The Impact of Parameterized Complexity to Interdisciplinary Problem Solving

Ulrike Stege*

Department of Computer Science
University of Victoria
stege@cs.uvic.ca

Abstract. We discuss interdisciplinary parameterized complexity research in biology and cognitive science.

1 Introduction

Thinking back to my time as a PhD student in the Bioinformatics group at ETH Zürich, my first true encounter with Parameterized Complexity was when my colleague Chantal Korostensky and I followed an invitation from Mike Fellows to visit his research group in Victoria, British Columbia. We were both working on problems that were computationally hard—Chantal investigated methods to compute multiple sequence alignments [62] and I studied evolutionary tree reconciliation problems—and were fascinated hearing Mike's novel ideas of how to deal with NP-hard problems without necessarily sacrificing optimality.

In the fall of 1997, Mike explained to us his vision to view NP-hard computational problems in a more refined way and challenged us to study these parameterized versions in a new framework, called *parameterized complexity* [27]. Using graph theoretic examples such as Vertex Cover, Independent Set and Dominating Set he illustrated that parameterization can provide a better understanding of why a problem, which is characterized as intractable in the first place, may not be truly intractable, and what aspect of the problem may or may not contribute to its intractability. All these foundational illustrations can be found in the famous book by Downey and Fellows, which was at the time almost completed [18].

When considering complexity in the classical sense, the time complexity of an algorithm is measured in the input size n of the problem input. That is, if a problem is identified to be NP-hard, there is likely no polynomial time algorithm. In other words, there is no algorithm with a running time $O(n^c)$ where c is a constant.

In contrast, in the parameterized world, the problem input is considered in terms of the input size n and a parameter k. If there exists an algorithm solving the problem with a running time of $O(f(k)n^c)$, then this parameterization of the problem is *fixed-parameter tractable* or a member of the parameterized

* Research supported by an NSERC Discovery Grant.

H.L. Bodlaender et al. (Eds.): Fellows Festschrift, LNCS 7370, pp. 56–68, 2012.

complexity class *FPT* [18]. $f(k)$ can be a super-polynomial time function; in this case, the "intractability" of the problem is "trapped," that is, it depends on parameter k only and not on input size n. If $f(k)$ can be kept small for practical applications such that $f(k)$ is not "too big," then the algorithm should behave just like a polynomial-time algorithm!

I was intrigued by this radical idea. The realization that the (in)tractability of hundreds of computational problems could be (re)considered, and that many of these problems might have efficient parameterized algorithms and therefore might not be that intractable—or not that impractical—after all, opened up a world of possibilities for me, and gave me ideas for plenty of PhD topics. Particularly, the attraction to consider parameterizations from the application perspective of the problem gave me new hope to escape heuristic approaches in interdisciplinary areas, such as bioinformatics.

This lead me directly to the for me most convincing argument why we should continue to pursue parameterized complexity: namely because of the great potential of parameterized algorithms to be used to solve applied problems, by users from many different fields, including researchers in sciences and social sciences, as well as developers in industry.

My first aha-experience back then was that the NP-hardness property of a computational problem is by no means evidence for solid intractability but rather an encouragement to refine the problem statement using thoughtful parameterizations and to design parameterized algorithms. This was especially fascinating as it appeared to be counter to how I was taught and as I understood computational intractability at that time. Bodlaender *et al.* articulated this argument well in their 1995 article in *Computer Applications in the Biosciences* [5]. They argue that parameterized complexity (and not NP-completeness) is the appropriate tool for studying intractability. They further point out that parameterized intractability results can provide insights in possibly restricted versions of the problem as intractability results suggest (with respect to practical results) useful constraints.

Researchers who know Mike can easily guess that the first parameterized algorithm he showed to me to prove that a parameterized NP-complete problem can be in FPT was the bounded-search tree algorithm for Vertex Cover with its promising practical running time of $O(2^k n)$ [21].[1] An other powerful method from the FPT-toolkit to apply to Vertex Cover is efficient preprocessing, such as *kernelization* (cf. [9] for the first polynomial-time kernelization algorithm of Vertex Cover). Kernelization preprocesses the input with guarantees, namely it reduces a problem instance to an equivalent problem instance whose size is bounded by some function of the parameter, or it solves the parameterized problem: In the case of Vertex Cover parameterized by the size of the vertex cover to be determined, kernelization determines too large instances as no-instances [18]. For other parameterized problems, such as Max Leaf Spanning Tree parameterized by the number of leaves, kernelization reports large enough instances as yes-instances [28]. Especially desirable is polynomial-time

[1] Mehlhorn already described the basic idea of this elegant algorithm in 1984 [51].

kernelization. One of the most convincing arguments for the use of parameterized complexity in practice should be that a parameterized problem is in FPT if and only if it is kernelizable [19].

In general, preprocessing is a useful first step for solving any computational problem when dealing with large inputs or intractability properties. The beauty about Vertex Cover is not just its presentability for all kinds of parameterized algorithmic techniques (e.g., [43,54,13]), and its impressive sequence of parameterized algorithms [21,9,58,18,17,2,19,55,70,69,12,13], but also its applicability—it is a powerful model for, for example, conflict graph resolution as it can be applied in post processing of gaps multiple sequence alignments [69] or in cleaning data of a character matrix when building phylogenetic trees (also called Compatibility [16]).

During my PhD studies, Mike provided me with many opportunities to meet and have discussions with biologists (e.g., Joe Felsenstein, University of Washington in Seattle, USA; my now colleague and collaborator Chris Upton, University of Victoria in British Columbia, Canada; Jack Heinemann, University of Canterbury in Christchurch, New Zealand, with whom we had most fascinating conversations about horizontal gene transfer) to explore how the message of parameterized complexity could be applied in sciences, and what collaboration with scientists could look like. He also introduced me to Todd Wareham who was finishing his PhD studies with Mike, and my future graduate student Iris van Rooij who introduced me to another favourite application area of mine, viz. Cognitive Psychology.[2] I fondly remember these memorable meetings and experiences.

The biggest challenge in these meetings was the research communication between the parties from the different areas: At the time, I became aware of the fact that academic groups develop their own particular science language, that is, for example, the "biology language" or the "computer science language,"[3] and that an English sentence is often interpreted very differently by different scientific communities. For this reason alone, to succeed[4] in interdisciplinary research, some of the necessary properties an interdisciplinary academic must possess are patience, the ability to question the (maybe existing) common grounds over and over again, and, therefore, one needs plenty of time to do well.

2 Parameterized Complexity and Interdisciplinary Research

A key during the problem solving process—when the goal is to design an algorithm or to provide even an implementation for an applied research problem—is

[2] Iris who had just finished her M.A. in Psychology was visiting Mike in Victoria during one of my later visits.

[3] Some sciences have many "dialects" as my collaboration with people from networks and software engineering has demonstrated to me.

[4] If you are bilingual, that is educated rigorously in both areas, then you can be a translator.

the ability to come up with a precise abstract problem description. Sometimes the algorithm designer may expect a formulation as a computational problem delivered by the user. However, formulating the problem can be challenging for the user. The algorithm designer must truly understand the problem at hand when making it abstract. Here, the translation between the different science languages is probably the most challenging part.

Even after a problem description is agreed upon, many times the first draft is just a stepping stone. To confirm that the right computational problem description has been found, often algorithmic results from real data inputs solving the problem are needed. If unsuccessful, the problem description must be revised. This process is part of the typical problem solving design cycle (such as the classic hypothetico-deductive method [83,84] used by scientists).

The task is not finished with the successful implementation of an algorithm. Practitioners typically rely on the integration of the implementation in a user-friendly and well-documented software package. Further, we want users to know what the advantages are of using optimal (or *exact*) algorithms compared to heuristics: for example, exact algorithms allow—in contrast to heuristics—an accurate evaluation of a computational problem when considered as a model of an applied research question. Therefore, our practical parameterized algorithms should be readily available for the user: we require user-friendly packages that combine parameterized and heuristic approaches and allow the user to opt for the exact approach whenever its running time is feasible. Efficient preprocessing in the form of data reduction should be applied as much as possible, even in cases where exact computation is impractical. Today, in many practical applications (such as sequence alignment problems), heuristic approaches are commonly applied to the underlying hard problems. One of the mission statements of applied parameterized complexity should be to offer additional exact approaches wherever possible as an alternative to heuristic approaches.

That parameterized algorithms can be practical is (still) best illustrated using the Vertex Cover problem: Vertex Cover instances containing a vertex cover of size up to at least $k = 200$ are considered practical. Best practices for implementations and use in bioinformatics for parameterized algorithms are the Vertex Cover ones implemented in Langston's and Dehne's groups [42,44,43,15,11].

While the FPT-toolkit is great for solving computational problems algorithmically, parameterized complexity may also inform science when studying computational models in the search for models of *cognitive capacities* and cognitive processes [76,75]. *The tractability design cycle* is suggested as method to support this process.

We now turn our attention to the art of parameterization. For a given NP-hard computational problem, what parameterizations should be studied? While some parameter choices may seem natural, other choices appear less obvious. Of course, often the application can shed light on natural parameterizations that are not obvious otherwise. When, for example, parameterizing Vertex Cover by the size of the cover to be determined (aka k-Vertex Cover, the natural parameterization of VERTEX COVER), the problem is in FPT, while when parameterizing by the number of vertices not to be included in the cover—aka k-INDEPENDENT SET, a parameterization of VERTEX COVER that is *dual* to k-Vertex Cover—the problem turns out to be complete for the class W[1] [18]. If we choose as the parameter a less obvious (but also dual) variant, namely the number of edges covered minus the number of vertices in the cover (aka p-PROFIT COVER [71]), then the problem is again in FPT.

In terms of the practicality of FPT algorithms, parameters should be small to achieve acceptable running times [54]: that is, to solve the problem's optimization version, the optimum value for the parameter, when solving the problem's optimization version, should be small.[5] Do the same limitations on the choice of parameters hold for all purposes of parameterized complexity? First, finding tractable parameterized algorithms for different parameterizations of a problem may improve practical running times [71]. Second, the application can shed light on the practicality of a parameter: If VERTEX COVER is used to model a conflict graph to clean a data set (c.f. [69]), then we expect k, the number of data points (vertices) to be removed from the data set (set of vertices of the graph), to be small as otherwise the data set can be considered as too noisy to be kept. We may want to use PROFIT COVER to verify that a data set is profitable (enough). Further, a complete parameterized analysis can be useful when for example evaluating computational models as models for cognitive theories [81,76]. When studying the cognitive limitation of humans in human problem solving, an exhaustive study of the complexity of possible parameterizations may prove informative. In particular, even tractable parameterized results that are considered trivial, such as parameterizations above or below the optimum solution [47], may be of interest when the goal is to evaluate possible cognitive theories as efficiently computable functions.

Niedermeier discusses the issue of how to parameterize in his book *Invitations to Fixed-Parameter Algorithms* from the perspective of an algorithm designer [54], and later distinguishes different ways of identifying parameters, namely solution quality driven ones (such as the size of a solution), and structural ones (parameterizing by distance from triviality, based on data analysis, by deconstructing hardness proofs, by dimensions,[6] and averaging out) [53]. Parameterizations that depend on the real data sets may be most effective when developing a data-driven algorithmic implementation for the user. However, this way of parameterizing involves strong interdisciplinary collaboration: in [22], Fellows *et al.* say

> "Identifying parameters relevant to real-world datasets is something of an art [53] and essential to the useful deployment of the multivariate outlook on NP-hard problems. In some sense, the search for relevant parameters brings this part of theoretical computer science to the fields of Heuristics and Algorithms Engineering and Artificial Intelligence."

[5] This discussion is closely related to the, in the early years discussed, *klam value* [18].
[6] e.g., dimensions of input objects.

Niedermeier observes that the identification of parameterizations based on data analysis is still underdeveloped [53]. This might be one of the strongest expectations that is expected from interdisciplinary parameterized complexity research. A rigorous understanding of the data sets will yield an improved data-driven algorithm design-process by combining theoretical approaches with data facts.

3 Parameterized Complexity and Its Contributions to Computational Biology

Computational biology has received considerable attention from parameterized complexity researchers who have attacked a number of computational problems from the area. Early work considered the Perfect Phylogeny problem [6], DNA Physical Mapping [26], and sequence or alignment problems [5]; variants of maximum agreement subtree problems were studied in [23,24]. Tree reconciliation problems for gene and species trees are studied in [25,68,69,35,3]. Over the years, many problems in computational biology have been considered in the parameterized framework. There is a number of PhD theses with significant contributions in this area, including [34,8,20,69,37,7,31,67,33,64], with Hallett's thesis being the first [34]. For a survey summarizing results on parameterized algorithms in phylogenetics we refer to the article by Gramm *et al.* [32]. Typical problem parameters in these problems include the number of input objects to be considered, the number of evolutionary events (following the parsimony assumption this number should be small), and combinations thereof.

Most parameterized complexity work in bioinformatics or computational biology is of theoretical nature. However, in addition to the practical work by the groups of Langston and Dehne mentioned above, Hüffner and his collaborators have done recent work in algorithmic engineering with focus on the bioinformatics area [38,36].

As discussed in Section 2, deep knowledge of the problem structure, the application domain and the properties of the real data sets are crucial for the design of algorithms and implementations driven by specific applications. Interdisciplinary collaboration and knowledge of both domains for all parties are useful if not necessary for success. It appears that most results in the intersection of parameterized complexity and computational biology are published in computer science conferences and journals, with some exceptions: examples are the work by Hallett *et al.* with their publication at the computational biology conference Recomb on identifying duplications and lateral gene transfers, van Brevern *et al.* with the recent article on motif search in *Transactions on Computational Biology and Bioinformatics* [4], and Hüffner *et al.* on clustering in *Biological Networks* [40]. The article *Developing Fixed-Parameter Algorithms to Solve Combinatorially Explosive Biological Problems* by Hüffner *et al.* published in *Bioinformatics* [39] is probably the one that best introduces the techniques of parameterized complexity to the biology community.

4 The Role of Parameterized Complexity in Cognitive Science

Human problem solving is a subarea of cognitive science that studies *human* problem solving in terms of problem solving strategies and performance, and looks for models of *cognitive capacities* (also denoted *computational-level theories* [48,76,75]) as well as cognitive processes. In the past, results from computational complexity have influenced this research,[7] and were used in particular to justify the rejection of computational problems as potential computational-level theories according to what van Rooij denoted the *P-cognition thesis* [75]: only computational problems that can be solved in polynomial time can serve as computational-level theories. That is, computational problems that are shown to be NP-hard were either rejected or inexact solution strategies were suggested as explanations for people dealing with the intractability of these problems (e.g., approximation algorithms or heuristic). Wareham was the first arguing in his 1996 paper that parameterized complexity is the better tool than classical computational complexity when measuring the complexity of computational models for cognitive systems or identifying the sources that isolate the model power [80]. In recent years, the consideration of parameterized complexity has lead to a relaxation of the P-cognition thesis, resulting in the *FPT-cognition thesis*: NP-hard but fixed-parameter tractable parameterized computational decision problems can also be considered candidates for computational-level theories [76,75].

Modelling Cognitive Capacities. A common assumption amongst a group of cognitive scientists is that cognitive capacities consist of input/output functions that are "efficiently computable" (cf. the works by van Rooi [76,75] as well as articles by Cherniak [14], Frixione [29], Levesque [45] and Tsotsos [73]). Further, it is widely assumed that computational complexity can aid cognitive science research [1,49,50,52,56,57,61,65,66,73]. As Tsotsos noted in his author's response to the commentaries of his article from 1990, complexity analysis is an important dimension of study when modelling vision [73].

Typical questions when investigating cognitive capacities and studying human problem solving include: What computational problems are good candidates for models of cognitive capacities? How do people solve these computational problems? What are people's limitations w.r.t. instance sizes and properties? What are plausible models for the human solution process?

Many computational problems that are discussed in the literature as candidates for models of cognitive capacities are characterized as NP-hard [66,73,59,46,80,50,81,30,72,76,79,78,77,82]. As a consequence, to serve as models for cognitive capacities, these problems are either disregarded in their general form, or they decoy the researcher to assume that people use approximation algorithms or heuristics to solve the tasks. Complexity analysis as such has lead to criticism of its relevance for this purpose (cf. commentaries to [73]). The use of parameterized complexity in this area has shed a different light on problems that are NP-hard but have parameterizations that are members of the

[7] The dissertations by Wareham and van Rooij are examples surveying this literature.

class FPT and thus put the plausibility of computational problems as cognitive models into a different perspective: since there exist exact algorithms that are tractable for some NP-hard problems, heuristics and approximation algorithms may not be the only possible explanation for approaches that humans choose when tackling certain problems.

Parameterized complexity was first suggested by Wareham as a tool to evaluate models of cognitive systems [80]. He argues that a refined analysis using parameterized complexity proves more useful than classical complexity analysis alone. To evaluate an NP-hard problem systematically as a model of a cognitive capacity—that is, to fully characterize a computational problem by showing what and what does not make it tractable—Wareham suggest to use what he calls *systematic parameterized complexity analysis* [81].

Some of the first discussed computational problems as models of cognitive capacities using parameterized complexity were phonological models in linguistics [80,81]—namely *Declarative Phonology* problems [63] and *Optimality Theory* [60]. Both are constraint-based theories: Optimality Theory has a priority order for constraint satisfaction while in Computational Declarative Phonology problems all constraints must be satisfied; described are rule based phonological mechanisms that manipulate the (mental or spoken) phonological representation. Wareham discusses two distinct processes for a *declarative phonology system* $S =< P, D >$, that is a pair of constraint sets where P encodes the phonological mechanisms of a language and D encodes the lexicon of the language: *encoding* with its corresponding computational problem DP-Encode and *decoding* with DP-Decode [81]. A systematic parameterized complexity analysis over a spectrum of input and solution driven parameterizations for both theories reveals several tractable parameterizations that may be investigated further as plausible models in linguistics [81].

Van Rooij *et al.* studied Subset Choice problems that model decision making tasks [76,78]. Subset choice problems can arise in a variety of settings: choice situations in everyday settings (e.g., when selecting toppings for a pizza) as well as highly specialized ones (e.g., prescribing medication). Different models of the task of choosing a subset of items from among a set of available items are investigated [76,78].

Most recent work includes the discussion of the complexity of self-organization of cognitive behaviour using Constraint Satisfaction by van Rooij [74], on Bayesian Intractability by Kwisthout *et al.* [41] and the human performance of *Vertex Cover* by Carruthers *et al.* [10].

5 Conclusions

Many hard problems in practical applications exhibit a rich structure that allows a number of realistic parameterizations that permit the design of parameterized algorithms. In bioinformatics, there is a tremendous need for developing exact algorithms for problems with very large data inputs. To move even more of the highly evolved algorithmic results from parameterized complexity in bioinformatics towards sophisticated practical implementations, the area will profit from: continued interdisciplinary collaboration, a push in parameterized algorithmic engineering, and a series of publications in the biology community to publicize the methods and enlighten an even larger number of researchers about the advantages of optimal algorithms compared to heuristics. To succeed, from the applied user's standpoint, more practical parameterized algorithms should be readily available.

The research findings in cognitive modeling have the potential to significantly impact the ability of cognitive psychologists to identify new theories that model cognitive capacities and problem solving processes. In contrast to the computational biology field, the majority of parameterized complexity research in cognitive science is published in cognitive science and psychology journals. In particular van Rooij has began to publicize findings in the cognitive science community.[8]

While parameterized complexity results in computational biology mainly focus on the development of fast algorithms, in cognitive psychology the main focus is on the modelling of cognitive capacities. Computational biology can learn from the the modeling research done in cognitive science to improve the process of formalizing of computational problems.

Acknowledgements. I want to thank Hausi Müller and my research group at UVic, in particular Sarah Carruthers, for lots of interesting discussions on the subject.

References

1. Anderson, J.R.: The Adaptive Character of Thought. Lawrence Erlbaum Associates (1990)
2. Balasubramanian, R., Fellows, M.R., Raman, V.: An Improved Fixed-Parameter Algorithm for Vertex Cover. Information Processing Letters 65, 163–168 (1998)
3. Bansal, M.S., Shamir, R.: A Note on the Fixed Parameter Tractability of the Gene-Duplication Problem. IEEE/ACM Trans. Comput. Biol. Bioinformatics 8(3), 848–850 (2011)
4. Betzler, N., van Bevern, R., Komusiewicz, C., Fellows, M.R., Niedermeier, R.: Parameterized Algorithmics for Finding Connected Motifs in Biological Networks. IEEE/ACM Transactions on Computational Biology and Bioinformatics 8(5), 1296–1308 (2011)
5. Bodlaender, H.L., Downey, R.G., Fellows, M.R., Hallett, M.T., Wareham, H.T.: Parameterized Complexity Analysis in Computational Biology. Computer Applications in the Biosciences 11(2), 49–57 (1995)
6. Bodlaender, H.L., Fellows, M.R., Warnow, T.J.: Two Strikes Against Perfect Phylogeny. In: Kuich, W. (ed.) ICALP 1992. LNCS, vol. 623, pp. 273–283. Springer, Heidelberg (1992)
7. Boucher, C.A.: Combinatorial and Probabilistic Approaches to Motif Recognition. PhD thesis, University of Waterloo (2010)

[8] Parameterized complexity was used by several participants in their presentations at the recent interdisciplinary Seminar *Computer Science & Problem Solving: New Foundations* that took place Dagstuhl in 2011.

8. Bryant, D.: Building Trees, Hunting for Trees, and Comparing Trees—Theory and Methods in Phylogenetic Analysis. PhD thesis, University of Canterbury (1997)
9. Buss, J.F., Goldsmith, J.: Nondeterminism within P. SIAM J. Comput. 22(3), 560–572 (1993)
10. Carruthers, S., Masson, M., Stege, U.: Human Performance on Hard Non-Euclidean Graph Problems: Vertex Cover. Accepted to the Journal of Problem Solving (2012)
11. Cheetham, J., Dehne, F., Rau-Chaplin, A., Stege, U., Taillon, P.J.: Solving Large FPT Problems on Coarse Grained Parallel Machines. Journal of Computer and System Sciences 67(4), 691–706 (2003)
12. Chen, J., Kanj, I.A., Jia, W.: Vertex Cover: Further Observations and Further Improvements. In: Widmayer, P., Neyer, G., Eidenbenz, S. (eds.) WG 1999. LNCS, vol. 1665, pp. 313–324. Springer, Heidelberg (1999)
13. Chen, J., Kanj, I.A., Xia, G.: Improved Parameterized Upper Bounds for Vertex Cover. In: Královič, R., Urzyczyn, P. (eds.) MFCS 2006. LNCS, vol. 4162, pp. 238–249. Springer, Heidelberg (2006)
14. Cherniak, C.: Minimal Rationality. MIT Press (1986)
15. ClustalXP, `http://clustalxp.cgmlab.org/` (retrieved 2012)
16. Day, W.H.E., Sankoff, D.: Computational Complexity of Inferring Phylogenies by Compatibility. Syst. Zool. 35(2), 224–229 (1986)
17. Downey, R.G., Fellows, M.R.: Fixed parameter tractability and completeness II: Completeness for W[1]. Theoretical Computer Science A 141, 109–131 (1995)
18. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Springer (1999)
19. Downey, R.G., Fellows, M.R., Stege, U.: Parameterized Complexity: a Framework for Systematically Confronting Computational Intractability. In: Graham, R., Kratochvil, J., Nesetril, J., Roberts, F. (eds.) Proc. DIMACS-DIMATIA Workshop, Prague. Contemporary Trends in Discrete Mathematics (1997), AMS-DIMACS Series in Discrete Mathematics and Theoretical Computer Science 49, 49–99 (1999)
20. Evans, P.: Algorithms and Complexity for Annotated Sequence Analysis. PhD thesis, University of Victoria (1999)
21. Fellows, M.R.: On the Complexity of Vertex Set Problems. Technical Report, Computer Science Department, University of New Mexico (1988)
22. Fellows, M.R., Gaspers, S., Rosamond, F.: Multivariate Complexity Theory. In: Blum, E.K., Aho, A.V. (eds.) Computer Science, pp. 269–293. Springer, New York (2011)
23. Fellows, M.R., Hallett, M.T., Korostensky, C., Stege, U.: Analogs and Duals of the MAST Problem for Sequences and Trees. In: Bilardi, G., Pietracaprina, A., Italiano, G.F., Pucci, G. (eds.) ESA 1998. LNCS, vol. 1461, pp. 103–114. Springer, Heidelberg (1998)
24. Fellows, M.R., Hallett, M.T., Stege, U.: Analogs and Duals of the MAST Problem for Sequences and Trees. Journal of Algorithms 49(1), 192–216 (2003)
25. Fellows, M.R., Hallett, M.T., Stege, U.: On the Multiple Gene Duplication Problem. In: Chwa, K.-Y., Ibarra, O.H. (eds.) ISAAC 1998. LNCS, vol. 1533, pp. 347–357. Springer, Heidelberg (1998)
26. Fellows, M.R., Hallett, M.T., Wareham, H.T.: DNA Physical Mapping: Three Ways Difficult. In: Lengauer, T. (ed.) ESA 1993. LNCS, vol. 726, pp. 157–168. Springer, Heidelberg (1993)
27. Fellows, M.R., Langston, M.: Nonconstructive Advances in Polynomial Time Complexity. Information Processing Letters 26, 157–162 (1987)

28. Fellows, M.R., McCartin, C., Rosamond, F.A., Stege, U.: Coordinatized kernels and Coordinatized Kernels and Catalytic Reductions: An Improved FPT Algorithm for Max Leaf Spanning Tree and Other Problems. In: Kapoor, S., Prasad, S. (eds.) FST TCS 2000. LNCS, vol. 1974, pp. 240–251. Springer, Heidelberg (2000)
29. Frixione, M.: Tractable competence. Minds and Machines 11, 379–397 (2001)
30. Graham, S.M., Joshi, A., Pizlo, Z.: The Traveling Salesman Problem: A Hierarchical Model. Memory & Cognition 28(7), 1191–1204 (2000)
31. Gramm, J.: Fixed-Parameter Algorithms for the Consensus Analysis of Genomic Data. PhD thesis, Universität Tübingen (2003)
32. Gramm, J., Nickelsen, A., Tantau, T.: Fixed-Parameter Algorithms in Phylogenetics. The Computer Journal 51(1), 79–101 (2008)
33. Guo, J.: Algorithm Design Techniques for Parameterized Graph Modification Problems. PhD thesis, Friedrich-Schiller-Universität Jena (2006)
34. Hallett, M.T.: An Integrated Complexity Analysis of Problems from Computational Biology. PhD thesis, University of Victoria (1996)
35. Hallett, M., Lagergren, J., Tofigh, A.: Simultaneous Identification of Duplications and Lateral Transfers. In: RECOMB 2004, pp. 47–356 (2004)
36. Helwig, S., Hüffner, F., Rössling, I., Weinard, M.: Chapter 3. Selected Design Issues. In: Müller-Hannemann, M., Schirra, S. (eds.) Algorithm Engineering. LNCS, vol. 5971, pp. 58–126. Springer, Heidelberg (2010)
37. Hermelin, D.: New Results in Parameterized Complexity. PhD thesis, University of Haifa (2009)
38. Hüffner, F.: Algorithm Engineering for Optimal Graph Bipartization. Journal of Graph Algorithms and Applications 13(2), 77–98 (2009)
39. Hüffner, F., Niedermeier, R., Wernicke, S.: Developing Fixed-Parameter Algorithms to Solve Combinatorially Explosive Biological Problems. Bioinformatics, 395–421 (2007)
40. Hüffner, F., Niedermeier, R., Wernicke, S.: Fixed-parameter algorithms for graph-modeled data clustering. In: Clustering Challenges in Biological Networks, pp. 3–28. World Scientific (2009)
41. Kwisthout, J., Wareham, T., van Rooij, I.: Bayesian Intractability is not an Ailment that Approximation Can Cure. Cognitive Science 35(5), 779–784 (2011)
42. Langston, M.A.: Homepage, `http://web.eecs.utk.edu/~langston/` (retrieved 2012)
43. Langston, M.A., Abu-Khzam, F.N., Collins, R.L., Fellows, M.R., Suters, W.H., Symons, C.T.: Kernelization Algorithms for the Vertex Cover Problem: Theory and Experiments. In: ALENEX 2004, pp. 62–69 (2004)
44. Langston, M.A., Abu-Khzam, F.N., Shanbhag, P.: Scalable Parallel Algorithms for Difficult Combinatorial Problems: a Case Study in Optimization. In: PDCS 2003, pp. 649–654 (2003)
45. Levesque, H.J.: Logic and the complexity of reasoning. Journal of Philosophical Logic 17, 355–389 (1988)
46. MacGregor, J.N., Ormerod, T.C., Chronicle, E.P.: A Model of Human Performance on the Traveling Salesperson Problem. Memory & Cognition 28(7), 1183–1190 (2000)
47. Mahajan, M., Raman, V., Sikdar, S.: Parameterizing Above or Below Guaranteed Values. J. Comput. Syst. Sci. 75(2), 137–153 (2009)
48. Marr, D.: Vision: A Computational Investigation into the Human Representation and Processing Visual Information. W.H. Freeman (1982)
49. Martignon, L., Hoffrage, U.: Fast, Frugal, and Fit: Simple Heuristics for Paired Comparison. Theory and Decision 52, 29–71 (2002)

50. Martignon, L., Schmitt, M.: Simplicity and Robustness of Fast and Frugal Heuristics. Minds and Machines 9, 565–593 (1999)
51. Mehlhorn, K.: Data Structures and Efficient Algorithms. Graph Algorithms and NP-Completeness, vol. 2. Springer (1984)
52. Millgram, E.: Coherence: The Price of the Ticket. Journal of Philosophy 97, 82–93 (2000)
53. Niedermeier, R.: Reflections on Multivariate Algorithmics and Problem Parameterization. In: STACS 2010, pp. 17–32 (2010)
54. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press (2006)
55. Niedermeier, R., Rossmanith, P.: Upper Bounds for Vertex Cover Further Improved. In: Meinel, C., Tison, S. (eds.) STACS 1999. LNCS, vol. 1563, pp. 561–570. Springer, Heidelberg (1999)
56. Oaksford, M., Chater, N.: Reasoning Theories and Bounded Tationality. In: Manktelow, K.I., Over, D.E. (eds.) Rationality: Psychological and Philosophical Perspectives, pp. 31–60 (1998)
57. Oaksford, M., Chater, N.: Rationality in an uncertain world: Essays on the cognitive science of human reasoning. Psychology Press, Hove (1993)
58. Papadimitriou, C.H., Yannakakis, M.: On limited nondeterminism and the complexity of the V-C dimension. Journal of Computer and System Sciences 53(2), 161–170 (1996)
59. Parberry, I.: Knowledge, Understanding, and Computational Complexity. In: Levine, D.S., Elsberry, W.R. (eds.) Optimality in Biological and Artificial Networks?, pp. 125–144. Lawrence Erlbaum Publishers, Hillsdale (1997)
60. Prince, A., Smolensky, P.: Optimality Theory Constraint Interaction in Generative Grammar. Tech. Rep. RuCCS TR-2, Center for Cognitive Science, Rutgers University (1993)
61. Rensink, R.A., Provan, G.: The Analysis of Resource-Limited Vision Systems. In: Proceedings of the 13th Annual Conference of the Cognitive Science Society, pp. 311–316 (1991)
62. Roth-Korostensky, C.: Algorithms for Building Multiple Sequence Alignments and Evolutionary Trees. PhD thesis, ETH Zürich (2000)
63. Scobbie, J.M.: Towards Declarative Phonology. In: Bird, S. (ed.) Declarative Perspectives in Phonology. Edinburgh Working Papers in Cognitive Science, vol. 7, pp. 1–27. University of Ediburgh (1992)
64. Shaw, P.: Advances in Cluster Editing: Linear FPT Kernels and Comparative Implementations. PhD thesis, The University of Newcastle (2010)
65. Simon, H.A.: Rationality as Process and as Product of Thought. In: Bell, D.E., Raiffa, H., Tversky, A. (eds.) Decision Making: Descriptive, Normative, and Prescriptive Interactions, pp. 58–77. Cambridge University Press, Cambridge (1988)
66. Simon, H.A.: Invariants of human behavior. Annual Review of Psychology 41(1), 1–19 (1990)
67. Snir, S.: Computational Issues in Phylogenetic Reconstruction: Analytic Maximum Likelihood Solutions, and Convex Recoloring. PhD thesis, Technion (2004)
68. Stege, U.: Gene Trees and Species Trees: The Gene-Duplication Problem is Fixed-Parameter Tractable. In: Dehne, F., Gupta, A., Sack, J.-R., Tamassia, R. (eds.) WADS 1999. LNCS, vol. 1663, pp. 288–293. Springer, Heidelberg (1999)
69. Stege, U.: Resolving Conflicts from Computational Biology. PhD thesis, ETH Zürich (2000)

70. Stege, U., Fellows, M.R.: An Improved Fixed-Parameter Tractable Algorithm for Vertex Cover. Technical Report 318, Department of Computer Science, ETH Zürich (April 1999)
71. Stege, U., van Rooij, I., Hertel, A., Hertel, P.: An $O(pn + 1.151^p)$-Algorithm for p-Profit Cover and Its Practical Implications for Vertex Cover. In: Bose, P., Morin, P. (eds.) ISAAC 2002. LNCS, vol. 2518, pp. 249–261. Springer, Heidelberg (2002)
72. Thagard, P.: Coherence in Thought and Action. MIT Press (2000)
73. Tsotsos, J.K.: Analyzing Vision at the Complexity Level. Behavioral and Brain Sciences 13(3), 423–469 (1990)
74. Rooij, I.: Self-Organization Takes Time too. Topics in Cognitive Science 4, 63–71 (2012)
75. van Rooij, I.: The tractable Cognition Thesis. Cognitive Science (2008)
76. van Rooij, I.: Tractable Cognition: Complexity Theory in Cognitive Psychology. Ph.D. Thesis, Department of Psychology, University of Victoria (2003)
77. van Rooij, I., Schactman, A., Kadlec, H., Stege, U.: Perceptual or Analytical Processing? Evidence from Children's and Adult's Performance on the Euclidean Traveling Salesman Problem. Journal of Problem Solving 1(1), 44–73 (2006)
78. van Rooij, I., Stege, U., Kadlec, H.: Sources of Complexity in Subset Choice. Journal of Mathematical Psychology 49(2), 160–187 (2005)
79. van Rooij, I., Stege, U., Schactman: Convex hull and Tour Crossings in the Euclidean Traveling Salesperson Problem: Implications for Human Performance Studies. Memory & Cognition 31(2), 215–220 (2003)
80. Wareham, H.T.: The role of Parameterized Computational Complexity Theory in Cognitive Modeling. In: AAAI 1996 Workshop Working Notes: Computational Cognitive Modeling: Source of the Power (1996)
81. Wareham, H.T.: Systematic Parameterized Complexity Analysis in Computational Phonology. Ph.D. Thesis, Department of Computer Science, University of Victoria (1999)
82. Wareham, H.T., Evans, P., van Rooij, I.: What does (and doesn't) make analogical problem solving easy? Journal of Problem Solving 3(2), 30–71 (2011)
83. Whewell, W.: History of the Inductive Sciences, from the Earliest to the Present Times, London, vol. 3 (1837)
84. Whewell, W.: The Philosophy of the Inductive Sciences, founded upon their history, London, vol. 2 (1840)