The Complexity Ecology of Parameters: An Illustration Using Bounded Max Leaf Number ⋆

Michael Fellows[1,2] and Frances Rosamond[1]

[1] University of Newcastle, Callaghan NSW 2308, Australia
{michael.fellows,frances.rosamond}@newcastle.edu.au
[2] Durham University, Institute of Advanced Study,
Durham DH1 3RL, United Kingdom

Abstract. In the framework of parameterized complexity, exploring how one parameter affects the complexity of a different parameterized (or unparameterized problem) is of general interest. A well-developed example is the investigation of how the parameter *treewidth* influences the complexity of (other) graph problems. The reason why such investigations are of general interest is that real-world input distributions for computational problems often inherit structure from the natural computational processes that produce the problem instances (not necessarily in obvious, or well-understood ways). The *max leaf number* of a connected graph G is the maximum number of leaves in a spanning tree for G. Exploring questions analogous to the well-studied case of treewidth, we can ask: how hard is it to solve 3-Coloring or Hamilton Path or Minimum Dominating Set for graphs of bounded max leaf number? We do two things:
(1) We describe much improved FPT algorithms for a large number of graph problems, for input of bounded max leaf number, based on the polynomial-time extremal structure theory associated to the parameter max leaf number.
(2) The way that we obtain these concrete algorithmic results is general and systematic. We describe the approach.

1 Introduction

The analysis of the complexity of problems, for graphs of bounded treewidth, is well-developed and supports many systematic approaches that have developed over a number of years [AP89,ALS91,MP94,Bod96,DF99,Nie06,BK07]. For example, determining whether a graph is 3-colorable can be solved in time $O(n)$ for

⋆ This research has been supported by the Australian Research Council through the Australian Centre for Bioinformatics, by the University of Newcastle Parameterized Complexity Research Unit under the auspices of the Deputy Vice-Chancellor for Research, and by a Fellowship to the Durham University Institute for Advanced Studies. The authors also gratefully acknowledge the support and kind hospitality provided by a William Best Fellowship at Grey College while the paper was in preparation.

graphs of treewidth at most k. In the terminology of parameterized complexity [DF99,FG06], Graph 3-Coloring is *fixed parameter tractable* for the parameter *treewidth*. In this small example, the asymptotic notation conceals serious costs associated to the treewidth bound k, from two sources:
(1) The complexity of computing a tree-decomposition of width k is $O(2^{35k^3}n)$ for an n-vertex graph.
(2) Once the tree-decomposition is obtained, one would then solve the problem by dynamic programming, in time $O(3^k n)$.

Suppose that we wish to solve Graph 3-Coloring for graphs having a different structural restriction — how should this be done? Here we consider the structural parameter of *bounded max leaf number*, where this is defined for a connected graph G as the maximum number of leaves of a spanning tree for G. (We choose this parameter mainly to illustrate the key issues, and because enough is known of the associated P-time extremal structure theory to provide a good example of the general approach. We are not aware of any strong direct applications of bounded max leaf number.)

One way to approach the problem of determining 3-colorability, parameterizing by max leaf number, is to note that graphs of bounded max leaf number exclude a tree minor and therefore have bounded pathwidth, so that the above sketched bounded treewidth approach can be used. This classifies Graph 3-Coloring, parameterized by max leaf number, as FPT, but this is not an efficient algorithm.

We have two objectives in this paper:
(1) We describe *efficient* FPT algorithms for Graph 3-Coloring and many other problems, for input parameterized by max leaf number.
(2) We do so in a way that is very generally systematic, and that "fits" the study of how parameterized structure affects computational complexity in what we term the "ecology" of parameterized complexity. One can view this effort as a kind of *generalized bidimensionality theory* in the sense of Demaine and Hajiaghayi [DFHT05,DH05,DH07].

In the next section, we discuss the basics of parameterized complexity and motivate the general setting for this investigation.

2 Background on Parameterized Complexity and the Complexity Ecology of Parameters

Contemporary sources of introductory material can be found in the survey articles [Ra97,DFS99,Fe02,Dow03,Nie04], and in the recent books and monographs [DF99,FG06,Nie06].

Parameterized complexity is basically a two-dimensional generalization of the familiar P versus NP framework. In addition to the overall input size n we consider the effects on complexity of a declared secondary "measurement" k (the *parameter*) that generally is used to capture some structure of the input or other aspect of our computational objective (for example, $k = 1/\epsilon$ turns out to be a useful parameterization in the analysis of approximation complexity).

Solvability in time $f(k)n^c$ is termed *fixed-parameter tractability* (FPT), where f is some function (usually exponential) and c is a constant independent of k.

Evidence that a parameterized problem is unlikely to admit an FPT algorithm is provided by a strong two-dimensional analog of NP-hardness, termed $W[1]$-hardness. A reference problem complete for $W[1]$ is the k-step halting problem for nondeterministic Turing machines of unlimited nondeterminism and alphabet size. This is obviously solvable by brute force in time $O(n^{O(k)})$. The positive toolkit of FPT turns out to be technically quite rich, and the negative toolkit of $W[1]$-hardness turns out to be widely applicable.

The main motivation for parameterized complexity is that in almost all real world settings and for almost all purposes of computing, the input has “extra structure” that we are able to relevantly capture with the mathematical device of the parameter.

Historically, a key motivating source for parameterized complexity has been the graph minors project of Robertson and Seymour [RS85]. The graph minors structure theory is related to FPT in the following way. The parameterized computational decision problem GRAPH MINOR takes as input graphs G and H and asks whether H is a minor of G (that is, whether a graph isomorphic to H can be obtained from G by contracting edges of a subgraph of G). This is a fundamental problem, naturally parameterized by H.

As far as we know, all of the beautiful structure theory of the graph minors project, *pathwidth*, *treewidth*, and the like, is necessary in order to show that the GRAPH MINOR problem, parameterized by H, is fixed-parameter tractable.

The following lemma codifies how every FPT parameterized problem has a canonically associated structure theory project, via the quest for efficient FPT kernelization bounds.

Lemma 1. *A parameterized problem Π is in FPT if and only if there is a transformation from Π to itself, and a function g, that reduces an instance (x, k) to (x', k') such that:*
(1) the transformation runs in time polynomial in $|(x, k)|$,
(2) (x, k) is a yes-instance of Π if and only if (x', k') is a yes-instance of Π,
(3) $k' \leq k$, and
(4) $|x'| \leq g(k)$.

In the situation described above, we say that we have a *kernelization bound* of $g(k)$. The proof of the above “point of view” on FPT that focuses on P-time kernelization is completely trivial, giving a kernelization bound of $g(k) = f(k)$ for an FPT problem solvable in time $f(k)n^c$. But for many important FPT problems, we can do *much* better, and the “pre-processing” routines that produce small kernels seem to have great practical value [ACFLSS04,Nie04,Nie06,Wei98]. For example, the VERTEX COVER problem can be kernelized in polynomial time to a graph on at most $2k$ vertices [NT75,ACFLSS04,CFJ04]. PLANAR DOMINATING SET also has a problem kernel of linear size [AFN04].

2.1 A Complexity Ecology of Parameters

The extent to which the structure theory of the graph minors project has turned out to be of practical relevance to computing is quite striking. For one example, many naturally occuring databases have bounded treewidth (or bounded hypertreewidth, a related notion) — a matter of immense significance to the realistic assessment of the complexity of database problems [GM99,FFG01,Gr01].

Another example is the problem of Type Checking of programs written in high-level logic-based programming languages such as ML. This problem has been shown to be complete for EXP, and thus "extremely" intractable from the classical point of view. Nevertheless, the ML compilers generally work just fine. The explanation is that most naturally occuring programs have a maximum type-declaration nesting depth k of no more than 5. The FPT type-checking algorithm that runs in time $O(2^k n)$ is entirely adequate in practice. The reason why naturally occuring programs have small nesting depth is that otherwise the programs quickly become incomprehensible to the programmer.

A possible perspective on this quoted from the survey [DFS99]:

> We feel that the parametric complexity notions, with their implicit ultrafinitism, correspond better to the natural landscape of computational complexity, where we find ourselves overwhelmingly among hard problems, dependent on identifying and exploiting thin zones of computational viability. Many natural problem distributions are generated by processes that inhabit such zones themselves (e.g., computer code that is written in a structured manner so that it can be comprehensible to the programmer), and these distributions then inherit limited parameter ranges because of the computational parameters that implicitly govern the feasibility of the generative processes, though the relevant parameters may not be immediately obvious. [3]

We want to know how all the various parameterized structural notions interact with all the other computational objectives one might have. The familiar paradigm of efficiently solving various problems for graphs of bounded treewidth just represents one row of a matrix of algorithmic questions that arise from the relevant parameterized structure theories. In the case of Max Leaf, we investigate how to solve the Independent Set problem, etc., on graphs bounded "max leaf number", exploiting the structure that bounding this parameter yields.

Consider the following table. We use here the shorthand: TW is Treewidth, BW is Bandwidth, VC is Vertex Cover, DS is Dominating Set, G is Genus and ML is Max Leaf. The entry in the 2nd row and 4th column indicates that there is an *FPT* algorithm to optimally solve the Dominating Set problem for a graph G of bandwidth at most k. The entry in the 4th row and second column indicates that it is unknown whether Bandwidth can be solved optimally by an *FPT* algorithm when the parameter is a bound on the domination number of the input.

[3] For a philosophically similar discussion see [Gur89].

	TW	BW	VC	DS	G	ML
TW	*FPT*	$W[1]$-hard	*FPT*	*FPT*	?	*FPT*
BW	*FPT*	$W[1]$-hard	*FPT*	*FPT*	?	*FPT*
VC	*FPT*	?	*FPT*	*FPT*	?	*FPT*
DS	?	?	$W[1]$-hard	$W[1]$-hard	?	?
G	$W[1]$-hard	$W[1]$-hard	$W[1]$-hard	$W[1]$-hard	*FPT*	?
ML	*FPT*	?	*FPT*	*FPT*	*FPT*	*FPT*

Table 1. The Complexity Ecology of Parameters

Our attention so far has mostly been concerned with:
(1) The diagonal — for example, TREEWIDTH is *FPT* and BANDWIDTH is $W[1]$-hard — as stand-alone problems, and
(2) The first row.
But if the natural world of complexity "runs" on a commerce of (sometimes rather hidden) structural parameters, then it is important to systematically investigate the entire matrix. The so-called *bidimensionality theory* gives a systematic approach to the first (treewidth) row [DH07].

3 Systematically Attacking a Row

We use the max leaf parameter to show how to systematically attack a row of the "complexity ecology table" (which should not be thought of as limited to the few illustrative examples of problems in the table above — the real table is unbounded). We use the P-time extremal theory approach that is developed in [Pr05,EFLR05] where it is used to give a $3.75k$ P-time kernelization for the parameterized MAX LEAF problem. The main point here is how to deploy such P-time kernelization structure theory to prove FPT results in a row of the complexity ecology table. The next two theorems extended and adapted from [EFLR05] illustrate the approach. (We depend heavily and unavoidably on this previous work.)

Theorem 1. *For graphs of max leaf number bounded by k, the minimum domination number can be computed in time $O^*(103^k)$ based on a polynomial-time reduction to a kernel of size at most $7k$.*

Proof. Sketch. Since this is an FPT result, we are necessarily (by Lemma 1) interested in effective kernelization for *this* problem. We must therefore develop a polynomial-time extremal account of the boundary case for the induction.

We take the following hypotheses:
(1) (G, k) is a yes-instance of MAX LEAF.
(2) $(G, k + 1)$ is a no-instance of MAX LEAF.
(3) There is a witness structure for (1) that satisfies the inductive priorities of the proof of Boundary Lemma II for MAX LEAF (Lemma 8 of [EFLR05]).
(4) G is *reduced* according to an admissible set of polynomial time kernelization rules.

Here we must confine the interpretation of *reduced* to P-time reduction rules that are compatible with the new computational objective of computing a minimum dominating set. Many of the structural claims proved in [EFLR05] can now be imported to this new situation, modified in some cases because of changes to the admissible set of reduction rules. To illustrate the point, when proving a kernelization bound for Max Leaf (as is done in [EFLR05]), one uses reduction rules that can be applied in polynomial time to produce from G a graph G' such that G has a k-leaf spanning tree if and only if G' has a k' leaf spanning tree, where $k' \leq k$. Here, because we are computing a minimum dominating set, we are allowed reduction rules where G has a k-dominating set if and only if G' has a k' dominating set. To the extent that we can find reduction rules for this new computational objective that "mimic" or approximate the ones that were available for the Max Leaf problem, the structural claims about the kernel still (with some modifications) carry over, and we can conclude similar kernelization bounds for problems in the row of the complexity ecology table that are amenable to this approach. (It turns out that many well-known NP-hard problems are amenable in this way, and this is the main point of the paper.)

The reduction rules shown in Figure 1 below can be used in this way for the Minimum Dominating Set problem, for graphs of bounded max leaf number.

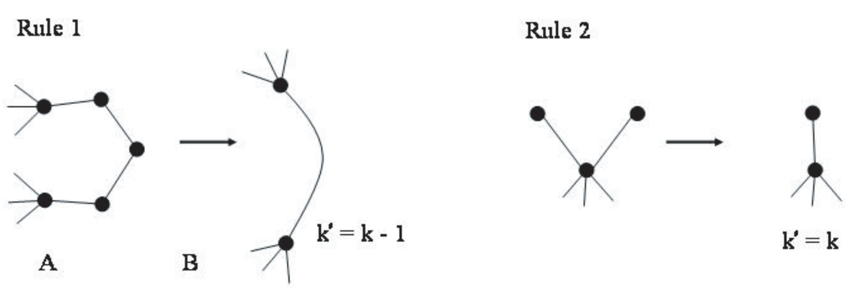

Fig. 1. Reduction rules for minimum domination.

The argument for the bound on the kernel size is by minimum counterexample. One of our hypotheses is that (G, k) is a yes-instance for Max Leaf. We can assume we are given as a witness structure a tree subgraph $T = (V', E')$ of G that has k leaves, and we can also assume that G is connected.

We do not assume that T is a spanning subgraph. (If T is not spanning, then it clearly extends to a spanning tree T' for G that has at least k leaves.)

A counterexample to our theorem would be a graph $G = (V, E)$ such that: (1) (G, k) is a reduced instance of Max Leaf, (2) (G, k) is a yes-instance of Max Leaf, (3) $(G, k + 1)$ is a no-instance, and (4) $|G| > 7k$.

Among all such counterexamples, we consider one where the witness subgraph tree T is as small as possible.

Let $O = V - V'$ be the set of vertices not in the witness subtree T, which we will refer to as *outsiders*. Let L denote the leaves of T, I the internal (non-leaf) vertices of T, $B \subseteq I$ the *branch vertices* of T (the non-leaf, internal vertices of T that have degree at least 3 with respect to T), and let J denote the *subdivider vertices* of T (the non-branch internal vertices of T that have degree 2 with respect to T). See Figure 2 for an illustration of the general situation. One of the key roles of the reduction rules is to bound the number of outsiders (Claim 7 of Lemma 7 of [EFLR05]). In pursuing this sketch, which summarizes much material in [EFLR05], necessarily many details are omitted.

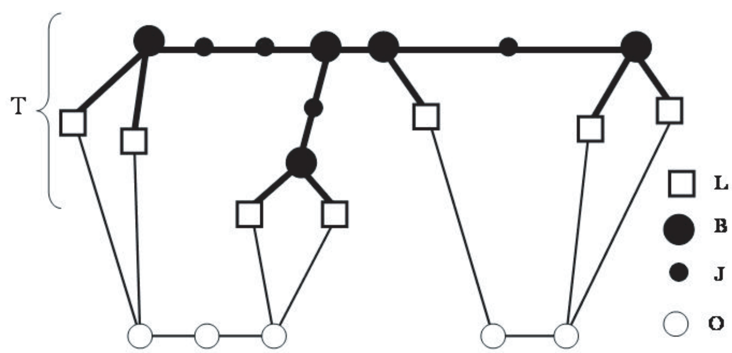

Fig. 2. The witness tree and various sets of vertices.

Almost all of the structural claims in the proof of Boundary Lemma II of [EFLR05] carry over (with a few requiring slight modification), yielding a kernel of size at most $7k$. The kernel can be analyzed by means of the algorithm due to Fomin, Kratsch and Woeginger [FKW04], yielding the running time stated for our algorithm. Knowing the domination number of the problem kernel allows us to compute the domination number of the input graph by retracing this information backwards along the kernelization path in polynomial time. $\square$

What was the best previous result for this problem? Using the structure theory of Boundary Lemma II of [EFLR05] we can show that a path decomposition of width at most $g(k) = 20k/3$ can be computed in polynomial time for graphs whose max leaf number is bounded by k. Combining this with the carefully engineered dynamic programming algorithm for DOMINATING SET in this setting of Telle and Proskurowsky [TP93] (refined by Alber and Niedermeier [AN02]) one would get a "best previous" running time of around $O^*(4^{20k/3})$ or $O^*(10322^k)$.

The following theorem is also reported in [EFLR05], based on essentially the same approach, making use of the reduction rules shown in Figure 3. (Quick sketch: The imported structural claims give a bound of $4.5k$ on the size of a vertex cover for the kernel, which yields the claimed running time by using the algorithm of Chen, Kanj and Xia [CKX05] to analyze the situation.)

Theorem 2. *For graphs of max leaf number bounded by k, the maximum size of an independent set can be computed in time $O^*(2.972^k)$ based on a polynomial-time reduction to a kernel of size at most $7k$.*

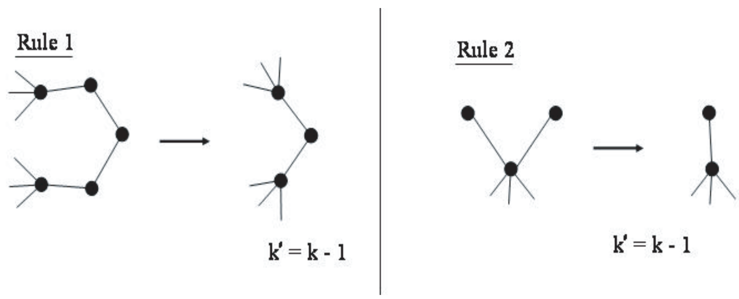

Fig. 3. Reduction rules for maximum independent set.

Many other NP-hard problems can be addressed for graphs of bounded max leaf number in much the same way.

Theorem 3. *For graphs of max leaf number bounded by k, it can be determined in $O^*(420.9^k)$ whether the graph is 3-colorable, based on a polynomial-time reduction to a kernel of size at most $5.5k$.*

Proof. The reduction rules: (1) delete vertices of degree 1, and (2) erase vertices of degree 2, are admissible for this problem. This yields an improved bound of $.5k$ over Claim 7 of Lemma 7 of [EFLR05]. The analysis of the kernel for the stated result just tries all possible 3-colorings. □

Graph Hamiltonicity admits the same reduction rules, and thus there is a $5.5k$ kernel for this problem as well. The same statement holds for the Feedback Vertex Set problem.

4 Summary

What we show in this paper is an example of how the structure theory associated (necessarily, by Lemma 1) to a good P-time kernelization result for an FPT problem (such as Max Leaf), can be exploited to give good FPT results for many of the entries in the corresponding "row" of the complexity ecology table. We have obviously picked off the easy examples, where all that is necessary is to identify reduction rules that are similar to the reduction rules used in the inductive proof of a kernelization bound for Max Leaf. Our results are not difficult, but the main point is the overall strategy, which clearly can be deepened. Some of the "columns" of our chosen row are still open. It is unclear how to use the Max Leaf kernel structure for the Bandwidth problem, for example. Since Bandwidth is NP-hard for trees (with unboundedly many leaves) this may be interesting to resolve.

References

[ACFLSS04] F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters and C. T. Symons. Kernelization algorithms for the vertex cover problem: theory and experiments. *Proceedings of the 6th Workshop on Algorithm Engineering and Experiments (ALENEX)*, New Orleans, January, 2004, ACM/SIAM, *Proc. Applied Mathematics 115*, L. Arge, G. Italiano and R. Sedgewick, eds.

[AFN04] J. Alber, M. Fellows and R. Niedermeier. Polynomial time data reduction for dominating set. *Journal of the ACM* 51 (2004), 363–384.

[AN02] J. Alber and R. Niedermeier. "Improved Tree Decomposition Based Algorithms for Domination-Like Problems," *Proceedings of the 5th Latin American Theoretical IN-formatics (LATIN 2002)*, Springer-Verlag LNCS 2286 (2002), 613–627.

[AP89] S. Arnborg and A. Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Disc. Appl. Math.* 23 (1989), 11–24.

[ALS91] S. Arnborg, J. Lagergren and D. Seese. Easy problems for tree-decomposable graphs. *J. Algorithms* 12 (1991), 308–340.

[BK07] H. L. Bodlaender and A. M. Koster. Combinatorial optimisation on graphs of bounded treewidth. To appear, *The Computer Journal*, 2007.

[Bod96] H. L. Bodlaender. A linear time algorithm for finding tree-decompositions of small width. *SIAM J. Computing* 25 (1996), 1305–1317.

[CFJ04] B. Chor, M. Fellows and D. Juedes. Linear kernels in linear time, or how to save k colors in $O(n^2)$ steps. *Proceedings WG 2004*, Springer-Verlag, *Lecture Notes in Computer Science* 3353 (2004), 257–269.

[CKX05] J. Chen, I. Kanj and G. Xia. Simplicity is beauty: improved upper bounds for vertex cover. Manuscript communicated by email, April, 2005.

[Cou90] B. Courcelle. The monadic second order logic of graphs I: Recognizable sets of finite graphs. *Information and Computation* 85 (1990), 12–75.

[DEFPR03] R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto-Rodriguez and F. Rosamond. Cutting up is hard to do: the complexity of k-cut and related problems. *Electronic Notes in Theoretical Computer Science* 78 (2003), 205–218.

[DF99] R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.

[DFHT05] E. D. Demaine, F. V. Fomin, M. Hajiaghayi and D. M. Thilikos. Subexponential parameterized algorithms on graphs of bounded genus and H-minor-free graphs. *Journal of the ACM* 52 (2005), 866–893.

[DFS99] R. Downey, M. Fellows and U. Stege. Parameterized complexity: a framework for systematically confronting computational intractability. In: *Contemporary Trends in Discrete Mathematics* (R. Graham, J. Kratochvil, J. Nesetril and

F. Roberts, eds.), Proceedings of the DIMACS-DIMATIA Workshop on the Future of Discrete Mathematics, Prague, 1997, *AMS-DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 49 (1999), 49–99.

[DH05] E. D. Demaine and M. Hajiaghayi. Bidimensionality: New connections between FPT algorithms and PTASs. *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005)*, Vancouver, January 2005, pp. 590–601.

[DH07] E. D. Demaine and M. Hajiaghayi. The bidimensionality theory and its algorithmic applications. *The Computer Journal*, to appear.

[Dow03] R. G. Downey. Parameterized complexity for the skeptic. *Proc. 18th IEEE Annual Conf. on Computational Complexity* (2003), 147–169.

[EFLR05] V. Estivill-Castro, M. Fellows, M. Langston and F. Rosamond. Fixed-parameter tractability is P-time extremal structure theory I: The case of max leaf. *Proceedings of ACiD 2005: Algorithms and Complexity in Durham* (2005), 1–41.

[Fe02] M. Fellows. Parameterized complexity: the main ideas and connections to practical computing. In: *Experimental Algorithmics*, Springer-Verlag, *Lecture Notes in Computer Science* 2547 (2002), 51–77.

[FFG01] J. Flum, M. Frick and M. Grohe. Query evaluation via tree-decompositions. *Proc. ICDT*, Springer-Verlag, *Lecture Notes in Computer Science* 1973 (2001), 22–32.

[FG06] *Parameterized Complexity Theory*, J. Flum and M. Grohe, Springer-Verlag, 2006.

[FKW04] F. Fomin, D. Kratsch and G. Woeginger. Exact (exponential) algorithms for the dominating set problem. *Proceedings of WG 2004*, Springer-Verlag, *Lecture Notes in Computer Science* 3353 (2004), 245–256.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman, 1979.

[GM99] M. Grohe and J. Marino. Definability and descriptive complexity on databases with bounded treewidth. *Proceedings of the 7th International Conference on Database Theory*, Springer-Verlag, *Lecture Notes in Computer Science* 1540 (1999), 70–82.

[Gr01] M. Grohe. The parameterized complexity of database queries. *Proc. PODS 2001*, ACM Press (2001), 82–92.

[Guo06] J. Guo. *Algorithm design techniques for parameterized problems.* Ph.D. Thesis, Friedrich-Schiller-Universität, Jena, 2006.

[Gur89] Y. Gurevich, "The Challenger-Solver Game: Variations on the Theme of *P=?NP*," *Bulletin EATCS* 39 (1989), 112–121.

[GGHNW05] J. Guo, J. Gramm, F. Hueffner, R. Niedermeier, S. Wernicke. Improved fixed-parameter algorithms for two feedback set problems. *Proceedings of WADS 2005*, Springer-Verlag, *Lecture Notes in Computer Science* 3608 (2005), 158–169.

[MP94] S. Mahajan and J. G. Peters, "Regularity and Locality in k-Terminal Graphs," *Discrete Applied Mathematics* 54 (1994), 229–250.

[Nie04] R. Niedermeier. Ubiquitous parameterization — invitation to fixed-parameter algorithms. In: *Mathematical Foundations of Computer Science MFCS 2004*, Springer-Verlag, *Lecture Notes in Computer Science* 3153 (2004), 84–103.

[Nie06] R. Niedermeier. *Invitation to Fixed Parameter Algorithms.* Oxford University Press, forthcoming, 2005.

[NT75] G. L. Nemhauser and L. E. Trotter. Vertex packings: structural properties and algorithms. *Mathematical Programming* 8 (1975), 232–248.

[Pr05] E. Prieto-Rodriguez. *Systematic kernelization in FPT algorithm design.* Ph.D. Thesis, School of EE&CS, University of Newcastle, Australia, 2005.

[Ra97] V. Raman, "Parameterized Complexity," in: *Proceedings of the 7th National Seminar on Theoretical Computer Science*, Chennai, India (1997), 1–18.

[RS85] N. Robertson and P. Seymour. Graph minors: a survey. In: J. Anderson, ed., *Surveys in Combinatorics*, Cambridge University Press (1985), 153–171.

[TP93] J.A. Telle and A. Proskurowski. "Practical Algorithms on Partial k-Trees with an Application to Domination-Like Problems." *Proceedings WADS'93 – The Third Workshop on Algorithms and Data Structures*, Springer-Verlag LNCS 709 (1993), 610–621.

[Wei98] K. Weihe. Covering trains by stations, or the power of data reduction. *Proc. ALEX'98* (1998), 1–8.