Beyond Satisfaction

Towards an Understanding of Real-World Efficient Computation

Jakob Nordström

Theoretical Computer Science Group
KTH EECS

SRA ICT TNG Research Challenge Day
May 3, 2018

This Is Me...

Jakob Nordström

Associate Professor

Theoretical Computer Science Group

School of Electrical Engineering and Computer Science

www.csc.kth.se/∼jakobn

jakobn@kth.se

. . . And This Is My Research Challenge

$(x_{1,1} \lor x_{1,2} \lor x_{1,3} \lor x_{1,4} \lor x_{1,5} \lor x_{1,6} \lor x_{1,7}) \land (x_{2,1} \lor x_{2,2} \lor x_{2,3} \lor x_{2,4} \lor x_{2,5} \lor x_{2,6} \lor x_{2,7}) \land (x_{3,1} \lor x_{3,2} \lor x_{3,3} \lor x_{3,4} \lor x_{3,5} \lor x_{3,6} \lor x_{3,7}) \land (x_{4,1} \lor x_{4,2} \lor x_{4,3} \lor x_{4,4} \lor x_{4,5} \lor x_{4,6} \lor x_{4,7}) \land (x_{5,1} \lor x_{5,2} \lor x_{5,3} \lor x_{5,4} \lor x_{5,5} \lor x_{5,6} \lor x_{5,7}) \land (x_{6,1} \lor x_{6,2} \lor x_{6,3} \lor x_{6,4} \lor x_{6,5} \lor x_{6,6} \lor x_{6,7}) \land (x_{7,1} \lor x_{7,2} \lor x_{7,3} \lor x_{7,4} \lor x_{7,5} \lor x_{7,6} \lor x_{7,7}) \land (x_{8,1} \lor x_{8,2} \lor x_{8,3} \lor x_{8,4} \lor x_{8,5} \lor x_{8,6} \lor x_{8,7}) \land (\overline{x}_{1,1} \lor \overline{x}_{2,1}) \land (\overline{x}_{1,1} \lor \overline{x}_{3,1}) \land (\overline{x}_{1,1} \lor \overline{x}_{4,1}) \land (\overline{x}_{1,1} \lor \overline{x}_{5,1}) \land (\overline{x}_{1,1} \lor \overline{x}_{6,1}) \land (\overline{x}_{1,1} \lor \overline{x}_{7,1}) \land (\overline{x}_{1,1} \lor \overline{x}_{8,1}) \land (\overline{x}_{2,1} \lor \overline{x}_{3,1}) \land (\overline{x}_{2,1} \lor \overline{x}_{4,1}) \land (\overline{x}_{2,1} \lor \overline{x}_{5,1}) \land (\overline{x}_{2,1} \lor \overline{x}_{6,1}) \land (\overline{x}_{2,1} \lor \overline{x}_{7,1}) \land (\overline{x}_{2,1} \lor \overline{x}_{8,1}) \land (\overline{x}_{3,1} \lor \overline{x}_{4,1}) \land (\overline{x}_{3,1} \lor \overline{x}_{5,1}) \land (\overline{x}_{3,1} \lor \overline{x}_{6,1}) \land (\overline{x}_{3,1} \lor \overline{x}_{7,1}) \land (\overline{x}_{3,1} \lor \overline{x}_{8,1}) \land (\overline{x}_{4,1} \lor \overline{x}_{5,1}) \land (\overline{x}_{4,1} \lor \overline{x}_{6,1}) \land (\overline{x}_{4,1} \lor \overline{x}_{7,1}) \land (\overline{x}_{4,1} \lor \overline{x}_{8,1}) \land (\overline{x}_{5,1} \lor \overline{x}_{6,1}) \land (\overline{x}_{5,1} \lor \overline{x}_{7,1}) \land (\overline{x}_{5,1} \lor \overline{x}_{8,1}) \land (\overline{x}_{6,1} \lor \overline{x}_{7,1}) \land (\overline{x}_{6,1} \lor \overline{x}_{8,1}) \land (\overline{x}_{7,1} \lor \overline{x}_{8,1}) \land (\overline{x}_{1,2} \lor \overline{x}_{2,2}) \land (\overline{x}_{1,2} \lor \overline{x}_{3,2}) \land (\overline{x}_{1,2} \lor \overline{x}_{4,2}) \land (\overline{x}_{1,2} \lor \overline{x}_{5,2}) \land (\overline{x}_{1,2} \lor \overline{x}_{6,2}) \land (\overline{x}_{1,2} \lor \overline{x}_{7,2}) \land (\overline{x}_{1,2} \lor \overline{x}_{8,2}) \land (\overline{x}_{2,2} \lor \overline{x}_{3,2}) \land (\overline{x}_{2,2} \lor \overline{x}_{4,2}) \land (\overline{x}_{2,2} \lor \overline{x}_{5,2}) \land (\overline{x}_{2,2} \lor \overline{x}_{6,2}) \land (\overline{x}_{2,2} \lor \overline{x}_{7,2}) \land (\overline{x}_{2,2} \lor \overline{x}_{8,2}) \land (\overline{x}_{3,2} \lor \overline{x}_{4,2}) \land (\overline{x}_{3,2} \lor \overline{x}_{5,2}) \land (\overline{x}_{3,2} \lor \overline{x}_{6,2}) \land (\overline{x}_{3,2} \lor \overline{x}_{7,2}) \land (\overline{x}_{3,2} \lor \overline{x}_{8,2}) \land (\overline{x}_{4,2} \lor \overline{x}_{5,2}) \land (\overline{x}_{4,2} \lor \overline{x}_{6,2}) \land (\overline{x}_{4,2} \lor \overline{x}_{7,2}) \land (\overline{x}_{4,2} \lor \overline{x}_{8,2}) \land (\overline{x}_{5,2} \lor \overline{x}_{6,2}) \land (\overline{x}_{5,2} \lor \overline{x}_{7,2}) \land (\overline{x}_{5,2} \lor \overline{x}_{8,2}) \land (\overline{x}_{6,2} \lor \overline{x}_{7,2}) \land (\overline{x}_{6,2} \lor \overline{x}_{8,2}) \land (\overline{x}_{7,2} \lor \overline{x}_{8,2}) \land (\overline{x}_{1,3} \lor \overline{x}_{2,3}) \land (\overline{x}_{1,3} \lor \overline{x}_{3,3}) \land (\overline{x}_{1,3} \lor \overline{x}_{4,3}) \land (\overline{x}_{1,3} \lor \overline{x}_{5,3}) \land (\overline{x}_{1,3} \lor \overline{x}_{6,3}) \land (\overline{x}_{1,3} \lor \overline{x}_{7,3}) \land (\overline{x}_{1,3} \lor \overline{x}_{8,3}) \land (\overline{x}_{2,3} \lor \overline{x}_{3,3}) \land (\overline{x}_{2,3} \lor \overline{x}_{4,3}) \land (\overline{x}_{2,3} \lor \overline{x}_{5,3}) \land (\overline{x}_{2,3} \lor \overline{x}_{6,3}) \land (\overline{x}_{2,3} \lor \overline{x}_{7,3}) \land (\overline{x}_{2,3} \lor \overline{x}_{8,3}) \land (\overline{x}_{3,3} \lor \overline{x}_{4,3}) \land (\overline{x}_{3,3} \lor \overline{x}_{5,3}) \land (\overline{x}_{3,3} \lor \overline{x}_{6,3}) \land (\overline{x}_{3,3} \lor \overline{x}_{7,3}) \land (\overline{x}_{3,3} \lor \overline{x}_{8,3}) \land (\overline{x}_{4,3} \lor \overline{x}_{5,3}) \land (\overline{x}_{4,3} \lor \overline{x}_{6,3}) \land (\overline{x}_{4,3} \lor \overline{x}_{7,3}) \land (\overline{x}_{4,3} \lor \overline{x}_{8,3}) \land (\overline{x}_{5,3} \lor \overline{x}_{6,3}) \land (\overline{x}_{5,3} \lor \overline{x}_{7,3}) \land (\overline{x}_{5,3} \lor \overline{x}_{8,3}) \land (\overline{x}_{6,3} \lor \overline{x}_{7,3}) \land (\overline{x}_{6,3} \lor \overline{x}_{8,3}) \land (\overline{x}_{7,3} \lor \overline{x}_{8,3}) \land (\overline{x}_{1,4} \lor \overline{x}_{2,4}) \land (\overline{x}_{1,4} \lor \overline{x}_{3,4}) \land (\overline{x}_{1,4} \lor \overline{x}_{4,4}) \land (\overline{x}_{1,4} \lor \overline{x}_{5,4}) \land (\overline{x}_{1,4} \lor \overline{x}_{6,4}) \land (\overline{x}_{1,4} \lor \overline{x}_{7,4}) \land (\overline{x}_{1,4} \lor \overline{x}_{8,4}) \land (\overline{x}_{2,4} \lor \overline{x}_{3,4}) \land (\overline{x}_{2,4} \lor \overline{x}_{4,4}) \land (\overline{x}_{2,4} \lor \overline{x}_{5,4})$

Three Problems. . .

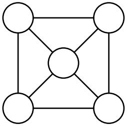
Colouring

Does graph $G = (V, E)$ have a colouring with k colours so that neighbours have distinct colours?

Three Problems...

Colouring

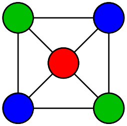
Does graph $G = (V, E)$ have a colouring with k colours so that neighbours have distinct colours?



Three Problems...

Colouring

Does graph $G = (V, E)$ have a colouring with k colours so that neighbours have distinct colours?

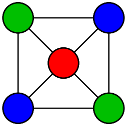


3-colouring exists

Three Problems...

Colouring

Does graph $G = (V, E)$ have a colouring with k colours so that neighbours have distinct colours?



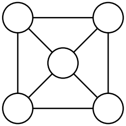
3-colouring exists but no 2-colouring

Three Problems...

Clique

Is there a clique in graph $G = (V, E)$ with k vertices that are all pairwise connected?

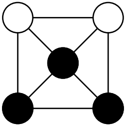
Three Problems...



Clique

Is there a clique in graph $G = (V, E)$ with k vertices that are all pairwise connected?

Three Problems...

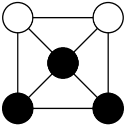


3-clique exists

CLIQUE

Is there a clique in graph $G = (V, E)$ with k vertices that are all pairwise connected?

Three Problems...



3-clique exists but no 4-clique

Clique

Is there a clique in graph $G = (V, E)$ with k vertices that are all pairwise connected?

Three Problems...

Colouring

Does graph $G = (V, E)$ have a colouring with k colours so that neighbours have distinct colours?

Clique

Is there a clique in graph $G = (V, E)$ with k vertices that are all pairwise connected?

Sat

Given propositional logic formula F, is there a satisfying assignment?

Three Problems...

Colouring

Does graph $G = (V, E)$ have a colouring with k colours so that neighbours have distinct colours?

Clique

Is there a clique in graph $G = (V, E)$ with k vertices that are all pairwise connected?

Sat

Given propositional logic formula F, is there a satisfying assignment?

Colouring:	frequency allocation for mobile base stations
Clique:	bioinformatics, computational chemistry
Sat:	easily models these and many other problems

. . . That Are Impossible to Solve in Theory. . .

All three problems NP-complete [Coo71, Lev73, Kar72]

Conventional wisdom $\Rightarrow$ infeasible to solve in practice

Even practically impossible to find approximate solution in any meaningful sense [Kho01, Zuc07, Hås99, Hås01]

. . . But Easy in Practice?!

SAT

Conflict-driven clause learning (CDCL) solvers [BS97, MS99, MMZ+01]
Deal with real-world instances containing millions of variables
Often run in (close to) linear time!

. . . But Easy in Practice?!

SAT

Conflict-driven clause learning (CDCL) solvers [BS97, MS99, MMZ^{+}01]

Deal with real-world instances containing millions of variables

Often run in (close to) linear time!

CLIQUE

Algorithms in [Pro12, McC17] often work very well in practice

. . . But Easy in Practice?!

Sat

Conflict-driven clause learning (CDCL) solvers [BS97, MS99, MMZ$^+$01]

Deal with real-world instances containing millions of variables

Often run in (close to) linear time!

Clique

Algorithms in [Pro12, McC17] often work very well in practice

Colouring

Award-winning sequence of papers [DLMM08, DLMO09, DLMM11]

Relatively simple linear algebra methods

Authors report being unable to find hard instances!

A Somewhat Sorry State of Affairs...

- Have exponential hardness results for worst-case running time under plausible mathematical assumptions

A Somewhat Sorry State of Affairs. . .

- Have exponential hardness results for worst-case running time under plausible mathematical assumptions
- But these worst-case lower bounds don't seem very relevant for "real-case" problems and algorithms

A Somewhat Sorry State of Affairs...

- Have exponential hardness results for worst-case running time under plausible mathematical assumptions
- But these worst-case lower bounds don't seem very relevant for "real-case" problems and algorithms
- For some of these algorithms we can't even rule out that they would solve NP-complete problems in linear time (also seems preposterous)

A Somewhat Sorry State of Affairs...

- Have exponential hardness results for worst-case running time under plausible mathematical assumptions
- But these worst-case lower bounds don't seem very relevant for "real-case" problems and algorithms
- For some of these algorithms we can't even rule out that they would solve NP-complete problems in linear time (also seems preposterous)
- Since we're not really able to analyse these algorithms, it's very hard to understand
  - when and why they sometimes fail miserably
  - how to improve them

Long-Term Research Goals

- Strengthen the mathematical analysis of algorithmic methods
- Construct stronger algorithms for combinatorial problems
- Develop a better understanding of real-world efficient computation

Mathematical Analysis of Algorithmic Methods

Study methods of reasoning that are powerful enough to capture state-of-the-art algorithms used in practice

Use mathematical tools to establish theorems abut the power and limitations of such algorithms and methods

Recent examples:

- Lower bound $\gtrapprox n^k$ for algorithms [Pro12, McC17] for k-CLIQUE in [ABdR$^+$18]
- Exponential lower bounds for algebraic algorithms [DLMM08, DLMO09, DLMM11] for COLOURING in [MN15, LN17]

Stronger Algorithms for Combinatorial Problems

Use insights into stronger mathemathical methods of reasoning to build algorithms for SAT and other NP-complete problems

Goal: More efficient algorithms having the potential to go significantly beyond state of the art

Stronger Algorithms for Combinatorial Problems

Use insights into stronger mathemathical methods of reasoning to build algorithms for SAT and other NP-complete problems

Goal: More efficient algorithms having the potential to go significantly beyond state of the art

- Pseudo-Boolean solver [EN18] performing very well in the pseudo-Boolean competitions 2015 and 2016 [Pse15, Pse16]
- Try to push further to, e.g.,
  - Pseudo-Boolean optimization
  - Integer linear programming (ILP)
  - Mixed integer linear programming (MIP)
  - Constraint programming (CP)
  - Satisfiability modulo theories (SMT)

Better Understanding of Real-World Efficient Computation

Propose program of "theory-insight-guided empirical research"
(Let's call it TIGER, somewhat tongue-in-cheek ;-))

Better Understanding of Real-World Efficient Computation

Propose program of "theory-insight-guided empirical research"
(Let's call it TIGER, somewhat tongue-in-cheek ;-))

1. Model algorithms as closely as possible and prove rigorous theorems
2. Using these theoretical insights, carefully construct *extremal* benchmarks w.r.t. different complexity-theoretic properties
3. Cannot prove anything *formally*, but theory intuition tells us that instances are likely to be challenging for different heuristics
4. So run experiments on these benchmarks to shed light on
   - what impact each heuristic has on performance
   - how this correlates with theoretical properties
5. Since benchmarks are crafted they are also *scalable*, meaning we can study *how performance scales as the instance size increases*

Better Understanding of Real-World Efficient Computation

Propose program of "theory-insight-guided empirical research"
(Let's call it TIGER, somewhat tongue-in-cheek ;-))

1. Model algorithms as closely as possible and prove rigorous theorems
2. Using these theoretical insights, carefully construct *extremal* benchmarks w.r.t. different complexity-theoretic properties
3. Cannot prove anything *formally*, but theory intuition tells us that instances are likely to be challenging for different heuristics
4. So run experiments on these benchmarks to shed light on
   - what impact each heuristic has on performance
   - how this correlates with theoretical properties
5. Since benchmarks are crafted they are also *scalable*, meaning we can study *how performance scales as the instance size increases*

Early successes reported for CDCL solvers [EGG$^+$18] and pseudo-Boolean solvers [EGNV18, VEG$^+$18]

Better Understanding of Real-World Efficient Computation

Propose program of "theory-insight-guided empirical research"
(Let's call it TIGER, somewhat tongue-in-cheek ;-))

1. Model algorithms as closely as possible and prove rigorous theorems
2. Using these theoretical insights, carefully construct *extremal* benchmarks w.r.t. different complexity-theoretic properties
3. Cannot prove anything *formally*, but theory intuition tells us that instances are likely to be challenging for different heuristics
4. So run experiments on these benchmarks to shed light on
    - what impact each heuristic has on performance
    - how this correlates with theoretical properties
5. Since benchmarks are crafted they are also *scalable*, meaning we can study *how performance scales as the instance size increases*

Early successes reported for CDCL solvers [EGG+18] and pseudo-Boolean solvers [EGNV18, VEG+18]

Thank you for your attention!

References I

[ABdR+18] Albert Atserias, Ilario Bonacina, Susanna F. de Rezende, Massimo Lauria, Jakob Nordström, and Alexander Razborov. Clique is hard on average for regular resolution. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC '18)*, June 2018. To appear.

[BS97] Roberto J. Bayardo Jr. and Robert Schrag. Using CSP look-back techniques to solve real-world SAT instances. In *Proceedings of the 14th National Conference on Artificial Intelligence (AAAI '97)*, pages 203–208, July 1997.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC '71)*, pages 151–158, 1971.

[DLMM08] Jesús A. De Loera, Jon Lee, Peter N. Malkin, and Susan Margulies. Hilbert's Nullstellensatz and an algorithm for proving combinatorial infeasibility. In *Proceedings of the 21st International Symposium on Symbolic and Algebraic Computation (ISSAC '08)*, pages 197–206, July 2008.

[DLMM11] Jesús A. De Loera, Jon Lee, Peter N. Malkin, and Susan Margulies. Computing infeasibility certificates for combinatorial problems through Hilbert's Nullstellensatz. *Journal of Symbolic Computation*, 46(11):1260–1283, November 2011.

References II

[DLMO09] Jesús A. De Loera, Jon Lee, Susan Margulies, and Shmuel Onn. Expressing combinatorial problems by systems of polynomial equations and Hilbert's Nullstellensatz. *Combinatorics, Probability and Computing*, 18(04):551–582, July 2009.

[EGG+18] Jan Elffers, Jesús Giráldez-Cru, Stephan Gocht, Jakob Nordström, and Laurent Simon. Seeking practical CDCL insights from theoretical SAT benchmarks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence (IJCAI-ECAI '18)*, July 2018. To appear.

[EGNV18] Jan Elffers, Jesús Giráldez-Cru, Jakob Nordström, and Marc Vinyals. Using combinatorial benchmarks to probe the reasoning power of pseudo-Boolean solvers. In *Proceedings of the 21st International Conference on Theory and Applications of Satisfiability Testing (SAT '18)*, July 2018. To appear.

[EN18] Jan Elffers and Jakob Nordström. Divide and conquer: Towards faster pseudo-Boolean solving. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence and the 23rd European Conference on Artificial Intelligence (IJCAI-ECAI '18)*, July 2018. To appear.

References III

[Hås99] Johan Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999. Preliminary version in *FOCS '96*.

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, July 2001. Preliminary version in *STOC '97*.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Springer, 1972.

[Kho01] Subhash Khot. Improved inapproximability results for MaxClique, chromatic number and approximate graph coloring. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS '01)*, pages 600–609, October 2001.

[Lev73] Leonid A. Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973. In Russian. Available at `http://mi.mathnet.ru/ppi914`.

References IV

[LN17] Massimo Lauria and Jakob Nordström. Graph colouring is hard for algorithms based on Hilbert's Nullstellensatz and Gröbner bases. In *Proceedings of the 32nd Annual Computational Complexity Conference (CCC '17)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:20, July 2017.

[McC17] Ciaran McCreesh. *Solving Hard Subgraph Problems in Parallel*. PhD thesis, University of Glasgow, 2017.

[MMZ+01] Matthew W. Moskewicz, Conor F. Madigan, Ying Zhao, Lintao Zhang, and Sharad Malik. Chaff: Engineering an efficient SAT solver. In *Proceedings of the 38th Design Automation Conference (DAC '01)*, pages 530–535, June 2001.

[MN15] Mladen Mikša and Jakob Nordström. A generalized method for proving polynomial calculus degree lower bounds. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC '15)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 467–487, June 2015.

[MS99] João P. Marques-Silva and Karem A. Sakallah. GRASP: A search algorithm for propositional satisfiability. *IEEE Transactions on Computers*, 48(5):506–521, May 1999. Preliminary version in *ICCAD '96*.

References V

[Pro12] Patrick Prosser. Exact algorithms for maximum clique: A computational study. *Algorithms*, 5(4):545–587, 2012.

[Pse15] Pseudo-Boolean evaluation 2015. `http://pbeva.computational-logic.org/`, September 2015.

[Pse16] Pseudo-Boolean competition 2016. `http://www.cril.univ-artois.fr/PB16/`, July 2016.

[VEG+18] Marc Vinyals, Jan Elffers, Jesús Giráldez-Cru, Stephan Gocht, and Jakob Nordström. In between resolution and cutting planes: A study of proof systems for pseudo-Boolean SAT solving. In *Proceedings of the 21st International Conference on Theory and Applications of Satisfiability Testing (SAT '18)*, July 2018. To appear.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(6):103–128, August 2007. Preliminary version in *STOC '06*.