$. This means that there has to be a maximal clique in $Y_{e'}$ containing u but not v, contradicting that u and v are contained in the same sets of maximal cliques in Y.

Putting all the arguments together, we can now show an upper bound for the number of vertices in the reduced instance. Clearly, $|X| \leq 2k$. To bound $|Y|$, note that we have at most k Y_e's. Each of them contains at most $4k$ maximal cliques of $G'[Y]$. Since every maximal clique of $G'[Y]$ is contained in Y_e for one $e \in S$, we have altogether at most $4k^2$ maximal cliques in $G'[Y]$. It remains to show a size bound for each of these cliques. From the vertices in one clique K, only $4k^2$ of these can be in more than one maximal clique in Y, since every two such cliques overlap in at most one vertex. The remaining vertices of K then have identical neighborhoods. Thus, by Rule 4, K contains at most $4k^2 + k + 2$ vertices. This yields the required size bound on $|Y|$ and, therefore, on the reduced instance. □

Next, we provide a kernelization for 2-Vertex Overlap Deletion. In the following, we say that a vertex is *satisfied* if it is contained in at most two maximal cliques, and a clique is satisfied if all its vertices are satisfied. A clique

is a neighbor of an other clique if they share some vertex or edge. Here, the polynomial-time executable data reduction rules read as follows.

Rule 1. If there is a critical clique K with more than $k+1$ vertices, then remove vertices from K until only $k+1$ vertices remain.

Rule 2. If there exists a satisfied maximal clique K and K's neighbors are all satisfied, then remove all edges in K that are not in other maximal cliques.

Rule 3 Let G be a graph reduced with respect to Rule 1. Let K be a maximal clique of G. Consider ℓ maximal cliques $K_1, \ldots, K_\ell$ fulfilling the following two conditions:
1.) $K \cap K_i \neq \emptyset$, $1 \leq i \leq \ell$, and
2.) all vertices in K_i are satisfied, $1 \leq i \leq \ell$.
If $\sum_{i=1}^{\ell} |K_i \cap K| \geq 3k+4$, then delete all edges between $K_1 \cap K$ and $K \setminus K_1$.

Rule 4. Remove connected components that fulfill the 2-vertex overlap property.

Theorem 10. 2-Vertex Overlap Deletion *admits a problem kernel with* $O(k^3)$ *vertices.*

6 Conclusion

We have studied for the first time new cluster graph modification problems motivated by the practical relevance of clustering with overlaps [7,18]. Naturally, studying a so far unexplored set of problems, there remain many challenges for future work. We list only a few of them. First, it is conceivable that the forbidden subgraph characterizations we developed for cluster graphs with overlaps can be further refined. Second, it is desirable to improve the upper bounds on our fixed-parameter algorithms (including the kernelization results) and to further extend the list of fixed-parameter tractability results (in particular, achieving kernelization results for problems other than 1-Edge Overlap Deletion and 2-Vertex-Overlap Deletion). Third, corresponding experimental studies (like those undertaken for Cluster Editing, see [4,7]) are a natural next step. Fourth, the polynomial-time approximability of our problems remains unexplored. Fifth and finally, it seems promising to study overlaps in the context of the more general correlation clustering problems (see [1]) or by relaxing the demand for (maximal) cliques in cluster graphs by the demand for some reasonably dense subgraphs (as previously considered for Cluster Editing [14]).

References

1. N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Mach. Learn.*, 56(1–3):89–113, 2004.
2. A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *J. Comput. Biol.*, 6(3/4):281–292, 1999.
3. S. Böcker, S. Briesemeister, Q. B. A. Bui, and A. Truß. Going weighted: Parameterized algorithms for cluster editing. In *Proc. 2nd COCOA*, volume 5165 of *LNCS*, pages 1–12. Springer, 2008.

4. S. Böcker, S. Briesemeister, and G. W. Klau. Exact algorithms for cluster editing: Evaluation and experiments. In *Proc. 7th WEA*, volume 5038 of *LNCS*, pages 289–302. Springer, 2008.
5. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Inf. Process. Lett.*, 58(4):171–176, 1996.
6. P. Damaschke. Fixed-parameter enumerability of Cluster Editing and related problems. *Theory Comput. Syst.*, 2009. To appear.
7. F. Dehne, M. A. Langston, X. Luo, S. Pitre, P. Shaw, and Y. Zhang. The cluster editing problem: Implementations and experiments. In *Proc. 2nd IWPEC*, volume 4169 of *LNCS*, pages 13–24. Springer, 2006.
8. R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.
9. M. R. Fellows, M. A. Langston, F. A. Rosamond, and P. Shaw. Efficient parameterized preprocessing for Cluster Editing. In *Proc. 16th FCT*, volume 4639 of *LNCS*, pages 312–321. Springer, 2007.
10. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
11. 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.
12. D. L. Greenwell, R. L. Hemminger, and J. B. Klerlein. Forbidden subgraphs. In *Proc. 4th Southeastern Conf. on Comb., Graph Theory and Computing*, pages 389–394. Utilitas Mathematica, 1973.
13. J. Guo. A more effective linear kernelization for Cluster Editing. *Theor. Comput. Sci.*, 410(8-10):718–726, 2009.
14. J. Guo, C. Komusiewicz, R. Niedermeier, and J. Uhlmann. A more relaxed model for graph-based data clustering: s-plex editing. In *Proc. 5th AAIM*, LNCS. Springer, 2009. To appear.
15. M. Křivánek and J. Morávek. NP-hard problems in hierarchical-tree clustering. *Acta Inform.*, 23(3):311–323, 1986.
16. K. Makino and T. Uno. New algorithms for enumerating all maximal cliques. In *Proc. 9th SWAT*, volume 3111 of *LNCS*, pages 260–272. Springer, 2004.
17. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
18. G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
19. R. Peeters. The maximum edge biclique problem is NP-complete. *Discrete Appl. Math.*, 131(3):651–654, 2003.
20. F. Protti, M. D. da Silva, and J. L. Szwarcfiter. Applying modular decomposition to parameterized cluster editing problems. *Theory Comput. Syst.*, 44(1):91–104, 2009.
21. R. Shamir, R. Sharan, and D. Tsur. Cluster graph modification problems. *Discrete Appl. Math.*, 144(1–2):173–182, 2004.
22. R. Sharan, A. Maron-Katz, and R. Shamir. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. *Bioinformatics*, 19(14):1787–1799, 2003.
23. M. Talmaciu and E. Nechita. Recognition algorithm for diamond-free graphs. *Informatica*, 18(3):457–462, 2007.
24. Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: theory and its application to image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1101–1113, 1993.
25. R. Xu and D. Wunsch II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.