From Twin-Width to Propositional Logic and Back

Stefan Szeider
TU Wien, Vienna, Austria

Joint work with
Robert Ganian, Filip Pokrývka, André Schidler, Kirill Simonov

FWF
Der Wissenschaftsfonds.

WWTF
VIENNA SCIENCE AND TECHNOLOGY FUND

GAČR
GRANTOVÁ AGENTURA ČESKÉ REPUBLIKY

Happy 60th Birthday, Vadim!

and many returns!

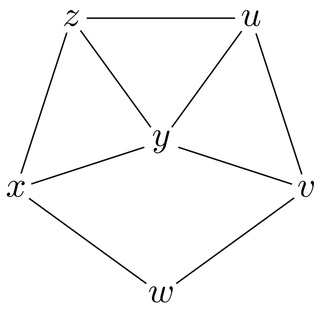
SAT and Graphs

- SAT: given a propositional formula in CNF (i.e., a set of clauses). Is the formula satisfiable?
- We want to apply graph parameters, hence need to associate with the formula formula a graph.
- For an overview see the Handbook of Satisfiability, Chapter 17, 2nd edition, [1]

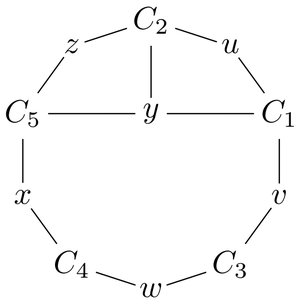
$$F = \{C_1, \ldots, C_5\}$$

$$C_1 = \{u, \overline{v}, y\}, C_2 = \{\overline{u}, z, \overline{y}\}, C_3 = \{v, \overline{w}\}, C_4 = \{w, \overline{x}\}, C_5 = \{x, y, \overline{z}\}$$

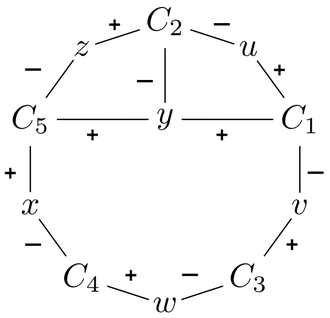
primal graph



incidence graph

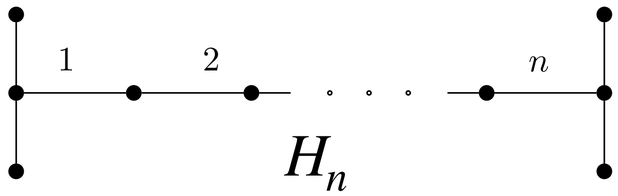


signed incidence graph



Use of incidence graphs

- **David Lichtenstein 1982 [2]:** SAT is NP-hard for planar incidence graphs.
- **Jan Kratochvil 1994 [3]:** remains hard for incidence graphs that are vertex-3-connected, maximum degree 4.
- **Vadim Lozin, Christopher Purcell 2013 [4]:** remains hard for incidence graphs without $K_{1,4}, C_4, \ldots, C_k, H_1, \ldots, H_k$ as induced subgraphs. For $k \to \infty$ this sequence converges to a "minimal boundary property".



SAT and width parameters

W-hard

FPT (certificate at least FPT-approximable)

cwd

↓

sign-cwd

↓

inc-tw

↓

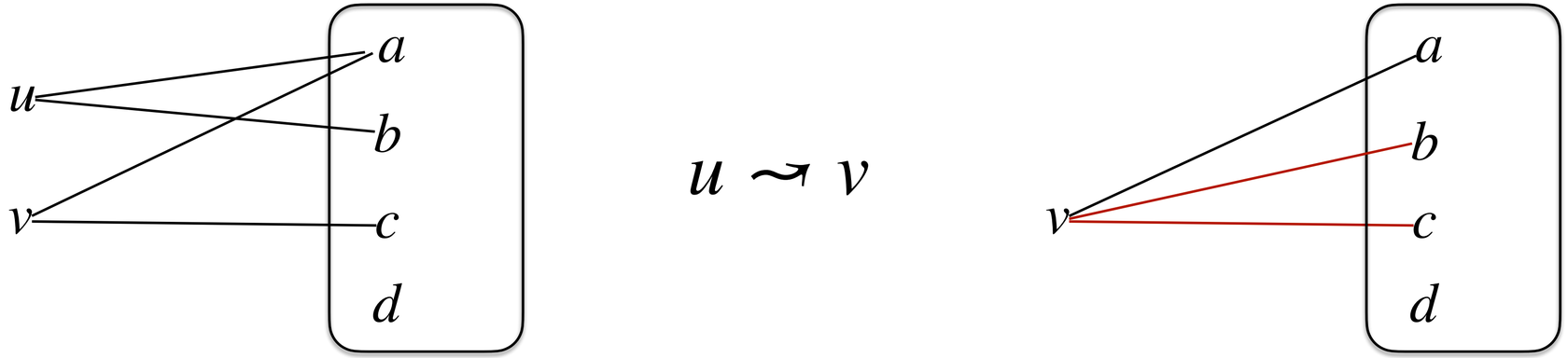
prim-tw

SAT and Twin-Width (tww)

- A new graph parameter, introduced by Bonnet et al. [5]
- Graphs of bounded tww capture many others sparse and dense graph classes.
- Graphs of bounded tww admit tractable First-Order model checking
- **Q1: can we use tww for SAT?**
- **Q2: can we use SAT to compute tww?**

Twin-Width of Graphs

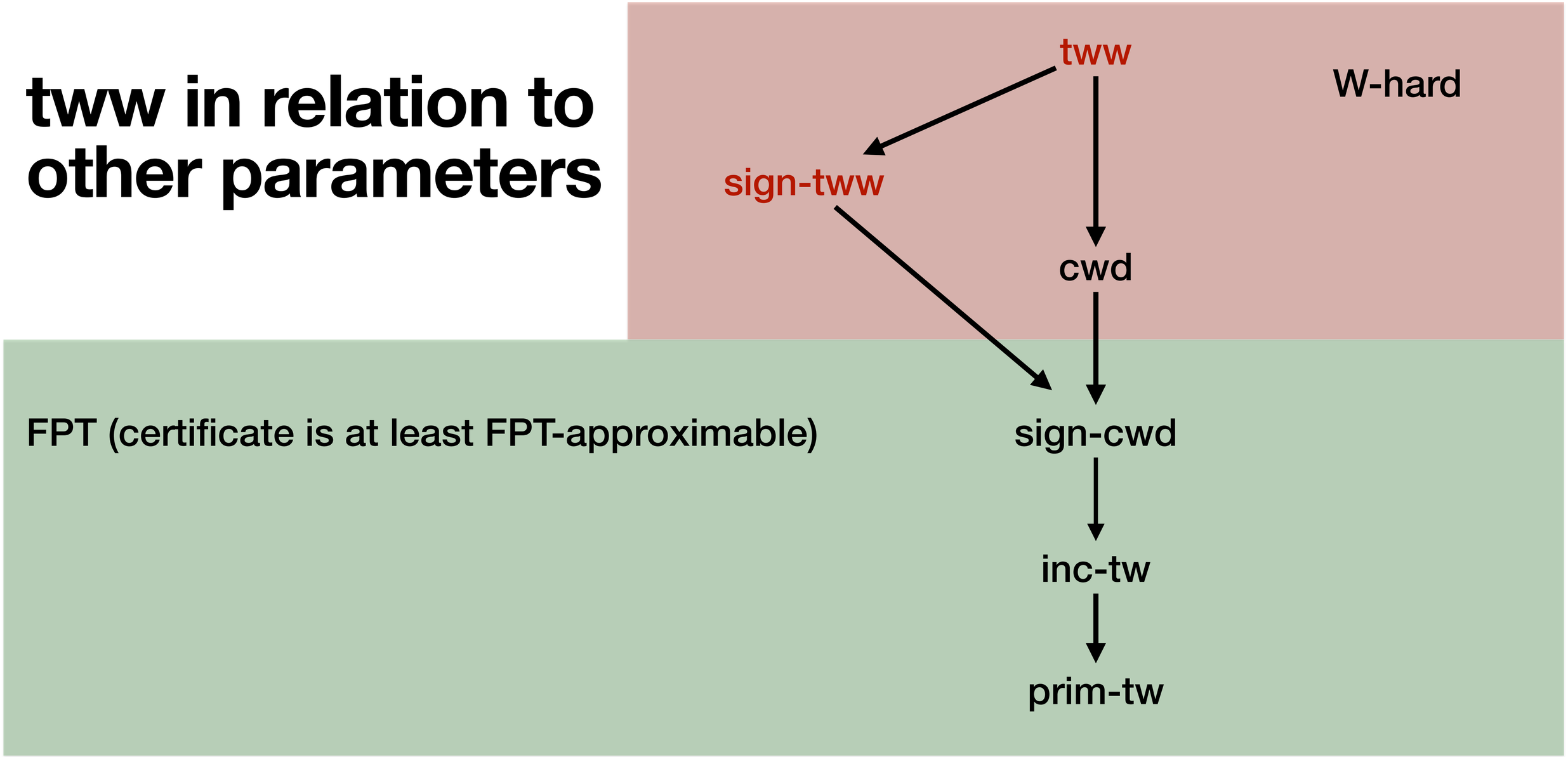
- Reduce a given Graph to a single vertex by a sequence of contractions.
- Each contraction removes a vertex u by contracting it to one of the remaining vertices v. In symbols $u \rightsquigarrow v$.
- If u, v are twins, then the contraction is perfect.
- if u, v are not twins, we record the error by colouring edges red.
- red edges remain red in subsequent steps



Twin-width of Graphs

- A **d-contraction sequence** of a graph contracts all vertices step-by-step to a single vertex graph, such that each intermediate graph has **red degree** at most d.
- $G = G_n \rightsquigarrow G_{n-1} \rightsquigarrow G_{n-2} \rightsquigarrow \cdots \rightsquigarrow G_2$
- The **twin-width** of a graph is the smallest d such that it admits a d-contraction sequence.

tww in relation to other parameters



Q1: can we use tww for SAT?

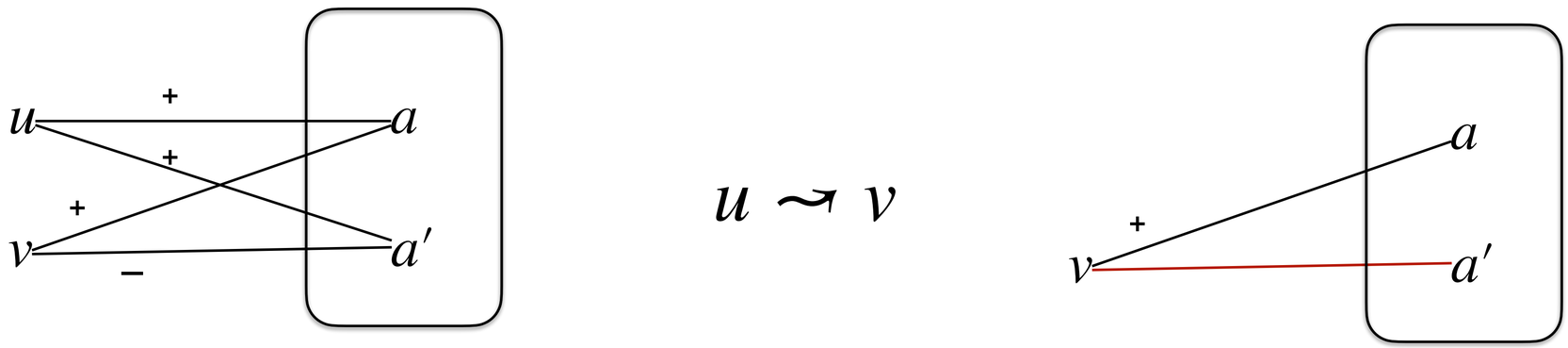
- paraNP-hard for primal-tww and incidence-tww

  follows from hardness of planar SAT

- not unexpected… does it help to consider the **signed** incidence graph?

Signed twin-width [6]

- The given graph G is **signed**, i.e., all edges are labeled + or -.
- A d-contraction sequence is defined as before, except that contracting black edges of different signs become red as well.



- For bipartite signed graphs, we can assume that we always contract vertices that belong to the same side of the partition. We can show that the tww does only change by a small constant factor [4].

Q1: can we use tww for SAT?

- SAT remains paraNP—hard for signed-tww
- Idea: take second parameter k which bounds the number of variables assigend true.
- **Bounded-Ones-SAT**: is the formula satisfiable by a truth assignment that sets at most k variables to true?
- W[2]-hard for parameter k, but together with (signed) tww we have a chance.
- With parameters k + tww we still have W[1]-hardness.

Finally….

Theorem: Bounded-ones-SAT is fixed-parameter tractable when parameterised by the certified signed tww of F and k.

- We can even do **bounded weighted model counting** in FPT time (compute the sum of weights of all satisfying assignements that set at most k variables to true)
- by Ganian, Pokrývka, Schidler, Simonov, Sz. SAT 2022 [6]

DP Algorithm

$$G = G_n \rightsquigarrow G_{n-1} \rightsquigarrow G_{n-2} \rightsquigarrow \cdots \rightsquigarrow G_2$$

R_n R_{n-1} R_{n-2} R_2

- We proceed by dynamic programming (as typical for width measures)
- Computing records for each intermediate graph from its predecessor
- Record for first graph is trivial.
- We can read off the solution from the record of the last graph
- Intuition: R_i only needs to store information on small local parts of G_i that a red-connected
- Size of local parts is bounded by $k(d^2 + 1)$

Computing tww

- Recognizing graphs of tww at most 4 is NP-hard, shown by Bergé, Bonnet, Déprés ICALP 2022 [7].
- However, for our FPT algorithm we only need an FPT-approximation for signed tww, i.e., an FPT algorithm that finds an f(d)-contruction sequence for graphs of signed tww d.
- Such algorithms exist for (signed) cwd, currently unknown for (signed) tww.

Q2: can we use SAT to compute tww?

- General Idea, for any width measure w
- Given a graph G and an integer k.
- Construct in poynomial time a propositional formula $F_w(G, k)$ (“encoding”) such that

  $F_w(G, k)$ is satisfiable if and only if $w(G) \leq k$.

- to determine $w(G)$ check the satisfiability of

$$F_w(G,1),\ F_w(G,2),\ F_w(G,1),\ F_w(G,3),\ldots,\ F_w(G,k),\ F_w(G,k+1),\ F_w(G,k+2)$$

UNSAT	SAT

- We can think of the encoding as a **nondeterministic** algorithm

SAT solvers

- here we utilize the enourmous (and unreasonable) efficiency of todays SAT solvers.
- Instances with millions of variables and clauses can be handled.
- Today's leading SAT solvers are based on the **conflict-driven clause learning algorithms** (CDCL)
- See our forthcoming survey in Comm. ACM [8]

SAT encodings for width measures

parameter	PC	ref
treewidth	fpt	Samer, Veith 2009
branchwidth	fpt	Lodha,Ordynaik,Sz 2016
pathwidth	fpt	Lodha,Ordynaik,Sz 2017
special tw	fpt	Lodha,Ordynaik,Sz 2017
tree-depth	fpt	Ganian,Lodha,Ordynaik,Sz 2018
clique-width	fpt-apx	Heule, Szeider 2015
tree-cut width	fpt-apx	Ganian,Lodha,Ordynaik,Sz 2018
twin-width	paraNP-h	Schidler, Sz 2022 [9]
hypertree width	XP	Schidler, Sz 2020, 2022
generalised htw	paraNP-h	Fichte,Hecher, Lodha, Sz 2018
fractional htw	paraNP-h	Fichte,Hecher, Lodha, Sz 2018

Relative encoding of tww

- Guess a linear ordering of the vertices by means of variables $ord_{i,j}$ meaning $v_i < v_j$ and enforcing transitivity.
- Guess for each vertex v_i the vertex v_j it gets contracted to by the variable $p_{i,j}$
- At each position in the ordering compute the red subgraph H_i of G_i
- With **cardinality constraints** bound the maximum degree of H_i
- For an input graph with n vertices, the number of clauses is $O(n^4)$

Absolute encoding of tww

- Reduce the number of clauses to $O(n^3)$.
- Guess for each vertex its position in the ordering
  $o(i, j) =$ true iff vertex v_i has position j in the ordering.
- Interestingly, the absolute encoding performs worse than the relative encoding, since it has poorer propagation properties.

tww of famous named graphs

- With our encodings we could determine the exact tww of several based graphs, Schidler, Sz. ALENEX 2022 [9]
- Some results have been independently established by Ahn, Hendrey, Kim, Oum 2021 [10]
- See Jungho's talk tomorrow!

Graph	tww
Brinkmann	6
Chvatal	3
Clebsch	6
Desargues	4
Dodecahedron	4
Errera	5
FlowerSnark	4
Folkman	3
Franklin	2
Frucht	3
Grid6×8*	3
Grötzsch	3
Herschel	2
Hoffman	4
Holt	6
Kittell	5
McGee	4
Nauru	4
Paley-73*	36
Pappus	4
Peterson	4
Poussin	3
Robertson	6
Rook 6 × 6	10
Shrikhande	6
Sousselier	4
Tietze	4

tww of signed incidence graphs [6]

- We modified the relative encoding to handle bipartite signed graphs
- computed the signed bipartite tww of some standard SAT instances
- up to 130 vertices and up to 550 edges.

Conclusion

- Q1: can we use tww for SAT?
- Q2: can we use SAT to compute tww?
- Future work 1: find parameter between signed cwd and signed tww that allows FPT without bounding the ones.
- Future work 2: improve the SAT encoding

(tww is the subject of the PACE 2023 competition! https://pacechallenge.org/2023/)

sign-tww

↓

?

↓

sign-cwd

Questions?

References

[1] Marko Samer and Stefan Szeider. Fixed-parameter tractability. In Armin Biere, Marijn Heule, Hans van Maaren, and Toby Walsh, editors, *Handbook of Satisfiability*, 2nd Edition, chapter 17, pages 693–736. IOS Press, 2021, https://www.ac.tuwien.ac.at/files/tr/ac-tr-21-004.pdf

[2] David Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11(2):329–343, 1982.

[3] Jan Kratochvil. A special planar satisfiability problem and a consequence of its NP-completeness. *Discret. Appl. Math.,* 52(3):233–252, 1994.

[4] Vadim V. Lozin and Christopher Purcell. Boundary properties of the satisfiability problems. *Inf. Process. Lett.*, 113(9):313–317, 2013.

[5] Edouard Bonnet, Eun Jung Kim, Stephan Thomasse, and Remi Watrigant. Twin-width I: tractable FO model checking. *J. ACM*, 69(1):3:1–3:46, 2022.

[6] Robert Ganian, Filip Pokryvka, Andre Schidler, Kirill Simonov, and Stefan Szeider. Weighted model counting with twin-width. In Kuldeep S. Meel and Ofer Strichman, editors, 25th International Conference on Theory and Applications of Satisfiability Testing, *SAT 2022*, August 2-5, 2022, Haifa, Israel, volume 236 of LIPIcs, pages 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. https://drops.dagstuhl.de/opus/volltexte/2022/16689/

[7] Pierre Berge, Edouard Bonnet, and Hugues Depres. Deciding twin-width at most 4 is NP-complete. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, 49th International Colloquium on Automata, Languages, and Programming, *ICALP 2022*, July 4-8, 2022, Paris, France, volume 229 of LIPIcs, pages 18:1–18:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[8] Johannes Fichte, Markus Hecher, Daniel Leberre, and Stefan Szeider. The silent (r)evolution of SAT. *Communications of the ACM*, 2022, to appear.

[9] Andre Schidler and Stefan Szeider. A SAT approach to twin-width. In Cynthia A. Phillips and Bettina Speckmann, editors, Proceedings of *ALENEX 2022*, the 24nd SIAM Symposium on Algorithm Engineering and Experiments, pages 67–77, 2022 https://doi.org/10.1137/1.9781611977042.6

[10] Jungho Ahn, Kevin Hendrey, Donggyu Kim, and Sang-il Oum. Bounds for the twin-width of graphs. *CoRR,* abs/2110.03957, 2021.