Efficient Parameterized Preprocessing for Cluster Editing

Michael Fellows[1,2⋆], Michael Langston[3⋆⋆], Frances Rosamond[1⋆⋆⋆] and Peter Shaw[1]

[1] University of Newcastle, Callaghan NSW 2308, Australia
{michael.fellows,frances.rosamond,peter.shaw}@newcastle.edu.au
[2] Durham University, Institute of Advanced Study,
Durham DH1 3RL, United Kingdom
[3] University of Tennessee, Knoxville, Tennessee 37996-3450, U.S.A.
langston@cs.utk.edu

Abstract. In the CLUSTER EDITING problem, a graph is to be changed to a disjoint union of cliques by at most k operations of *edge insertion* or *edge deletion*. Improving on the best previously known quadratic-size polynomial-time kernelization, we describe how a crown-type structural reduction rule can be used to obtain a $6k$ kernelization bound.

1 Introduction

The CLUSTER EDITING problem takes as input an undirected graph G, and asks whether k *edge changes* are sufficient to transform G into a graph G' that is a disjoint union of complete subgraphs. Such a graph G' is called a *cluster graph*. The problem was first introduced by Bansal, Blum and Chawla [2] (where it is called CORRELATION CLUSTERING) in the context of machine learning, and by Shamir, Sharan and Tsur [25] in the context of bioinformatics applications such as the analysis of gene expression data. Chen, Jiang and Lin [9] and Damaschke [12] have described applications in phylogenetics. An implementation with target applications in gene regulatory network analysis has been described in [15].

In the latter application, the vertices represent genes, and edges join coregulated genes belonging to functional groups represented by the complete subgraphs. The observed graph G might not be a cluster graph, due to experimental

⋆ Research supported by the Australian Research Council through the Australian Centre in Bioinformatics, by the University of Newcastle Parameterized Complexity Research Unit under the auspices of the Deputy Vice-Chancellor for Research, by a Fellowship to the Durham University Institute for Advanced Studies and by a William Best Fellowship at Grey College, Durham, while the paper was in preparation.

⋆⋆ Research supported in part by the U.S. National Institutes of Health under grants 1-P01-DA-015027-01, 5-U01-AA-013512 and 1-R01-MH-074460-01, by the U.S. Department of Energy under the EPSCoR Laboratory Partnership Program, by the European Commission under the Sixth Framework Programme, and by the Australian Research Council under grants to Griffith University and the University of Newcastle.

⋆⋆⋆ Research supported by the Australian Centre in Bioinformatics.

errors, noisy data and other reasons. A reasonable approach to formulating a parsimonious hypothesis concerning a hidden clustering is to determine a minimum number of edge changes that can transform the observed graph into a cluster graph.

1.1 Previous Work

Cluster Editing is NP-hard [20] and does not admit a PTAS unless $P = NP$ [8]. A polynomial-time 4-approximation algorithm for Cluster Editing is described in [8]. The problem is easily seen to be in FPT by a search tree algorithm that runs in time $O^*(3^k)$, based on the observation that the problem is equivalent to destroying (by means of the allowed operations) all occurences of an induced P_3 (a vertex-induced path consisting of three vertices). (It can also be classified as FPT using general results of Cai [5].) This was improved by a more sophisticated search tree strategy by Gramm, Guo, Hüffner and Niedermeier [18] to $O^*(2.27^k)$, and then further improved to $O^*(1.92^k)$ based on automated search tree generation and analysis [17]. In realistic applications, the enumeration of all possible solutions, for a given G and k, may be important, and Damaschke has described practical FPT algorithms for this [10]. Damaschke has also shown that a number of nontrivial and applications-relevant generalizations of Cluster Editing are fixed-parameter tractable [11]. (These generalizations study situations where the clusters may have limited overlap, rather than be completely disjoint, a matter of importance in many data-clustering applications.) For general background on parameterized complexity, see [13, 16, 21].

The best known FPT kernelization for Cluster Editing, to a graph on $O(k^2)$ vertices having $O(k^3)$ edges, is shown in [18] (exposited in [21]) and has been further improved by a constant factor by Damaschke [10]. Subsequent to, and extending the work reported here, a polynomial-time kernelization to a graph on at most $4k$ vertices has been announced [19].

1.2 Our Results

We describe a many:1 polynomial time kernelization to a problem kernel graph on $O(k)$ vertices, based on a crown-type reduction rule. Crown-type reduction rules have proved to be a surprisingly powerful method in FPT kernelization, applicable to a wide variety of problems [7, 14, 24, 23]. In particular, we obtain a kernelization to a graph on at most $6k$ vertices. Our result is roughly analogous to the $2k$ kernelization for the Vertex Cover problem due to Nemhauser and Trotter [22]. In the case of Vertex Cover, linear kernelization can be achieved more than one way. In particular, a $2k$ kernelization for the Vertex Cover problem can be achieved by a crown reduction rule (see [21] for an exposition), that resembles the reduction rule for Cluster Editing that we employ here. The main idea of a crown-type reduction rule is to identify cutsets that separate off a subgraph with homogeneous structure, allowing the input to be simplified.

Reduction rules often cascade and can have great power in practical settings. Although parameterization allows the efficiency of reduction rules to be measured, it is not necessary for the parameter to be small in order for this output of the study of parameterized algorithmics and complexity to be useful, because the reduction rules can be applied in polynomial time, and hence are of use even when the parameter is not guaranteed to be small.

2 A Crown Reduction Rule

Suppose that (G, k) is a yes-instance of the CLUSTER EDITING problem. Figure 1 shows a depiction of a solution, where the cliques that result from the editing are represented by the boxes $\mathcal{C}_i$, $i = 1, ..., m$. The depiction shows only the *edits* of the solution. Define a vertex to be *of type A* if it is involved in an edge addition. Define a vertex to be *of type B* if it is not of type A, and is involved in an edge deletion. Define a vertex to be *of type C* if it is not of type A and not of type B.

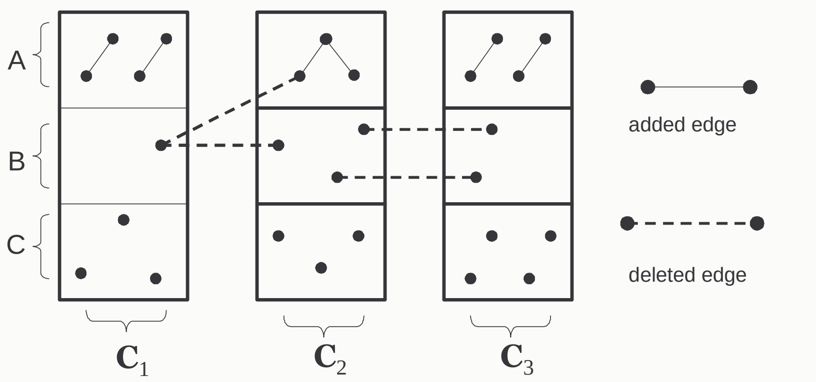

Fig. 1. A depiction of a solution showing (only) the edits.

The following lemma is trivial.

Lemma 1. *There are at most $2k$ vertices of type A or B in a solution $\mathcal{S}$.*

It follows from the lemma that if G is "large", then almost all of the vertices of the graph are of type C. Consider two vertices u and v of type C that belong to the clique $\mathcal{C}_i$. Then u and v are adjacent, and adjacent to every vertex of type A or type B in $\mathcal{C}_i$, and are not adjacent to any vertex of a clique $\mathcal{C}_j$ for $j \neq i$. In other words, we know everything there is to know about u and v, merely because they are of type C. We might expect that if the number of vertices of type C, for a given clique $\mathcal{C}_i$ is sufficiently large, then some reduction rule might apply. We identify such a reduction rule, based on the following notion of a structural decomposition applicable to this problem.

Definition 1. *A cluster crown decomposition of a graph $G = (V, E)$ is a partition of the vertices of V into four sets (C, H, N, X) satisfying the following conditions:*

1. *C is a clique.*
2. *Every vertex of C is adjacent to every vertex of H.*
3. *H is a cutset, in the sense that there are no edges between C and $N \cup X$.*
4. *$N = \{v \in V - C - H : \exists u \in H, uv \in E\}$.*

Our kernelization algorithm is based on the following reduction rule.

The Cluster Crown Reduction Rule. If (G, k) is an instance of the Cluster Editing problem, and G admits a cluster crown decomposition (C, H, N, X) where

$$|C| \geq |H| + |N| - 1$$

then replace (G, k) with (G', k') where $G' = G - C - H$ and $k' = k - e - f$, where e is the number of edges that need to be added between vertices of H in order to make $C \cup H$ into a clique, and f is the number of edges between H and N.

3 A Linear Kernelization Bound

We defer the discussion of soundness for the Cluster Crown Reduction Rule, as well as a proof that it can be exhaustively applied in polynomial time, to §4 and §5.

We next argue that this rule gives us a linear kernelization for the problem.

Theorem 1. *Suppose that (G, k) is a yes-instance that does not admit a cluster crown decomposition to which the Cluster Crown Reduction Rule applies. Then G has less than $6k$ vertices.*

Proof. In order to discuss the situation, we introduce some notation. Suppose $\mathcal{S}$ is a solution for the instance (G, k). Let $\mathcal{C}_i$ denote the cliques formed by $\mathcal{S}$, $i = 1, ..., m$. Corresponding to each $\mathcal{C}_i$ is a cluster crown decomposition (C_i, H_i, N_i, X_i) of G. Let $c_i = |C_i|$, $h_i = |H_i|$ and $n_i = |N_i|$.

By the assumption that the Cluster Crown Reduction Rule does not apply, we have that for $\forall i$:

$$c_i \leq h_i + n_i - 2$$

It follows that

$$\sum_{i=1}^{m} c_i \leq \sum_{i=1}^{m} h_i + \sum_{i=1}^{m} n_i - 2m$$

Lemma 1 shows that $\sum_{i=1}^{m} h_i \leq 2k$, and we also have the bound $\sum_{i=1}^{m} n_i \leq 2k$ because the solution $\mathcal{S}$ involves deleting at most k edges between the sets H_i. These edges are the only source of neighbors in the sets N_i, and each such edge is counted twice in the sum $\sum_{i=1}^{m} n_i$. Therefore $\sum_{i=1}^{m} c_i \leq 4k - 4$, and G therefore has at most $6k - 4$ vertices, noting that $|V| = \sum_i (c_i + h_i)$. □

Figure 2 shows that the bound of Theorem 1 cannot be improved: shown is a construction (for $k = 3$, this easily generalizes) of an irreducible yes-instance having $6k - 4$ vertices.

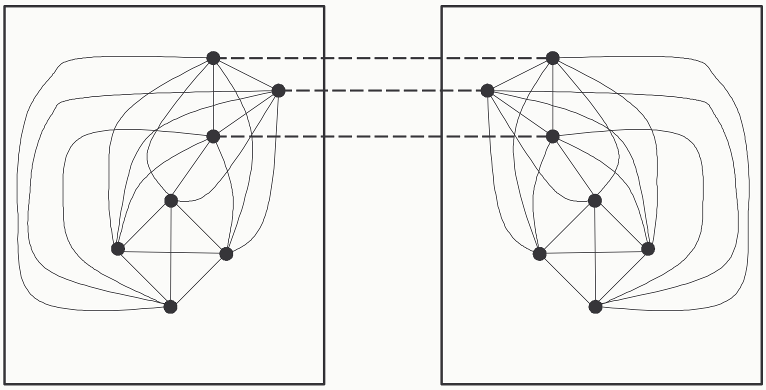

Fig. 2. Irreducible "yes" instance for $k = 3$ with $6k - 4$ vertices.

4 Soundness of the Reduction Rule

In this section we prove the soundness of the Cluster Crown Reduction Rule.

Lemma 2. *The Cluster Crown Reduction Rule is sound. That is, suppose we have a cluster crown decomposition* (C, H, N, X) *where* $|C| \geq |H| + |N| - 1$ *for an instance* (G, k)*, and that* (G', k') *is the reduced instance. Then* (G, k) *is a YES-instance for* Cluster Editing *if and only if* (G', k') *is a YES-instance.*

Proof. Any solution to the problem may be viewed as a partition of the vertex set of G, with associated costs: (1) the number of edges that must be *added* within classes of the partition, (2) the number of edges that must be *deleted* between classes of the partition, where the total cost is minimized. If π is a partition of V, then we will use $\Gamma(\pi)$ to denote the total editing cost of making the vertex classes of π into the disjoint cliques of a solution.

Consider a partition π of the vertex set V that minimizes the total editing cost, where π partitions V into m classes V_i, $i = 1, ..., m$. In this situation, we use C_i to denote $C \cap V_i$, H_i to denote $H \cap V_i$, N_i to denote $N \cap V_i$ and X_i to denote $X \cap V_i$. Let $c_i = |C_i|$, $h_i = |H_i|$, $n_i = |N_i|$ and $x_i = |X_i|$. If U and W are two disjoint sets of vertices, we write $e(U, W)$ to denote the number of adjacent pairs of vertices, $u \in U$ and $w \in W$, and we will write $\overline{e}(U, W)$ to denote the number of nonadjacent pairs of vertices, $u \in U$ and $w \in W$.

We argue that the partition π' that subtracts all vertices of $C \cup H$ from the classes of π and makes a new class consisting of $C \cup H$, has total editing cost no worse than that of π. That is, we will argue that $\Gamma(\pi) - \Gamma(\pi') \geq 0$. It is sufficient to show that

$$\sum_{i<j} c_i c_j + \sum_{i<j}(c_i h_j + c_j h_i) + \sum_i c_i n_i + \sum_i h_i x_i \tag{1}$$

$$+ \sum_{i<j} e(H_i, H_j) - \sum_{i<j} \overline{e}(H_i, H_j) \tag{2}$$

$$+ \sum_i \overline{e}(H_i, N_i) - \sum_i e(H_i, N_i) \tag{3}$$

$$\geq 0 \tag{4}$$

The terms in line (2) are greater than or equal to $-\sum_{i<j} h_i h_j$, and the terms in line (3) are greater than or equal to $-\sum_i h_i n_i$. Therefore it is enough to show that the following inequality holds.

$$\sum_{i<j} c_i c_j + \sum_{i<j}(c_i h_j + c_j h_i) + \sum_i c_i n_i - \sum_{i<j} h_i h_j - \sum_i h_i n_i \quad \geq \quad 0 \tag{5}$$

Let $c = |C| = \sum_i c_i$, $h = |H| = \sum_i h_i$ and $n = |N| = \sum_i n_i$. We can think of the inequality (5) as simply a statement to be proved about 3 by m matrices of non-negative integers, where the first row is $c_1 ... c_m$, the second row is $h_1 ... h_m$ and the third row is $n_1 ... n_m$. The statement is to hold for any such matrix, so long as $c \geq h + n - 1$.

Observation. An inspection of (5) shows that if there is any counterexample, then there is a counterexample where $c = h + n - 1$. Call such a matrix M *balanced* if for all i, $c_i \geq h_i$. It is straightforward to verify that (5) holds for all balanced matrices; the first positive sum dominates the first negative sum, and the third positive sum dominates the second negative sum.

The inequality (5) also holds if $m = 1$, since the hypothesis that $c \geq h + n - 1$ implies that $cn \geq hn + n^2 - n$ so that $cn \geq hn$ (which is what we must show in this simple case). We have now established the necessary base cases for an induction. Suppose the matrix M is not balanced and that M has l columns. Then M has a column j where $c_j < h_j$. The truth of the inequality (5) is unaffected by permutations of the columns, so we can assume that $j = l$. Write $\alpha(M)$ to denote the value of the lefthand side of (5) for a matrix M. Let M' be M with column l deleted. Certainly M' satisfies $c' \geq h' + n' - 1$, where c', h' and n' denote the row sums for M'. However, while we assume that $c = h + n - 1$

for M, the analogous equality does not hold for M', but this is not a problem, because, by the observation above, and our induction on l, $\alpha(M') \geq 0$ for *any* M' with $l-1$ columns. It suffices to argue that $\Delta(M) = \alpha(M) - \alpha(M') \geq 0$. Elaborating, we must show that:

$$\Delta(M) \;=\; \sum_{i=1}^{l-1} c_i c_l + \sum_{i=1}^{l-1}(h_i c_l + c_i h_l) - \sum_{i=1}^{l-1} h_i h_l + c_l n_l - h_l n_l \;\geq\; 0$$

Factoring some of these terms, what we must show is:

$$c_l \left(\sum_i c_i\right) + c_l \left(\sum_i h_i\right) + h_l \left(\sum_i c_i\right) - h_l \left(\sum_i h_i\right) + c_l n_l - h_l n_l \;\geq\; 0$$

The last inequality holds if

$$c_l h' + h_l c' - h_l h' + c_l n_l - h_l n_l \;\geq\; 0$$

and this holds if

$$c_l h' + h_l c - c_l h_l - h_l h' + c_l n_l - h_l n_l \;\geq\; 0$$

Replacing c with $h + n - 1$, it is enough to show

$$c_l h' + h_l \left(\sum_{i=1}^{l}(h_i + n_i) - 1\right) - c_l h_l - h_l h' + c_l n_l - h_l n_l \;\geq\; 0$$

which can be rewritten as

$$c_l h' + (h_l h' + h_l n' + h_l^2 + h_l n_l - h_l) - c_l h_l - h_l h' + c_l n_l - h_l n_l \;\geq\; 0$$

Cancelling and gathering terms, our task is to show

$$c_l h' + h_l n' + h_l(h_l - c_l - 1) + c_l n_l \;\geq\; 0$$

which is true, because $c_l < h_l$. □

5 Efficiently Applying the Reduction Rule

In this section we describe how to compute in polynomial time whether the input graph G admits a cluster crown decomposition to which the Cluster Crown Reduction Rule can be applied.

Definition 2. *Distinct vertices u, v of a graph G are termed twins if:*
(1) u and v are adjacent, and
(2) $N[u] = N[v]$.
For vertices u and v that are twins, we will denote this by $u \sim v$.

Observe that if (C, H, N, X) is a cluster crown decomposition for a graph G, then every pair of vertices in C are twins. The next lemma shows that in some sense we can restrict our attention to cluster crown decompositions that have a kind of "maximality" property.

Lemma 3. *If G admits any cluster crown decomposition (C, H, N, X) satisfying:*
(1) $|C| \geq |H| + |N| - 1$
then G admits a cluster crown decomposition (C', H', N', X') that satisfies (1) and also the further condition:
(2) $\forall u \in C'\ \forall v \in H :\ \neg(u \sim v)$.

Proof. Suppose (C, H, N, X) is a cluster crown decomposition satisfying (1), and that there are vertices $u \in C$ and $v \in H$ with $u \sim v$. This implies that v has no neighbors in $N \cup X$. If we take $C' = C \cup \{v\}$ and $H' = H - \{v\}$ then we also have a cluster crown decomposition satisfying (1). □

A cluster crown decomposition that satisfies the two conditions of the lemma above is termed *maximal*.

Definition 3. *The twin graph $\tau(G)$ of a graph $G = (V, E)$ has the same vertex set V, and two vertices u, v are adjacent in $\tau(G)$ if and only if $u \sim v$.*

We say that a subgraph H of a graph G is an *isolated clique* if H is a complete subgraph, and the vertices of H are adjacent in G only to vertices of H. (In other words, G consists of a disjoint union of H and the subgraph $G - H$.)

Lemma 4. *(1) If (C, H, N, X) is a maximal cluster crown decomposition of a graph $G = (V, E)$, then the vertices of C form an isolated clique in $\tau(G)$.*
(2) Conversely, an isolated clique in $\tau(G)$ corresponds to a maximal cluster crown decomposition where the vertices of the set C are the vertices of the isolated clique.

Proof. (1) follows from the observation that every pair of vertices in C are twins, and the definition of maximality. To see that (2) holds, let C denote the set of vertices of an isolated clique in $\tau(G)$. Take

$$H = (\cup_{v \in C} N(v)) - C$$
$$N = N(H) - C$$
$$X = V - C - H - N$$

It is easy to check that the conditions for a cluster crown decomposition are satisfied. □

Our algorithm is described as follows.

Kernelization Algorithm.
On input (G, k):

Step 1. Compute the twin graph $\tau(G)$.
Step 2. Identify the isolated cliques in $\tau(G)$, and the corresponding maximal cluster crown decompositions. Apply the Reduction Rule if an opportunity is found.

The algorithm can clearly be implemented in polynomial time.

6 Discussion and Open Problems

We have shown that the Cluster Editing problem admits a polynomial time many:1 kernelization to a graph on at most $6k$ vertices. Based in part on the ideas introduced here, this kernelization bound has recently been improved to $4k$ [19].

Linear kernelization for the Cluster Editing problem could be viewed as an analog result, for a problem somewhat related to the Vertex Cover problem, of the first linear kernelization result for that problem due to Nemhauser and Trotter [22]. Later (recently) a different combinatorial route to linear kernelization for Vertex Cover was discovered [7, 1], based on so-called "crown-type" reduction rules. This has proved useful in practical applications [3, 4]. Both approaches yield a $2k$ kernelization for Vertex Cover. The importance of kernelization is that pre-processing is a nearly universal practical strategy for coping with hard problems. Since kernelization rules can be applied in polynomial time, the overall situation is that parameterization allows us to measure their *efficiency*, but their *relevance* and practical significance is not tied to situations where the parameter is small: the main outcome, as seen from the practical side, is just a "smart" preprocessing subroutine that can be deployed in *any* algorithmic approach to the NP-hard Cluster Editing problem, including heuristic approaches.

The Vertex Cover problem, the Cluster Editing problem, and (for example) the generalizations studied by Damaschke [11], are all, roughly speaking, concerned with editing a graph to one or more (or a specified number) of clusters (possibly with limited overlap). We may roughly conceptualize here a class of "graph editing" problems, parameterized by the number of edit operations. Suitably formalized: are all such problems fixed-parameter tractable? Do they all admit linear kernels? Are general results possible concerning this area of investigation?

References

1. F. Abu-Khzam, R. Collins, M. Fellows, M. Langston, W. H. Suters and C. Symons. Kernelization algorithms for the vertex cover problem: theory and experiments. *Proceedings of ALENEX'04*, ACM-SIAM Publications (2004), 62–69.
2. N. Bansal, A. Blum and S. Chawla. Correlation clustering. *Machine Learning* 56 (2004), 89–113. Preliminary version in: *Proceedings 43rd IEEE FOCS 2002*, 238–247.
3. N. E. Baldwin, E. J. Chesler, S. Kirov, M. A. Langston, J. R. Snoddy, R. W. Williams and B. Zhang. Computational, integrative and comparative methods for

the elucidation of gene regulatory networks. *J. Biomedicine and Biotechnology* 2 (2005), 172–180.

4. N. E. Baldwin, R. L. Collins, M. A. Langston, M. R. Leuze, C. T. Symons and B. H. Voy. High performance computational tools for motif discovery. *Proceedings of the IEEE International Workshop on High Performance Computational Biology* (HiCOMB), 2004.
5. L. Cai. Fixed-parameter tractability of graph modification problems for hereditary properties. *Information Processing Letters* 58 (1996), 171–176.
6. M. Chlebík and J. Chlebíková. Improvement of Nemhauser-Trotter theorem and its applications in parameterized complexity. *Proceedings 9th SWAT*, Springer-Verlag, *Lecture Notes in Computer Science* 3111 (2004), 174–186.
7. B. Chor, M. Fellows and D. Juedes. Linear kernels in linear time. *Proceedings 30th WG*, Springer-Verlag, *Lecture Notes in Computer Science* 3353 (2004), 36–53.
8. M. Charikar, V. Guruswami and A. Wirth. Clustering with qualitative information. *Journal of Computer and System Sciences* 71 (2005) 360–383.
9. Z. Z. Chen, T. Jiang, and G. Lin. Computing phylogenetic roots with bounded degrees and errors. *SIAM J. Computing* 32 (2003), 864–879.
10. P. Damaschke. On the fixed-parameter enumerability of cluster editing. In: *Proceedings of the International Workshop on Graph-Theoretic Concepts in Computer Science* (WG'05), Springer-Verlag, *Lecture Notes in Computer Science* (2005), 283–294.
11. P. Damaschke. Fixed-parameter tractable generalizations of cluster editing. *Proceedings of CIAC'06*, Springer-Verlag, *Lecture Notes in Computer Science* 3998 (2006), 321–332.
12. P. Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science* 351 (2006), 337–350.
13. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
14. F. Dehne, M. Fellows, F. Rosamond and P. Shaw. Greedy localization, iterative compression and modeled crown reductions: new FPT techniques, an improved algorithm for set splitting and a novel $2k$ kernelization for vertex cover. *Proceedings of the First International Workshop on Parameterized and Exact Computation*, Springer-Verlag, *Lecture Notes in Computer Science* vol. 3162 (2004), 271–280.
15. F. Dehne, M. A. Langston, X. Luo, S. Pitre, P. Shaw and Y. Zhang. The Cluster Editing problem: implementations and experiments. *Proceedings of IWPEC 2006*, Springer-Verlag, *Lecture Notes in Computer Science* 4169 (2006), 13–24.
16. J. Flum and M. Grohe, *Parameterized Complexity Theory*, Springer-Verlag, 2006.
17. J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Automated generation of search tree algorithms for hard graph modification problems. *Algorithmica* 39 (2004), 321–347.
18. J. Gramm, J. Guo, F. Hüffner, and R. Niedermeier. Graph-modeled data clustering: exact algorithms for clique generation. *Theory of Computing Systems* 38 (2005), 373–392, preliminary version in: *Proceedings of the 5th Italian Conference on Algorithms and Complexity* (CIAC '03), Springer-Verlag, *Lecture Notes in Computer Science* 2653 (2003), 108–119.
19. J. Guo. Manuscript, 2007.
20. M. Krivanek and J. Moravek. NP-hard problems in hierarchical tree clustering. *Acta Informatica* 23 (1986), 311–323.
21. R. Niedermeier. *Invitation to Fixed Parameter Algorithms.* Oxford University Press, 2006.

22. G. Nemhauser and L. Trotter. Vertex packings: structural properties and algorithms. *Mathematical Programming* 8 (1975), 232–248.
23. E. Prieto-Rodriguez. *Systematic kernelization in FPT algorithm design.* Ph.D. Thesis, School of EE&CS, University of Newcastle, Australia, 2005.
24. E. Prieto and C. Sloper. Looking at the stars. *Proceedings of IWPEC 2004*, Springer-Verlag, *Lecture Notes in Computer Science* 3162 (2004), 138–148.
25. R. Shamir, R. Sharan and D. Tsur. Cluster graph modification problems. *Discrete Applied Mathematics* 144 (2004), 173–182. Preliminary version in: *28th WG 2002, LNCS* 2573 (2002), 379–390.