Capturing Structure in Hard Combinatorial Problems

(Invited Keynote Talk)

Stefan Szeider,
Vienna University of Technology, Vienna, Austria
www.szeider.net

For many hard combinatorial problems that arise from real-world applications, the conventional theory of algorithms and complexity cannot give reasonable (i.e., poly-time) performance guarantees and considers such problems as *intractable*. Nevertheless, heuristics-based algorithms and solvers work surprisingly well on real-world instances, which suggests that

> our world may be "friendly enough" to make many typical computational tasks poly-time—challenging the value of the conventional worst-case complexity view in CS. *Bart Selman* [1]

Indeed, there is an enormous gap between theoretical performance guarantees and the empirically observed performance of solvers. Efficient solvers exploit the "hidden structure" of real-world problems, and so a theoretical framework that explains practical problem hardness and easiness must not ignore such structural aspects.

Parameterized Complexity [2,3,4] is a rigorous theoretical framework that offers a great potential for reducing the theory-practice gap. The key idea is to consider—in addition to the input size in bits—a secondary dimension, the parameter, and to design and analyse algorithms in this two-dimensional setting. Virtually in every conceivable context we know more about the input data than just its size. The second dimension (the parameter) can represent this additional information and capture some of the hidden structure present in problem instances. This two-dimensional setting gives raise to a foundational theory of algorithms and complexity that can be closer to the problems as they appear in the real world. *Fixed-parameter tractability (FPT)*, a fundamental concept of parameterized complexity, refers to solubility in time $f(k)n^c$ where f is some (possibly exponential) function of the parameter k, n is the size of the input, and c is a constant independent from k. It is important to observe that algorithms with such a running time scale significantly better in the parameter k than algorithms with running times of, say, n^k. Parameterized complexity has been introduced and pioneered by Downey and Fellows [2] and is receiving growing interest as reflected by thousands of research papers[1].

There are many ways of choosing parameters, and different parameters might be preferred for problem instances form different problem domains. Often the specific problem at hand offers a suitable parameter. For instance, the problem of extracting a small unsatisfiable subset from a set of clauses is naturally parameterized by the size of the subset [5], the MAX-SAT problem is naturally parameterized by the number of satisfied clauses beyond a guaranteed value [6], the local consistency checking problem for constraint networks is naturally parameterized by the level of local consistency (k-consistency) [7], and the problem of reusing a plan is naturally parameterized by the edit distance between the old and the new plan [8].

On the other hand, there are several generic approaches to parameterize a problem, two particularly versatile approaches are based on decompositions and backdoors.

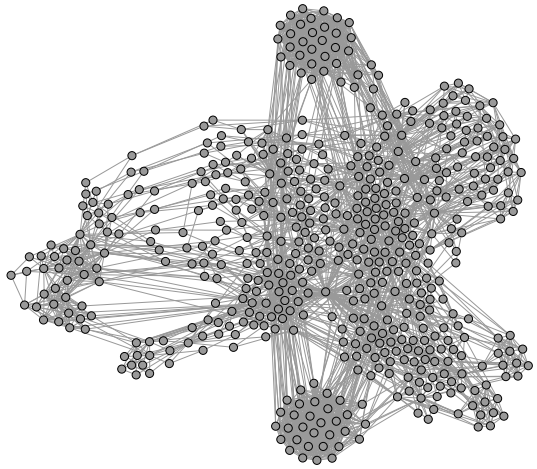

Figure 1. Real-world SAT instance.

Decomposition width captures the common approach to solve a hard problem by decomposing the instance into small tractable parts, and to reassemble the solutions of the parts to a solution of the entire instance. One aims at decompositions for which the overall complexity depends on how much the parts overlap, the "width" of the decomposition, which also provides the parameter for a parameterized complexity analysis. Our recent work in this direction covers problems arising in constraint satisfaction [9], propositional model counting [10,11], answer-set programming [12], argumentation [13], and machine learning [14].

If a computational problem is intractable in general, it is natural to ask for subproblems for which the problem is

[1]Google Scholar lists over 5000 results on the query "parameterized complexity" OR "fixed parameter tractable" (retrieved September 20, 2013).

solvable in poly-time, and indeed much research has been devoted to this question. Such tractable subproblems are sometimes called "islands of tractability." The concept of *backdoors*, introduced by Williams, Gomes, and Selman [15] offers a generic way to gradually enlarge and extend an island of tractability and thus to solve problem instances efficiently if they are close to an island of tractability. The size of a smallest backdoor indicates the distance between an instance and the island and is a natural complexity parameter. Real-world SAT instances often contain small backdoors [16]. Our recent survey [17] describes several parameterized complexity results on backdoors for propositional satisfiability (SAT).

Many important problems in AI and Reasoning are "harder" than SAT and are located beyond NP in the Polynomial Hierarchy. In order to render such problems fixed-parameter tractable, one needs to choose parameters that are quite restrictive. Therefore, for such problems it seems to be an even more interesting approach to exploit hidden structure not to *solve* the instance, but to *reduce* it. The hidden structure captured by a parameter can be used to translate the instance to an output-equivalent instance of a different problem of lower complexity. The parameter can thus be less restrictive and can be reasonably small for larger classes of inputs. Clearly such a reduction cannot run in poly-time, unless the Polynomial Hierarchy collapses, but the enhanced power of FPT-reductions can *break complexity barriers* between conventional complexity classes. SAT is well-suited as a target problem since by means of FPT-reductions to SAT we can make today's extremely powerful SAT solvers applicable to problems at higher levels of the Polynomial Hierarchy. In fact, there are some extremely successful known techniques such as Bounded Model Checking [18] that, in retrospect, can be seen as FPT-reductions to SAT. In recent work [19,20] we have developed FPT-reductions that break complexity barriers for the main reasoning problems of disjunctive answer-set programming and problems arising in propositional abductive reasoning, both located at the second level of the Polynomial Hierarchy.

Acknowledgment: The author gratefully acknowledges the support by the European Research Council (ERC), project COMPLEX REASON 239962.

References

[1] B. Selman, "Solving large-scale computational problems using insights from statistical physics," Workshop on Statistical Mechanics of Unsatisfiability and Glasses, Mariehamn, Finland, 23–26 May 2012, 2012.

[2] R. G. Downey and M. R. Fellows, *Parameterized Complexity*. New York: Springer Verlag, 1999.

[3] R. Niedermeier, *Invitation to Fixed-Parameter Algorithms*. Oxford: Oxford University Press, 2006.

[4] J. Flum and M. Grohe, *Parameterized Complexity Theory*. Berlin: Springer Verlag, 2006, vol. XIV.

[5] M. R. Fellows, S. Szeider, and G. Wrightson, "On finding short resolution refutations and small unsatisfiable subsets," *Theoretical Computer Science*, vol. 351, no. 3, pp. 351–359, 2006.

[6] N. Alon, G. Gutin, E. J. Kim, S. Szeider, and A. Yeo, "Solving MAX-r-SAT above a tight lower bound," *Algorithmica*, vol. 61, no. 3, pp. 638–655, 2011.

[7] S. Gaspers and S. Szeider, "The parameterized complexity of local consistency," in *Proceedings of CP 2011*, ser. Lecture Notes in Computer Science, J. H.-M. Lee, Ed., vol. 6876. Springer Verlag, 2011, pp. 302–316.

[8] R. de Haan, A. Roubickova, and S. Szeider, "Parameterized complexity results for plan reuse," in *Proceedings of AAAI 2013*, M. des Jardins and M. L. Littman, Eds. The AAAI Press, 2013, pp. 224–231.

[9] M. Samer and S. Szeider, "Constraint satisfaction with bounded treewidth revisited," *J. of Computer and System Sciences*, vol. 76, no. 2, pp. 103–114, 2010.

[10] ——, "Algorithms for propositional model counting," *J. Discrete Algorithms*, vol. 8, no. 1, pp. 50–64, 2010.

[11] S. Ordyniak, D. Paulusma, and S. Szeider, "Satisfiability of acyclic and almost acyclic CNF formulas," *Theoretical Computer Science*, vol. 481, pp. 85–99, 2013.

[12] R. Pichler, S. Rümmele, S. Szeider, and S. Woltran, "Tractable answer-set programming with weight constraints: Bounded treewidth is not enough," in *Proceedings of KR 2010*, F. Lin, U. Sattler, and M. Truszczynski, Eds. AAAI Press, 2010.

[13] E. J. Kim, S. Ordyniak, and S. Szeider, "Algorithms and complexity results for persuasive argumentation," *Artificial Intelligence*, vol. 175, pp. 1722–1736, 2011.

[14] S. Ordyniak and S. Szeider, "Parameterized complexity results for exact Bayesian network structure learning," *J. Artif. Intell. Res.*, vol. 46, pp. 263–302, 2013.

[15] R. Williams, C. Gomes, and B. Selman, "Backdoors to typical case complexity," in *Proceedings of IJCAI 2003*, G. Gottlob and T. Walsh, Eds. Morgan Kaufmann, 2003, pp. 1173–1178.

[16] Z. Li and P. van Beek, "Finding small backdoors in SAT instances," in *Proceedings of the 24th Can. Conf. on A. I. 2011*, ser. Lecture Notes in Computer Science, C. J. Butz and P. Lingras, Eds., vol. 6657. Springer Verlag, 2011, pp. 269–280.

[17] S. Gaspers and S. Szeider, "Backdoors to satisfaction," in *The Multivariate Algorithmic Revolution and Beyond - Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, ser. Lecture Notes in Computer Science, H. L. Bodlaender, R. Downey, F. V. Fomin, and D. Marx, Eds., vol. 7370. Springer Verlag, 2012, pp. 287–317.

[18] A. Biere, A. Cimatti, E. M. Clarke, and Y. Zhu, "Symbolic model checking without bdds," in *Proceedings of TACAS 1999*, ser. Lecture Notes in Computer Science, R. Cleaveland, Ed., vol. 1579. Springer Verlag, 1999, pp. 193–207.

[19] J. K. Fichte and S. Szeider, "Backdoors to normality for disjunctive logic programs," in *Proceedings of AAAI 2013*, M. des Jardins and M. L. Littman, Eds. The AAAI Press, 2013, pp. 320–337.

[20] A. Pfandler, S. Rümmele, and S. Szeider, "Backdoors to abuction," in *Proceedings of IJCAI 2013*, 2013.