New Races in Parameterized Algorithmics

Christian Komusiewicz and Rolf Niedermeier

Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Germany
{christian.komusiewicz,rolf.niedermeier}@tu-berlin.de

Abstract. Once having classified an NP-hard problem fixed-parameter tractable with respect to a certain parameter, the race for the most efficient fixed-parameter algorithm starts. Herein, the attention usually focuses on improving the running time factor exponential in the considered parameter, and, in case of kernelization algorithms, to improve the bound on the kernel size. Both from a practical as well as a theoretical point of view, however, there are further aspects of efficiency that deserve attention. We discuss several of these aspects and particularly focus on the search for "stronger parameterizations" in developing fixed-parameter algorithms.

1 Introduction

Efficiency is the central concern of algorithmics. In parameterized algorithmics, one tries to solve NP-hard problems in $f(k) \cdot n^{O(1)}$ time, where k is a problem-specific parameter (such as solution size), n is the size of the overall input, and f is an arbitrary function only depending (exponentially) on the parameter k [16, 23, 38]. Thus, the first step in parameterized algorithm design is to show that the considered problem is fixed-parameter tractable with respect to the chosen parameter k. Fixed-parameter algorithms are fast in case k is small and f grows "moderately". Consequently, once a problem is classified as fixed-parameter tractable, the race for the smallest function $f(k)$ starts.[1] There are many success stories in this direction, including problems such as Vertex Cover [13] and Undirected Feedback Vertex Set [11, 14]. Accompanied by these are similar races for the problem kernel size of the considered problem, for instance see the problems Cluster Editing [12] and again Undirected Feedback Vertex Set [45]. Furthermore, there is an ongoing deep theoretical effort for proving lower bounds (under complexity-theoretic assumptions) both for the $f(k)$ in the running time [24, 34] and the kernel size (polynomial vs non-polynomial) [8, 26].

In the two main lines of efficiency research in parameterized algorithmics described above, however, polynomial factors in the running time mostly are ignored. This is somewhat contrary to the fact that some key fixed-parameter tractability results have been termed "linear time for constant parameter value", basically meaning that the underlying problem can be solved in time $f(k) \cdot n$. Two examples in this direction are the "linear-time algorithms" for Treewidth [5]

[1] See the web site on FPT races: http://fpt.wikidot.com/fpt-races

and Crossing Number [31]. As to striving for linear-time algorithms for effective data reduction, only recently the concept of "linear-time kernelization" gained more attention [3, 4, 29]. Altogether, these are important subjects of study besides the established races described in the beginning.

The focus of this article, however, is on another, practically and theoretically fruitful aspect in the race for efficiency. As an example, consider the following NP-hard problem with applications in graph drawing.

> 2-Layer Planarization
> Input: An undirected graph $G = (V, E)$ and a $k \geq 0$.
> Question: Is there an $E' \subseteq E$ with $|E'| \leq k$ such that deleting the edges in E' from G results in a biplanar graph.

Herein, a graph is biplanar if the vertices can be arranged on two parallel lines such that the edges (drawn as straight lines) do not cross. The best known fixed-parameter algorithm with respect to solution size k is due to Suderman [44] and runs in $O(3.6^k + |G|)$ time, improving on previous algorithms [18, 22] with exponential factors 6^k and 5.2^k, respectively. Later Uhlmann and Weller [47] followed a different improvement approach by considering the parameter "feedback edge set number" f. The essential point here is that for the parameter feedback edge set number f it holds that $f \leq k$ and f is expected to be significantly smaller than k in many realistic settings. In other words, f is a "stronger parameter" than k and thus the fixed-parameter tractability result with respect to the parameter f can be considered as an improvement over the fixed-parameter tractability result with respect to the solution size parameter k. Uhlmann and Weller [47] showed that 2-Layer Planarization can be solved in $O(6^f \cdot f^2 + f \cdot |E|)$ time.[2] If $1.7 \cdot f \leq k$, which can be the case in real-world instances, this algorithm is faster than the previous ones [18, 22].

In the spirit of the above considerations, we discuss in the following new efficiency races in parameterized algorithmics mainly based on the concept of stronger (and, correspondingly, weaker) parameterizations.

Let k_1 and k_2 be two natural numbers denoting parameters for input instances (here, graphs) of an NP-hard problem under study. Note that often, these parameters are functions of the input graph G (such as the maximum degree of G). We say that k_1 is *stronger than* k_2 if there is a constant c such that $k_1 \leq c \cdot k_2$ for all input instances of the underlying problem and furthermore there is *no constant* c' such that $k_2 \leq c' \cdot k_1$ in all input instances. Correspondingly, k_2 is *weaker than* k_1 in this case. There are several other reasonable possibilities to define the notions of weaker and stronger parameters. Here, we choose a "linear upper bound" for the following two reasons:

- For polynomial-size problem kernels, the main measure of effectiveness is the degree of the polynomial function in the size bound.
- In the theoretical analysis of running times of fixed-parameter algorithms, the most important feature is usually considered to be the function class of the exponential factor, for example $2^{O(k)}$ vs. $2^{O(k^2)}$.

[2] Weller [48] recently reported on an improvement to $O(3.8^f \cdot f^2 + f \cdot |E|)$ time.

Due to the linear bounds, these results can be transferred from stronger to weaker parameters. Note that having neither the stronger nor the weaker relation between two parameters does not necessarily imply that they are *incomparable.*

In the remainder of this article we describe some findings and challenges in the context of considering stronger and weaker parameters with respect to questions of efficiency. Herein, we deal both with structural parameters as well as parameterizations related to solution size.[3]

Preliminaries. We use n to denote the input size. A problem is called *fixed-parameter tractable* (FPT) if it can be solved in $f(k) \cdot \text{poly}(n)$ time, where f is a computable function only depending on k. The basic class of *parameterized intractability* is called W[1]. Problems that can be solved in polynomial time for constant parameter values are contained in the class XP. Note that these problems are not necessarily fixed-parameter tractable since the degree of the polynomial can be a function of the parameter k. 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 in polynomial time into a new instance x' with parameter $k' \leq k$ such that the size of x' is upper-bounded by some function g only depending on k and the instance (x, k) is a yes-instance if and only if (x', k') is a yes-instance. The reduced instance is called a *problem kernel*; in case the function g is a polynomial function it is called a *polynomial-size problem kernel. Turing kernelization* is a similar approach, where the main difference is that not only one, but polynomially many kernels can be created.

2 Structural Parameterizations or Navigating Through Parameter Space

In this section, we give three examples for algorithmic studies of NP-hard graph problems that have been conducted in the spirit of identifying stronger or weaker parameters yielding tractability results. Most of the parameters considered in these studies are shown in Figure 1. The idea behind these parameters is that they measure the distance to easy, that is, polynomial-time solvable, input instances. Here, these are certain classes of graphs, for instance forests or bipartite graphs.

Small-Diameter Subgraphs. The following NP-hard problem is motivated by social and biological network analysis [1].

> 2-Club
> Input: An undirected graph G and an integer ℓ.
> Question: Is there a subgraph G' with at least ℓ vertices that has diameter at most two?

[3] While all our case studies are based on graph problems, the fundamental ideas and concepts presented here are clearly not restricted to these.

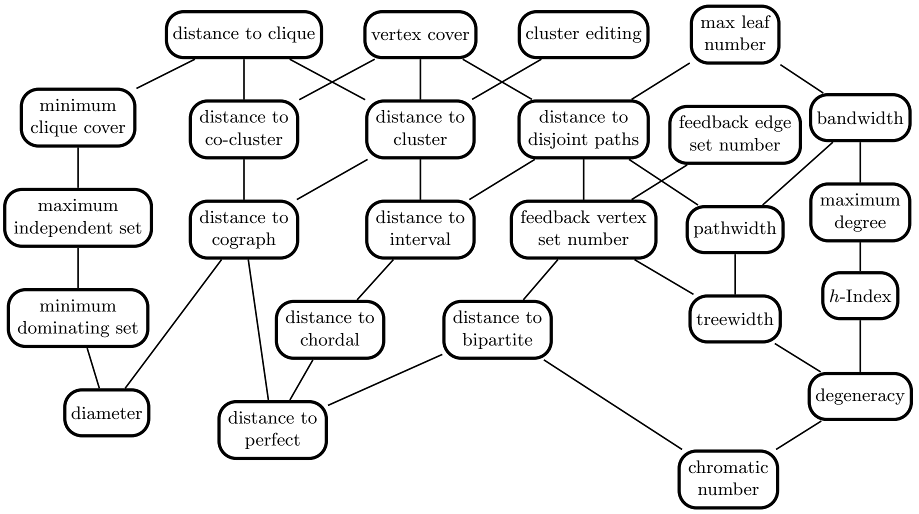

Fig. 1. An overview of the relation between some structural parameterizations for undirected graphs. Herein, "distance to X" is the number of vertices that have to be deleted in order to transform the input graph into a graph from the graph class X. For two parameters that are connected by a line, the upper parameter is weaker than the parameter below. For example, vertex cover is weaker than "distance to vertex-disjoint paths" since deleting a vertex cover produces an independent set, which also belongs to the graph class of disjoint paths. Similarly, distance to disjoint paths is weaker than "feedback vertex set number" which is weaker than "distance to bipartite".

By an easy polynomial-time data reduction rule it follows that 2-Club admits a Turing kernel with $O(\Delta^2)$ vertices where Δ is the *maximum degree* of G [42].[4] This implies fixed-parameter tractability for the parameter Δ. For applications in social network analysis, this parameterization seems not very useful, since social networks typically contain so-called *hubs*, that is, vertices of high degree. The number of hubs, however, is usually relatively small. Consequently, it is interesting to consider parameterizations expressing that many vertices in the graph have low degree. An established parameter in this context is the *degeneracy* of a graph: A graph G has degeneracy d if for each induced subgraph of G there is at least one vertex that has degree at most d. The concept of degeneracy was introduced as "coloring number" by Erdős and Hajnal [20].

While degeneracy is an interesting parameter, it is sometimes too strong in order to obtain fixed-parameter tractability results. An interesting alternative is therefore the *h-index* of a graph G. This is the maximum number h such that G contains h vertices of degree at least h [19]. Roughly speaking, the main difference between h-index and degeneracy is that a low h-index captures the global property that there are few high-degree vertices, whereas degeneracy

[4] These results were presented for the parameter solution size ℓ. They also hold for the stronger parameter maximum degree Δ.

can be one even if there are many high-degree vertices in the input graph. For example, a 2012 version of the DBLP co-author graph,[5] a typical social network, has $\approx 750,000$ vertices, maximum degree 804, h-index 208, and degeneracy 113. The 2-Club problem turns out to be solvable in $n^{f(h)}$ time but W[1]-hard with respect to the h-index of the input graph which also implies W[1]-hardness for the degeneracy of the input graph [30]. These two results now motivate the following two questions:

- Is 2-Club solvable in $n^{f(k)}$ time when k is the degeneracy of the input graph?
- Is there a parameter k between maximum degree and h-index for which 2-Club has the same kernelization properties as for the parameter maximum degree, that is, it admits an $O(k^2)$-vertex Turing kernel?

Preprocessing for Treewidth. Computing the treewidth of a graph is a fundamental problem in graph algorithms:

> Treewidth
> Input: An undirected graph G, and an integer t.
> Question: Does G have treewidth at most t?

Treewidth is fixed-parameter tractable with respect to the treewidth itself [5] but is unlikely to admit a polynomial-size problem kernel for this parameter [8, 17]. However, several data reductions for the Treewidth problem are known to be effective in practice [6, 7]. In order to explain this effectiveness, the power of these data reduction rules was analyzed with respect to structural parameters that are weaker than treewidth [9]. On the one hand, it was shown that Treewidth admits an $O(k^3)$-vertex kernel when k is the vertex cover size, and an $O(k^4)$-vertex kernel when k is the size of a feedback vertex set of the input graph. On the other hand, it was shown that no polynomial-size kernels can be obtained for the distance to cluster graphs or the distance to chordal graphs [10]. Using the structural parameter map in Figure 1 one can now identify parameters that are weaker or stronger than the previously considered parameters and thus identify the following open questions:

- Does Treewidth admit a problem kernel with $O(k^3)$ vertices when k denotes the "distance to disjoint paths"? As shown in Figure 1, this parameter is stronger than the vertex cover size and weaker than the feedback vertex set number.
- Is there a parameter between feedback vertex set size and treewidth, for example "distance to outerplanar graphs" for which Treewidth admits a polynomial-size problem kernel [9]?

Long Path and Long Cycle. In the NP-hard problems Long Path and Long Cycle one is given an undirected graph and asks for the existence of a simple path (or cycle, respectively) of length at least k. Both problems do not admit polynomial problem kernels with respect to the parameter solution size [8].

[5] The graph was parsed using the data from http://dblp.uni-trier.de/xml/.

Motivated by this fact, Bodlaender et al. [10] studied both problems with respect to their kernelizability when parameterized by structural parameters. For instance, they showed that Long Cycle

- admits an $O(k^2)$-vertex problem kernel when parameterized by the vertex cover size,
- admits a polynomial problem kernel when parameterized by the size of the cluster vertex deletion set,
- does not admit a polynomial-size kernel when parameterized by the size of a vertex set whose deletion produces an outerplanar graph.

Since all forests are outerplanar, the parameter "distance to outerplanar graphs" is stronger than the parameter feedback vertex set number. Hence, Bodlaender et al. [10] posed the question whether Long Cycle admits a polynomial-size kernel when parameterized by feedback vertex set number. Assuming this question were settled, this would immediately raise another open question:

- In case Long Cycle admits a polynomial-size problem kernel when parameterized by the feedback vertex set number: is there a parameter that is stronger than feedback vertex set but for which Long Cycle still admits a polynomial-size problem kernel?
- In case Long Cycle does not admit a polynomial-size problem kernel when parameterized by the feedback vertex set number: is there a parameter stronger than vertex cover and weaker than feedback vertex set number, for which Long Cycle admits a polynomial kernel?

Furthermore, Bodlaender et al. [10] asked whether Long Cycle admits a polynomial-size problem kernel when parameterized by the "distance to co-graph". Similar to the discussion above, the answer to this question immediately raises a new question: either is there a parameter stronger than distance to co-graph such that Long Cycle admits a polynomial-size problem kernel, or is there a parameter weaker than distance to co-graph and stronger than cluster vertex deletion for which this is the case?

To conclude, navigating through the space of structural parameters (of which Figure 1 shows an excerpt) helps in identifying new research directions and open questions. Many of these questions are of purely algorithmic nature in the sense that the main question is fixed-parameter tractability (or the existence of a polynomial-size problem kernel) for a specific parameter. Some of these questions, however, also ask for the "discovery" of hidden parameters and are thus also closely related to general combinatorial aspects of the considered input structure.

3 Parameterizations Related to Solution Size

In this section, we present two examples for algorithmic studies in which parameterizations that are stronger than the "classical" parameterization by solution size have been proposed.

Above-Guarantee Parameterizations. Vertex Cover is one of the best-studied problems in parameterized algorithmics.

> Vertex Cover
> Input: An undirected graph G, and an integer k.
> Question: Is there a vertex set S of size at most k such that deleting S from G results in an independent set?

As mentioned in the introduction, the $f(k)$-race for Vertex Cover parameterized by vertex cover size k has brought a significant running time improvement. The vertex cover of a graph, however, is often relatively large. In order to obtain more meaningful parameters, Mahajan and Raman [35] proposed to study "parameterizations above guaranteed values". For vertex cover, one such parameterization is "parameterization above matching lower bound", that is, one asks whether there is a vertex cover of size at most $\ell + k$, where ℓ is the size of a maximum matching in the input graph. Clearly one vertex has to be used to cover each of the m matching edges and the parameter k measures the excess of this bound. The fixed-parameter tractability of Vertex Cover for this parameter follows from a fixed-parameter tractability result for Almost 2-SAT [41]. An $f(k)$-race for this parameter resulted in several improvements [15, 40].

Recently, Narayanaswamy et al. [36] showed that Vertex Cover is fixed-parameter tractable with respect to the "parameterization above linear program (LP) lower bound", that is, one asks whether there is a vertex cover of size at most $\ell + k$ where k is the value of an optimal solution of the LP relaxation of an ILP formulation of Vertex Cover. Again, the excess k above the lower bound is the parameter. Since the LP lower bound is at least as large as the matching lower bound, the new above-guarantee parameter is stronger. The fixed-parameter result by Narayanaswamy et al. [36] now starts a new $f(k)$-race for this parameter. For all three of the above-mentioned parameters there are algorithms that solve the problem in $c^k \cdot \text{poly}(n)$ time where c is a small constant. One interesting open question is thus: Is there an even stronger parameter for which a running time of $c^k \cdot \text{poly}(n)$ can be achieved, where c is relatively small.

Cluster Editing. Cluster Editing is a well-studied problem with applications in graph-based data clustering [28].

> Cluster Editing
> Input: An undirected graph G and an integer k.
> Question: Can G be transformed into a vertex-disjoint union of cliques (a so-called *cluster graph*) by at most k edge modifications?

Most investigations for Cluster Editing were concerned with the two classical FPT-races running time and problem kernel size for the parameter solution size k. More recently, other parameterizations for Cluster Editing have been considered. For instance, it was shown that Cluster Editing is fixed-parameter tractable when parameterized by the stronger parameter cluster vertex deletion number [32, 46] but NP-hard already on graphs of bounded degree [25, 33]. Furthermore, a parameter called "local modification bound" was introduced [33]. This parameter is a stronger parameter than the solution size: The solution size is the overall number of edge modifications; the local modification bound is the maximum of incident edge modifications over all vertices of the graph. Cluster Editing admits a kernel with at most $O(d \cdot t)$ vertices where d is an upper bound on the number of clusters in the cluster graph and t is the local modification bound [33] .

By means of a simple dynamic programming algorithm, the above problem kernel result implies an algorithm with running time $2^{O(dt)}$ for Cluster Editing. In contrast, for the parameter "solution size k and cluster number d", Cluster Editing can be solved in $2^{O(\sqrt{dk})} \cdot \mathrm{poly}(n)$ time [25]. On the other hand, $2^{O(\sqrt{dt})} \cdot \mathrm{poly}(n)$ time is unlikely to be achievable [33]. Hence, is there a parameter x between solution size k and local modification bound t for which an $2^{O(\sqrt{dx})} \cdot \mathrm{poly}(n)$-time algorithm can be achieved? For instance, it would be interesting to consider the parameter number ℓ of edge deletions performed by a solution to Cluster Editing:

– Is Cluster Editing fixed-parameter tractable with respect to the number ℓ of edge deletions performed by a size-k solution?
– Does Cluster Editing admit a problem kernel that is polynomial in ℓ?
– Can Cluster Editing be solved in $2^{O(\ell)} \cdot \mathrm{poly}(n)$ or in $2^{O(\sqrt{d\ell})} \cdot \mathrm{poly}(n)$ time?

Summarizing, a very natural way of obtaining parameterizations that are stronger than the solution size is the approach of "parameterizing above guarantee" as discussed in the Vertex Cover example. The identification of stronger parameters, however, is not limited to this approach as demonstrated by Cluster Editing where simple combinatorial considerations have been used to identify parameters that are stronger than the solution size parameterization.

4 Conclusion

With the advent of parameterized algorithmics from single-parameter studies to multi-parameter studies, investigating the relation between parameters becomes more and more important. In this line, studying the "stronger"-relation between parameters as proposed in this paper is a natural and fruitful undertaking, directly leading to numerous challenges on designing efficient and practically relevant fixed-parameter algorithms. This is encompassed by challenging combinatorial questions related to the "parameter space" (cf. Figure 1) associated with a specific problem. The study of the relationship between structural parameters measuring the input complexity should not be limited to graph problems. Indeed, obtaining analogous parameter spaces for "string parameters and problems" or "set system parameters and problems" is an open field.

The topic of alternative races in parameterized algorithmics is closely related to multivariate algorithm design and analysis [21, 39] in at least two ways. First, in both approaches it is essential that various parameters come into play. Second, when combining parameters as done in the multivariate design of fixed-parameter algorithms, then this usually happens in order to either improve the running time of a fixed-parameter tractability result towards practical relevance (by adding parameters one clearly further restricts the generality of the result, hopefully for the benefit of improved efficiency) or to turn W[1]-hardness with respect to a specific parameterization into fixed-parameter tractability by further adding one or more parameters. In this spirit, a parameter k_1 is likely to be stronger than a combined parameter (k_1, k_2), whatever k_2 is. So, parameter addition gives a way to generate weaker parameters. Let us describe one specific open questions in this context: It is open whether the NP-hard arc routing problem Rural Postman is fixed-parameter tractable with respect to the number of weakly connected components of the graph induced by the "required arcs" [27, 43]. As a step towards answering this long-standing open problem, it is interesting to identify weaker parameters making the problem fixed-parameter tractable.

To conclude, let us only briefly mention two further efficiency races in the context of parameterized algorithmics. First, there are many further problems that can be studied along the lines of this article. In particular, it is a challenge to extend the presented approaches for graph-theoretic problems to more complex real-world problems. For instance, the NP-hard Target Set Selection (modeling the spread of influence) is a prominent graph problem occurring in social networks. It is known to be W[1]-hard with respect to the parameters treewidth and also the weaker parameter feedback vertex set size [2] while it becomes fixed-parameter tractable for the still weaker parameter vertex cover number [37]. Second, a standard way of starting races is instead of making the parameter stronger is to make the underlying classes of input instances larger. For instance, not changing the parameter, does a result holding for planar graphs generalize to bounded-genus graphs and further to certain classes of minor-free graphs? This is closer to our approach than it might appear at first sight since many structural graph parameters basically also restrict the allowed graph types, thus also defining a specific graph class.

Finally, from a more applied point of view, the most natural way of spotting relevant parameterizations is to make measurements in the (real-world) input data. Quantities that turn out to be small particularly qualify for parameterized complexity studies. To this end, tools for data analysis are needed.[6] After having performed the data analysis task, one may set up the parameter navigation map and perform algorithmic studies as sketched in this article.

Acknowledgments. We thank André Nichterlein, Manuel Sorge, and Mathias Weller for their comments which have improved this article.

[6] We have implemented the *Graphana* tool, which can be used to compute or estimate graph parameters as shown in Figure 1. The software is available from http://fpt.akt.tu-berlin.de/graphana.

References

[1] B. Balasundaram, S. Butenko, and S. Trukhanovzu. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

[2] O. Ben-Zwi, D. Hermelin, D. Lokshtanov, and I. Newman. Treewidth governs the complexity of target set selection. *Discrete Optimization*, 8(1):87–96, 2011.

[3] R. van Bevern. Towards optimal and expressive kernelization for d-Hitting Set. In *Proceedings of the 18th Annual International Computing and Combinatorics Conference (COCOON '12)*, volume 7434 of *Lecture Notes in Computer Science*, pages 121–132. Springer, 2012.

[4] R. van Bevern, S. Hartung, F. Kammer, R. Niedermeier, and M. Weller. Linear-time computation of a linear problem kernel for dominating set on planar graphs. In *Proceedings of the 6th International Symposium on Parameterized and Exact Computation (IPEC '11)*, volume 7112 of *Lecture Notes in Computer Science*, pages 194–206. Springer, 2012.

[5] H. L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

[6] H. L. Bodlaender and A. M. C. A. Koster. Safe separators for treewidth. *Discrete Mathematics*, 306(3):337–350, 2006.

[7] H. L. Bodlaender, A. M. C. A. Koster, and F. van den Eijkhof. Preprocessing rules for triangulation of probabilistic networks. *Computational Intelligence*, 21(3): 286–305, 2005.

[8] H. L. Bodlaender, R. G. Downey, M. R. Fellows, and D. Hermelin. On problems without polynomial kernels. *Journal of Computer and System Sciences*, 75(8): 423–434, 2009.

[9] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Preprocessing for treewidth: A combinatorial analysis through kernelization. In *Proceedings of the 38th International Colloqium on Automata, Languages and Programming (ICALP '11)*, volume 6755 of *Lecture Notes in Computer Science*, pages 437–448. Springer, 2011.

[10] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernel bounds for path and cycle problems. In *Proceedings of the 6th International Symposium on Parameterized and Exact Computation (IPEC '11)*, volume 7112 of *Lecture Notes in Computer Science*, pages 145–158. Springer, 2012.

[11] Y. Cao, J. Chen, and Y. Liu. On feedback vertex set, new measure and new structures. In *Proceedings of the 12th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT '10)*, volume 6139 of *Lecture Notes in Computer Science*, pages 93–104. Springer, 2010.

[12] J. Chen and J. Meng. A $2k$ kernel for the cluster editing problem. *Journal of Computer and System Sciences*, 78(1):211–220, 2012.

[13] J. Chen, I. A. Kanj, and G. Xia. Improved upper bounds for vertex cover. *Theoretical Computer Science*, 411(40-42):3736–3756, 2010.

[14] M. Cygan, J. Nederlof, M. Pilipczuk, M. Pilipczuk, J. M. M. van Rooij, and J. O. Wojtaszczyk. Solving connectivity problems parameterized by treewidth in single exponential time. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '11)*, pages 150–159. IEEE, 2011.

[15] M. Cygan, M. Pilipczuk, M. Pilipczuk, and J. O. Wojtaszczyk. On multiway cut parameterized above lower bounds. In *Proceedings of the 6th International Symposium on Parameterized and Exact Computation (IPEC '11)*, volume 7112 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2011.

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

[17] A. Drucker. New limits to classical and quantum instance compression. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS '12)*, 2012. To appear.

[18] V. Dujmovic, M. R. Fellows, M. T. Hallett, M. Kitching, G. Liotta, C. McCartin, N. Nishimura, P. Ragde, F. A. Rosamond, M. Suderman, S. Whitesides, and D. R. Wood. A fixed-parameter approach to 2-layer planarization. *Algorithmica*, 45(2): 159–182, 2006.

[19] D. Eppstein and E. S. Spiro. The h-index of a graph and its application to dynamic subgraph statistics. In *Proceedings of the 11th International Algorithms and Data Structures Symposion (WADS 2009)*, volume 5664 of *Lecture Notes in Computer Science*, pages 278–289. Springer, 2009.

[20] P. Erdős and A. Hajnal. On chromatic number of graphs and set-systems. *Acta Mathematica Academiae Scientiarum Hungaricae*, 17:61–99, 1966.

[21] M. R. Fellows. Towards fully multivariate algorithmics: Some new results and directions in parameter ecology. In *Proceedings of the 20th International Workshop on Combinatorial Algorithms (IWOCA'09)*, volume 5874 of *Lecture Notes in Computer Science*, pages 2–10. Springer, 2009.

[22] H. Fernau. Two-layer planarization: Improving on parameterized algorithmics. *Journal of Graph Algorithms and Applications*, 9(2):205–238, 2005.

[23] J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer, 2006.

[24] J. Flum, M. Grohe, and M. Weyer. Bounded fixed-parameter tractability and $\log^2 n$ nondeterministic bits. *Journal of Computer and System Sciences*, 72(1): 34–71, 2006.

[25] F. V. Fomin, S. Kratsch, M. Pilipczuk, M. Pilipczuk, and Y. Villanger. Subexponential fixed-parameter tractability of cluster editing. *CoRR*, abs/1112.4419, 2011.

[26] L. Fortnow and R. Santhanam. Infeasibility of instance compression and succinct PCPs for NP. *Journal of Computer and System Sciences*, 77(1):91–106, 2011.

[27] G. N. Frederickson. Approximation algorithms for some postman problems. *Journal of the ACM*, 26(3):538–554, 1979.

[28] J. Guo. Fixed-parameter algorithms for graph-modeled date clustering. In *Proceedings of the 6th Annual Conference on Theory and Applications of Models of Computation (TAMC '09)*, volume 5532 of *Lecture Notes in Computer Science*, pages 39–48. Springer, 2009.

[29] T. Hagerup. Simpler linear-time kernelization for planar dominating set. In *Proceedings of the 6th International Symposium on Parameterized and Exact Computation (IPEC '11)*, volume 7112 of *Lecture Notes in Computer Science*, pages 181–193. Springer, 2012.

[30] S. Hartung, C. Komusiewicz, and A. Nichterlein. On structural parameterizations for the 2-Club problem: Classical and parameterized hardness, 2012. Manuscript, June 2012.

[31] K. Kawarabayashi and B. A. Reed. Computing crossing number in linear time. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC '07)*, pages 382–390. ACM, 2007.

[32] C. Komusiewicz and J. Uhlmann. Alternative parameterizations for cluster editing. In *Proceedings of the 37th International Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM '11)*, volume 6543 of *LNCS*, pages 344–355. Springer, 2011.

[33] C. Komusiewicz and J. Uhlmann. Cluster editing with locally bounded modifications. *Discrete Applied Mathematics*, 2012. Online available.

[34] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

[35] M. Mahajan and V. Raman. Parameterizing above guaranteed values: MaxSat and MaxCut. *Journal of Algorithms*, 31(2):335–354, 1999.

[36] N. S. Narayanaswamy, V. Raman, M. S. Ramanujan, and S. Saurabh. LP can be a cure for parameterized problems. In *Proceedings of the 29th International Symposium on Theoretical Aspects of Computer Science (STACS '12)*, volume 14 of *LIPIcs*, pages 338–349. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[37] A. Nichterlein, R. Niedermeier, J. Uhlmann, and M. Weller. On tractable cases of target set selection. *Social Network Analysis and Mining*, 2012. Online available.

[38] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Number 31 in Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2006.

[39] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proceedings of the 27th International Symposium on Theoretical Aspects of Computer Science (STACS'10)*, volume 5 of *LIPIcs*, pages 17–32. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010.

[40] V. Raman, M. S. Ramanujan, and S. Saurabh. Paths, flowers and vertex cover. In *Proceedings of the 19th Annual European Symposium on Algorithms (ESA '11)*, volume 6942 of *Lecture Notes in Computer Science*, pages 382–393. Springer, 2011.

[41] I. Razgon and B. O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *Journal of Computer and System Sciences*, 75(8):435–450, 2009.

[42] A. Schäfer, C. Komusiewicz, H. Moser, and R. Niedermeier. Parameterized computational complexity of finding small-diameter subgraphs. *Optimization Letters*, 6 (5):883–891, 2012.

[43] M. Sorge, R. van Bevern, R. Niedermeier, and M. Weller. A new view on rural postman based on Eulerian extension and matching. *Journal of Discrete Algorithms*, 2012. Available online.

[44] M. Suderman. *Layered Graph Drawing.* PhD thesis, School of Computer Science, McGill University, 2005.

[45] S. Thomassé. A $4k^2$ kernel for feedback vertex set. *ACM Transactions on Algorithms*, 6(2), 2010.

[46] J. Uhlmann. *Multivariate Algorithmics in Biological Data Analysis.* PhD thesis, Technische Universität Berlin, Berlin, Germany, 2011.

[47] J. Uhlmann and M. Weller. Two-layer planarization parameterized by feedback edge set. In *Proceedings of the 7th Annual Conference on Theory and Applications of Models of Computation (TAMC '10)*, volume 6108 of *Lecture Notes in Computer Science*, pages 431–442. Springer, 2011.

[48] M. Weller. An improved branching algorithm for two-layer planarization parameterized by the feedback edge set number. Manuscript, 2012.