Fachgruppe Komplexität

— Newsletter 11 —

In dieser Ausgabe …

Konferenzen 2021

Konferenz	Frist'20	Event'21	Art
ICDT	(21.9.)	23.–26.03.	🌍
FoSSaCS	(15.10.)	27.03.–01.04.	🌍
PODS	(11.12.)	20.–25.06.	ⓘ
STOC	(6.11.)	21.–25.06.	🌍
LICS	(20.01.)	29.06.–02.07.	🌍
SAT	(15.3.)	05.-09.07.	ⓘ
ICALP	(12.02)	12.–16.07.	🌍
CCC	(15.2.)	20.–23.07.	🌍
MFCS	30.04.	23.–27.08.	ⓘ
IPEC	25.06.	08.–10.09.	ⓘ
Highlights	04.06.	15.-17.09.	ⓘ
FOCS	03.06.	early 2022	ⓘ

🌍: online, ◑: hybrid, ⓘ: unknown

Die letzten Theorietage

TT	Wo	Wann
82	Bei Ihnen?	–
81	INFORMATIK'21 🌍	28.09.21
80	Berlin 🌍	13.04.21
79	Hannover 🌍	17.11.20
78	Berlin	10./11.10.19
77	Marburg	28.03.19
76	Halle	24.–25.09.18
75	Ulm	10.–11.04.18
74	Lübeck	23.–25.11.17
73	Hamburg	18./19.05.17

🌍: online

Fachgruppenleitung (2020–2023)

E-Mail an die Fachgruppenleitung

- Arne Meier (Sprecher)
- Till Tantau (stv. Sprecher)

Mitgliederzahl (GI): 300

Kostenlos Mitglied in FG-KP werden

März 2021

Liebe Mitglieder der Fachgruppe Komplexität,

Sie halten den elften Newsletter der GI-Fachgruppe „Komplexität“ in den Händen. Mitte November gab es nach einer einjährigen, Pandemie-geschuldeten, Unterbrechung wieder einen Theorietag – mit der Nummer 79 – der außerdem zum ersten Mal virtuell stattgefunden hat. Auf diesem Workshop wurde auch die Fachgruppenleitung für drei Jahre neu gewählt (mich als Sprecher und Till Tantau als Stellvertreter). Einen kleinen Bericht zum Theorietag (TT) und Ankündigungen zum kommenden sowie zum übernächsten TT finden Sie auf der letzten Seite.

Wie üblich gilt, wenn Sie eine spezielle Konferenz in der linken Spalte vermissen, dann melden Sie sich bei mir, damit wir die Konferenz für die Zukunft aufnehmen können.

Außerdem möchte ich, wie üblich, auf die Möglichkeit zu kurzen inhaltlichen Beiträgen hinweisen. Bei Interesse Ihrerseits melden Sie sich bitte direkt bei mir. Wir planen mit Textvorschlägen von 1–2 Seiten Länge. In dieser Ausgabe gibt es einen Gastbeitrag von Dominik Scheder zu aktuellen Entwicklungen bezüglich dem PPSZ-Algorithmus für k-SAT.

Wenn Sie in die Fachgruppe eintreten möchten, dann ist dies kostenlos als assoziiertes Mitglied möglich — auch ohne eine GI-Mitgliedschaft. Falls Sie bei der GI bisher keine Email hinterlegt haben, so geben Sie diese auf Ihrer Mitgliederseite an, damit der Newsletter auch Sie halbjährig automatisch erreicht.

Der Newsletter ist natürlich auch weiterhin online von unserer Webseite zu beziehen (Publikationen → Newsletter).

Und nun wünsche ich Ihnen viel Spass beim Lesen und bleiben Sie gesund!

Arne Meier, Sprecher der Fachgruppe KP

Die Fachgruppe Komplexität

Die Fachgruppe Komplexität ist ein Teil der Gesellschaft für Informatik. Diese Fachgruppe beschäftigt sich mit komplexitätstheoretischen Fragestellungen. Manche der Themen sind eng gekoppelt an bzw. werden gemeinsam bearbeitet mit anderen Fachgruppen, insbesondere sind dies die FG Algorithmen (Thema: Obere Schranken), FG Automaten und formale Sprachen (Thema: spezielle Berechnungsmodelle, Abschlusseigenschaften von Klassen) FG Logik in der Informatik (Thema: Komplexität logischer Entscheidungsprobleme, Komplexität des logischen Programmierens, subrekursive Hierarchien).

Ein Workshop über Algorithmen und Komplexität, gemeinsam mit der Fachgruppe Algorithmen, findet zweimal jährlich statt.

Gastbeitrag: Super-Strong ETH is true for PPSZ with small resolution width

Dominik Scheder (Shanghai Jiao Tong University, `dominik@cs.sjtu.edu.cn`*)*

The PPSZ algorithm, named after Paturi, Pudlák, Saks, and Zane [3], is the currently fastest known algorithm for k-SAT (ignoring, for the moment, the recent small improvement by Hansen, Kaplan, Zamir, and Zwick [2]). In k-SAT, we are given a Boolean formula F in conjunctive normal form, where each *clause* (conjunct) has at most k literals. Our goal is to find a satisfying assignment, if there is any. Here is how PPSZ does it. It takes a random ordering π of the m variables and goes through them in that order. When processing a variable x, it sets it randomly unless the correct value can be inferred by some *proof heuristic*. Once all variables have been processed, we have either found a satisfying assignment (success) or not (failure). The time complexity of the heuristic is usually assumed to be efficient (polynomial or sub-exponential), so the important parameter to study is the *success probability*, and its dependence on m and k.

There are currently two types of proof heuristics on the market: the *strong heuristic* tries to determine the correct value of x by applying resolution of width bounded by w; the *weak heuristic* tries to derive it by going through all sets of up to d clauses of F. Note that the former heuristic subsumes the latter when $w \geq kd$; therefore, we refer to the first as the *strong heuristic* and the second as the *weak heuristic*, and to the overall algorithm as *strong PPSZ* and *weak PPSZ*, respectively. The parameters w and d can be any slowly growing function in m.

The time complexity of k-SAT is subject to many conjectures. From most to least well-known, these are: P $\neq$ NP (no polynomial time algorithm possible); the exponential time hypothesis, or ETH (no $2^{o(m)}$ algorithm); strong ETH (suppose the "best" k-SAT algorithm has running time $2^{(1-\sigma_k)m}$, then $\sigma_k \to 0$; the σ_k are called the *savings* for k-SAT); and super-strong ETH, which conjectures that $\sigma_k \in \Theta(1/k)$. The justification for super-strong ETH is somewhat empirical: several very different algorithmic paradigms for k-SAT lead to algorithms of running time $2^{m(1-\Theta(1/k))}$.

The complexity of PPSZ. In this contribution, we look at the time complexity of PPSZ and discuss two recent results that, informally, show that super-strong ETH holds for PPSZ. As argued above, the interesting complexity parameter of PPSZ is not its time complexity (which is sub-exponential) but its *success probability*. Obviously, we can trade the former for the latter by repetition. Paturi, Pudlák, Saks, and Zane showed that the success probability of PPSZ on k-CNF formulas is at least $2^{-m(1-s_k)}$, where s_k is the extinction probability in a certain branching process. Moreover, they showed that $s_k = \frac{\pi^2}{6k} + o(1/k)$. It turns out to be surprisingly difficult to construct *hard instances*, i.e., satisfiable k-CNF formulas on which the success probability is small. In this note, we will first introduce a general approach to bounding the success probability from above; then describe two constructions of hard instances and briefly talk about the proofs; finally, we conclude by discussing some shortcomings of current techniques.

Remember the random ordering π of the variables mentioned in the beginning? Suppose it is not random but given to PPSZ as part of the input. When PPSZ can infer the correct value of x by its proof heuristic, we call x *forced*; otherwise we call it *guessed*. If F has a unique satisfying assignment, then every time PPSZ guesses a variable, it guesses correctly with probability $1/2$; if it guesses wrongly even once, it is doomed to fail. We conclude that, for a fixed ordering π, the success probability is $2^{-\textsc{Guessed}(\pi)}$, where $\textsc{Guessed}(\pi)$ is the number of guessed variables. For lower bounds on the success probability, the usual step is to use Jensen's inequality to conclude that $\mathbb{E}_\pi\left[2^{-\textsc{Guessed}(\pi)}\right] \leq 2^{-\mathbb{E}_\pi[\textsc{Guessed}(\pi)]}$ and then bound $\textsc{Guessed}(\pi)$ from above. For upper bounds, this goes in the wrong direction. Currently, the only useful bound is $\mathbb{E}_\pi\left[2^{-\textsc{Guessed}(\pi)}\right] \leq 2^{-\min_\pi \textsc{Guessed}(\pi)}$. This means we have to show that PPSZ has small success probability even if we allow an angel to provide it with the *optimal* ordering π. It is not clear how much oopmh we lose in this step. The upside is that we can now assume that under the optimal π, all guessed variables come before all forced variables (this is an easy exercise). Thus, to prove that $|\textsc{Guessed}(\pi)| > m - B$, we have to show that *not all* of the last B variables in π are forced. Suppose, without loss of generality, that our unique satisfying assignment is $\mathbf{0}$, setting all variables to 0. Fixing the first $m - B$ variables of π to 0, we obtain a new formula F'. Now the statement "the last B variables are forced" translates to "we can infer $x = 0$ for all remaining variables in F' iteratively". All we have to do is to construct a formula F where this is impossible.

The construction Let G be a k-regular graph with n vertices and $m = \frac{kn}{2}$ edges. For each edge e we introduce a variable x_e, and for each vertex u we add the constraint "$\sum_{e:u\in e} x_e \equiv 0 \mod 2$". This can be transformed into a k-CNF formula with m variables and $2^{k-1}n$ clauses. This is a *Tseitin formula* of the graph G. It is obviously satisfiable: $\mathbf{0}$ satisfies it, but many more assignments do, too.

In an exercise in doublethink, let us pretend that $\mathbf{0}$ is the only satisfying assignment. What does fixing the first $m - B$ variables to 0 mean? It means removing the corresponding edges; the remaining formula F' is a Tseitin formula of the remaining graph G' with B edges. Obviously, if $B \geq n$, then G' contains a cycle. Not all edges on that cycle can be forced to 0: after all, setting the edges on the cycle to 1 and the rest to 0 constitutes an alternative satisfying assignment. Thus, the statement "the last B variables are forced" is false whenever $B \geq n$. Now let us leave our fantasy world in which $\mathbf{0}$ is the unique satisfying assignment; after all, in the presence of multiple solutions, it is not even clear what *guessed* and *forced* is supposed to mean. The goal becomes clear: kill off all "alternative" solutions, leaving only $\mathbf{0}$; do this by carefully adding new clauses, making sure that the presence of a cycle in the remaining graph prevents edges thereon from being forced.

Construction for weak PPSZ Doing this last killing step is easy if we want to construct a formula F that is hard for weak PPSZ; remember: weak PPSZ tries to infer variables by looking at size-d-subsets of F. We choose the underlying graph G to have girth much larger than d. Call two edges $e, f \in V(G)$ *far apart* if $\mathrm{dist}_G(e, f) \geq d + 1$ (without defining formally what the distance between two edges is). It should be clear that every cycle of G contains a pair of far-apart edges. For each such far apart pair e, f in G, we add the clause $(\bar{x}_e \vee \bar{x}_f)$ to our Tseitin formula. These 2-clauses are called *bridges* [4]. The resulting formula F has only one satisfying assignment: $\mathbf{0}$. It takes a couple of minutes to see that not all of the last n variables can be forced: they correspond to a set E' of n edges and thus to a subgraph $G' = (V, E')$ of G. Let x_e be the first variable, in the order of π, such that $e \in E'$ and e lies on a cycle. When PPSZ reaches x_e, it cannot infer that $x_e = 0$. Doing so would require it to look at some bridge f_1, f_2 on that cycle, and all edges connecting e to f_1 and f_2. These are simply too many, since it can only look at d clauses at the same time and f_1 and f_2 are far apart. The formal argument is only slightly more elaborate. We conclude: at most $n - 1$ of the $m = \frac{kn}{2}$ variables can be forced. We have just proved the main theorem from Pudlák et al. [4]:

Theorem 1. *Let F be the k-CNF formula just described. Then PPSZ, using the weak heuristic with parameter d, finds the unique satisfying assignment $\mathbf{0}$ with probability at most $2^{-m+n} = 2^{-m(1-2/k)}$.*

Construction for strong PPSZ Sadly, this does not work for strong PPSZ: even starting with the "full" formula F, the solution $\mathbf{0}$ can be deduced by resolution of width at most k. We were simply too liberal in adding bridges $(\bar{x}_e, \bar{x}_f)$. The construction of Scheder and Talebanfard [5] overcomes this but works only for very small resolution width $w \leq c\sqrt{\log \log n}$. We only sketch the idea.

Again, we choose a k-regular graph G on n vertices of large girth g. Every edge e receives a *color* $c(e)$ from a set of *fewer than g colors*. We now add a bridge $(\bar{x}_e \vee \bar{x}_f)$ for all pairs where e and f have the same color. Since every cycle of G contains a pair of edges of the same color, it is easy to see that the resulting formula F has only one satisfying assignment: $\mathbf{0}$. The challenge is to choose the coloring c such that it is hard to infer $x_e = 0$ for at least one of the last B variables. One requirement is intuitive: edges of the same color should be far apart. Furthermore, to facilitate the subsequent resolution width lower bound proof, the coloring should "locally look the same everywhere". Concretely: if an edge e has color 1 and is incident to an edge of color 2, then *every* edge of color 1 should be incident to an edge of color 2. This is achieved by taking a smaller graph, H, also k-regular, with at most $g - 1$ edges, and fixing a *locally bijective homomorphism* $\phi\colon G \to H$. Here, locally bijective means that the k neighbors of u in G are mapped bijectively to the k neighbors of $\phi(u)$ in H. Once the homomorphism ϕ has been fixed, we define the color of an edge $e = \{u, v\}$ of G to be simply $\{\phi(u), \phi(v)\}$. So the "color" is an edge of H, and thus there are fewer than g colors.

The eventual proof uses a couple of additional ingredients. For example, for $B = (1 + \epsilon)n$ for arbitrarily small $\epsilon > 0$, one can show that the graph $G' = (V, E')$ consisting of the last B edges (according to π) still possesses some mild expansion properties, simply by courtesy of having large girth. Second, for the actual proof of the resolution width lower bound, one uses a two-player characterization of resolution width by Atserias and Dalmau [1]. The winning strategy for this game requires H to have large girth. Since H has $\Theta(g)$ vertices, its girth is at most $O(\log g)$, and g, the girth of G, is at most $O(\log n)$. This is where the $\log \log n$ term in the width comes from. The theorem of Scheder and Talebanfard [5] is, in informal terms:

Theorem 2. *The success probability of strong PPSZ with parameter $w \in c\sqrt{\log \log n}$ is at most $2^{-m\left(1-\frac{2+\epsilon}{k}\right)}$, where c depends on ϵ.*

Open questions. The results outlined above show that the worst-case success probability of PPSZ on k-SAT is $2^{-n(1-s_k)}$, where s_k is sandwiched between $\frac{\pi^2}{6k} + o(1/k)$ and $\frac{2}{k} + o(1/k)$. How can we narrow or close this gap? Are there better constructions for hard instances? Even for the instances described in this note, our analysis of the success probability is overly optimistic: we assume PPSZ is given access to the *optimal* variable ordering π. In fact, every known upper bound on its success probability uses this "best permutation" reasoning. How can we break this barrier and argue that the overwhelming majority of orderings is presumably very far from optimal?

Literatur

[1] Albert Atserias and Víctor Dalmau. A combinatorial characterization of resolution width. *J. Comput. Syst. Sci.*, 74(3):323–334, 2008.

[2] Thomas Dueholm Hansen, Haim Kaplan, Or Zamir, and Uri Zwick. Faster k-SAT algorithms using biased-PPSZ. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 578–589. ACM, 2019.

[3] Ramamohan Paturi, Pavel Pudlák, Michael E. Saks, and Francis Zane. An improved exponential-time algorithm for k-SAT. *J. ACM*, 52(3):337–364, 2005.

[4] Pavel Pudlák, Dominik Scheder, and Navid Talebanfard. Tighter hard instances for PPSZ. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 85:1–85:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017.

[5] Dominik Scheder and Navid Talebanfard. Super strong ETH is true for PPSZ with small resolution width. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 3:1–3:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

Nächster Theorietag virtuell am 14. April

Die Frühjahres-Version des Workshops über Algorithmen und Komplexität wird in der 80ten Ausgabe am 14.04.2021 virtuell stattfinden. Ausrichter des Workshops sind Till Fluschnik und Rolf Niedermeier aus Berlin. Der Workshop findet über Zoom™ *virtuell* statt.

Die Webseite zum Workshop gibt weitere Informationen. Der Workshop findet von 9:30–17:00 Uhr statt. Einen eingeladenen Vortrag hält Thomas Erlebach (Uni Leicester) mit dem Titel „Computing with Uncertainty".

Wenn Sie auch einen Vortrag auf dem Workshop beisteuern möchten, melden Sie sich bitte per E-Mail mit einem Titel, Abstract in englischer Sprache sowie (etwaigem) Manuskriptlink. Die Länge eines Vortrags wird 20 Minuten sein gefolgt von einer 5-minütigen Diskussion.

Wenn Sie teilnehmen möchten, bitte via E-Mail anmelden. **Deadline 1. April!**

Impressum

GI Fachgruppe Komplexität

Fachgruppenleitung:
Arne Meier (Sprecher, ViSdPR),
Till Tantau (stv. Sprecher).

Sekretariat +49 511 762 19692
Web https://fg-kp.gi.de
Postalisch
GI-FG Komplexität
PD Dr. Arne Meier
Institut für Theoretische Informatik
Appelstrasse 9A
D-30167 Hannover
Mail fg-kp-leitung@gi.de

Rückblick: Theorietag 79

Der 79. Theorietag fand zum ersten mal virtuell statt. Insgesamt gab es zeitweise über 55 Teilnehmer im virtuellen Vortragsraum. Es war ein gar internationaler Theorietag: es gab Zuschauer aus den Niederlanden, Finnland, Österreich und Frankreich. Man kann zurecht behaupten, dass auf Grund der nicht notwendigen Anreise, es mehr teilgenommen wird als wenn der Workshop in Präsenz stattfindet, wenngleich man auf die geselligen Kaffeepausen mit Gesprächen in kleinen Grüppchen verzichten muss.

Es gab ein vielfältiges Programm: es wurde von dynamischer Komplexität von Problemen (Thomas Zeume) berichtet, QBF thematisiert (Benjamin Böhm), Approximationsalgorithmen präsentiert (Henning Fernau, Kernels ausgenutzt (Tomohiro Koana), dichte und spärliche Teilgraphen berechnet (Frank Sommer), Intervall-Graphen studiert (Klaus Erich Heeger), argumentiert (Yasir Mahmood) sowie Hashfunktionen räumlich untersucht (Stefan Walzer).

Anschließend gab es eine Versammlung der beiden Fachgruppen Algorithmen und Komplexität, um ihre Leitungen neu zu wählen.

Bei der Wahl der Leitung der FG-Komplexität wurden Arne Meier (Hannover) als Sprecher und Till Tantau (Lübeck) als stellvertretenden Sprecher einstimmig gewählt.

Die Fachgruppe Algorithmen wählt Rolf Niedermeier (TU Berlin) als Sprecher und Heiko Röglin (Bonn) als stellvertretender Sprecher. Weiterhin wird ein Leitungsteam gewählt: Petra Berenbrink (Hamburg), Christian Komusiewicz (Marburg), Ulrich Meyer (Frankfurt), Matthias Müller-Hannemann (Halle), Melanie Schmidt (Köln), Sabine Storandt (Konstanz).

Vorausschau: Theorietag 81

Der 81. Workshop über Algorithmen und Komplexität findet im Rahmen der GI-Jahrestagung „INFORMATIK 2021" am *28. September 2021* virtuell in Berlin statt.

Weitere Informationen entnehmen Sie bitte der Webseite: `https://www.thi.uni-hannover.de/de/tt81/`, der Workshop wird von Heribert Vollmer und Arne Meier organisiert.