A SAT Approach to Clique-Width

Marijn Heule
University of Texas at Austin

Stefan Szeider
Vienna University of Technology

GROW 2013
Santorini Island

SAT solving

- a SAT solver checks whether a propositional formula in CNF is satisfiable
- Over the last 20 years, SAT solvers have become extremely efficient
- State-of-the-art solvers:
  CDCL ("*conflict-driven clause-learning*")
  [Marques-Silva, Sakallah 1999] plus other techniques
- Real-world instances with millions of variables can be solved in seconds
- key application: SW/HW verification

"NP is the new P"

SAT solvers in Combinatorics

- Quasigroup problems [Zhang, Hsiang 2004]
- Van der Waerden numbers [Dransfield, Liu, Marek, Truszczynski 2004; Herwig, Heule, van Lambalgen, van Maaren 2007; Ahmed 2011]
- Tao-Green numbers [Kullmann 2010]
- Treewidth [Samer, Veith 2009]

Our work: use SAT to compute clique-width

- new formulation of clique-width
- special SAT-encoding
- Experiments

Clique-Width

- introduced by [Courcelle, Engelfriet, and Rozenberg 1991,1993]
- based on graph composition process using vertex-labeled graphs.
- clique-width(G) is the smallest number of labels needed to construct G

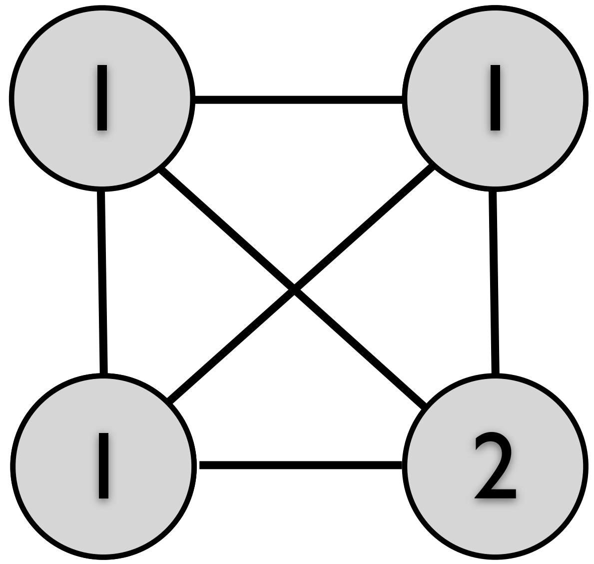
Axiom: singleton-graphs *denoted* $i(v)$

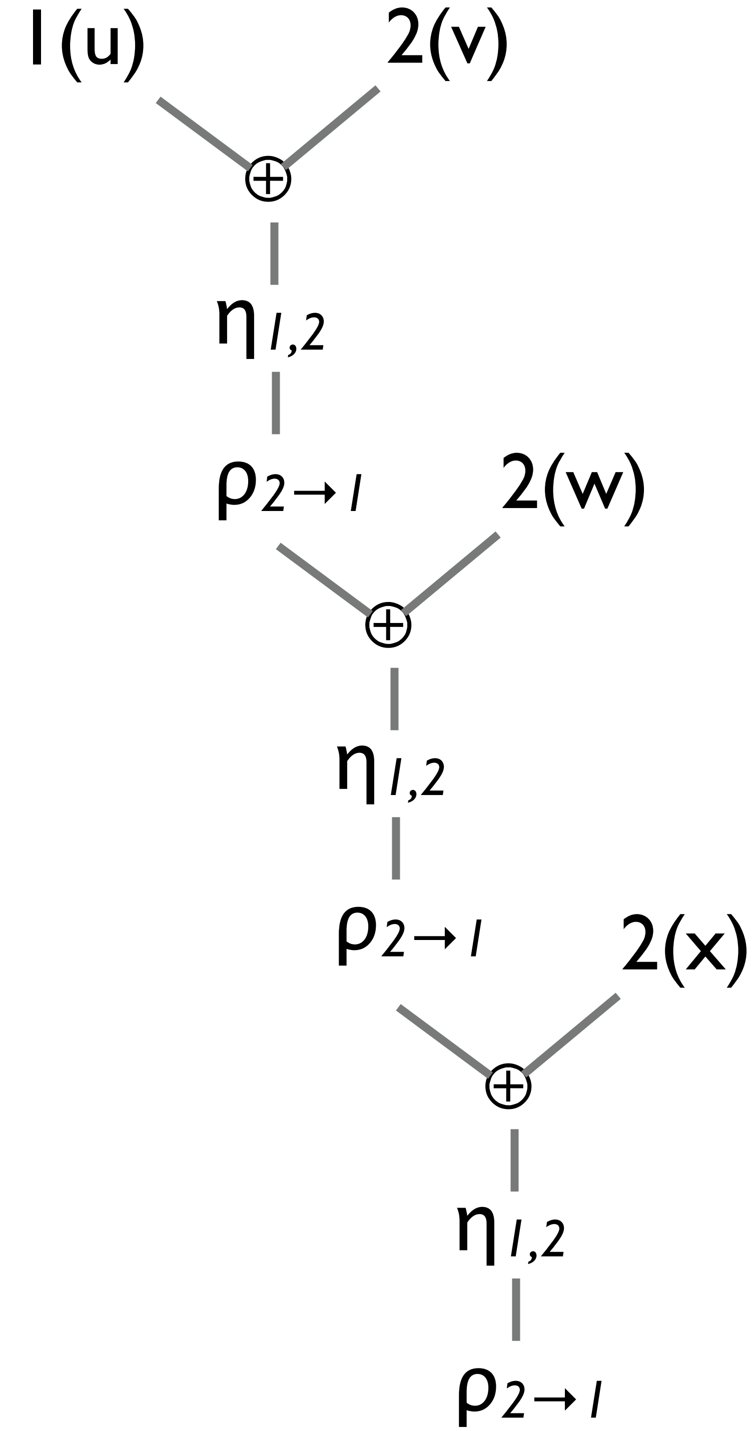
Rules: disjoint union *denoted* $\oplus$

uniform relabeling *denoted* $\rho_{i \to j}$

uniform edge insertion *denoted* $\eta_{i,j}$

Example: K_4





certificate: k-expression

$$(\rho_{2\to 1}(\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(\rho_{2\to 1}(\eta_{1,2}(1(u) \oplus 2(v)))) \oplus 2(w)) \oplus 2(x))$$

Clique-Width

- MSO formulas can be checked in linear time on graphs of certified bounded clique-width [Courcelle, Makowsky, Rotics. 2000,2001]
- Requires certificates (i.e., k-expression).
- How to find certificates? How to recognize graphs of clique-width $\leq k$?

recognizing clique-width $\leq$ k

- k part of input:
  - NP-complete
    [Fellows, Rosamond, Rotics, Szeider 2006,2009]
- Fixed k:
  - k=1 trivial (edge-less graphs)
  - k=2 polynomial-time (cographs)
  - k=3 polynomial time (very complicated, [Corneil, Habib, Lanlignel, Reed, Rotics 2000,2012])
  - k=4 open
- Parameterized by k:
  - fixed-parameter tractability: open
  - fixed-parameter approximable (error exponential in k) via *rank-width* [Oum, Seymour 2006, Oum 2008]

practical algorithms?

- no *practical* algorithms known
- not well-suited for heuristics
  in contrast to
  Boolean-width [Hvidevold, Sharmin, Telle, Vatshelle 2012] and
  rankwidth [Beyß 2013]
- clique-width is unknown, even for very small graphs

The SAT approach

Use SAT solvers!

- Clearly the problem "cwd(G) $\leq$ k?" is in NP
- Hence, in principle we can produce in polynomial time a propositional formula F(G,k) which is satisfiable iff cwd(G) $\leq$ k
- Then we can check the satisfiability of F(G,k) with a SAT solver
- From a satisfying assignment we can obtain a k-expression.
- To make this work in practice needs some work

SAT encodings

- naive encoding (with tricks): $n \leq 8$
- developed new encoding, based on k-derivations
- *Remark*: similar formulation of cwd was used by [Heggernes, Meister, Rotics 2011]

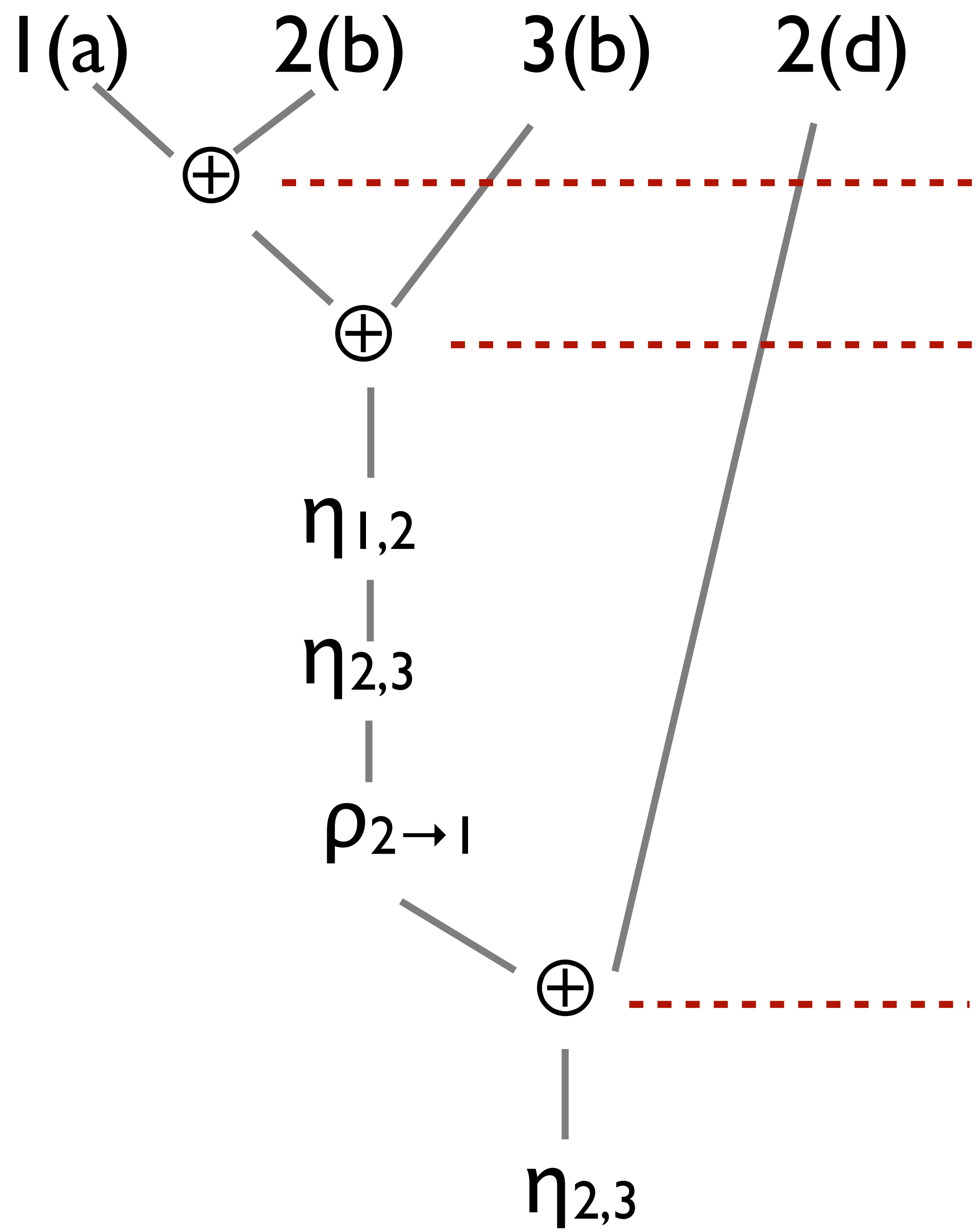
k-derivations

- A *k-derivation* consists of two sequences of partitions of V(G)

	components;	*groups*
T_0:	{{a},{b},{c},{d}};	{{a},{b},{c},{d}}
T_1:	{{a,b},{c},{d}};	{{a},{b},{c},{d}}
T_2:	{{a,b,c},{d}};	{{a},{b},{c},{d}}
T_3:	{{a,b,c,d}};	{{a,b},{c},{d}}

- both start with singletons and get coarser and coarser
- at any time T_i each component is the union of at most k groups (“*Cardinality Condition*”).

k-derivations: components



T_0 {a}, {b}, {c}, {d}

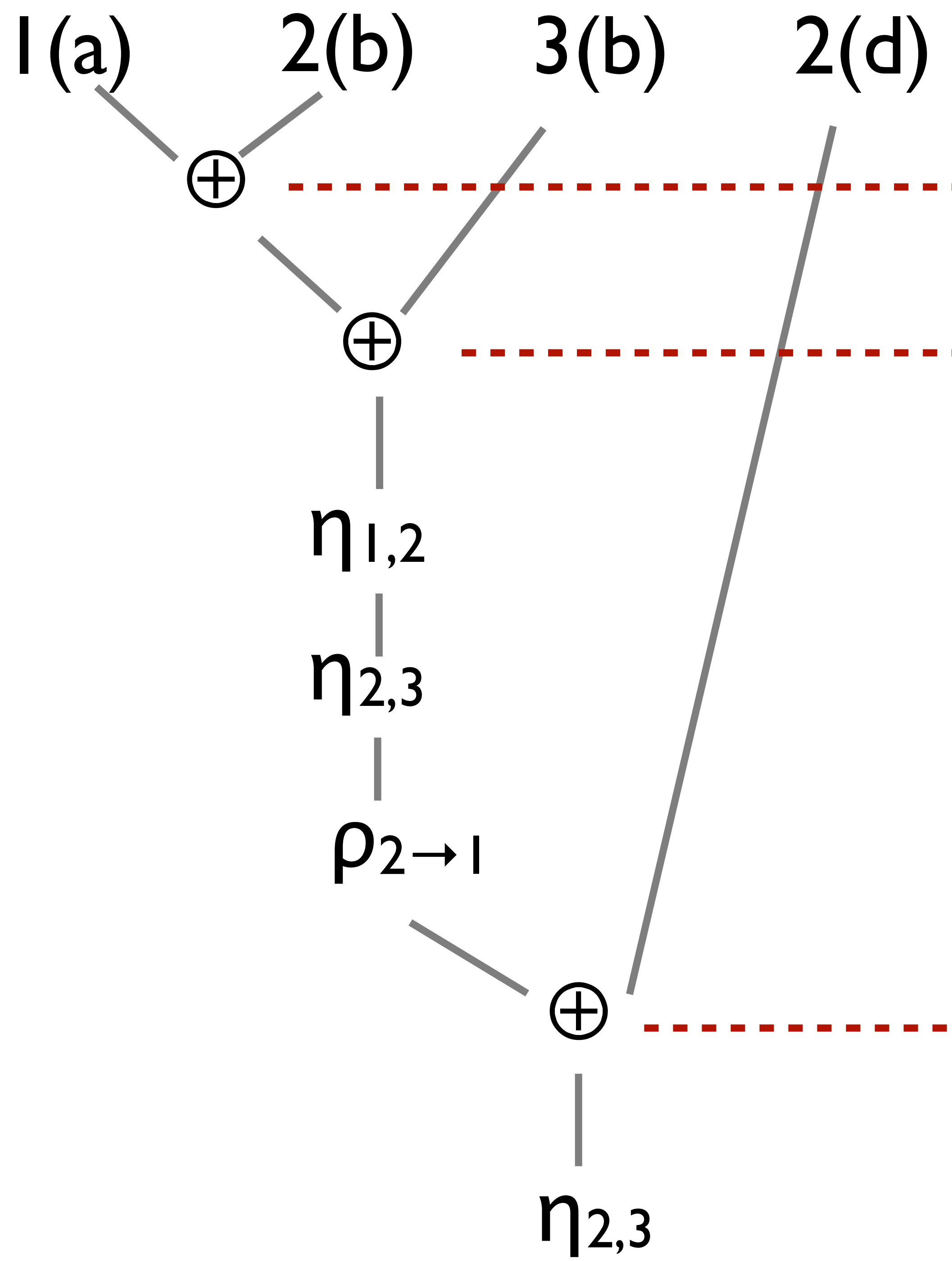
T_1 {a,b}, {c}, {d}

T_2 {a,b,c}, {d}

T_3 {a,b,c,d}

partition of V, two vertices are in the same equivalence class iff they are in the same graph at time T_i

k-derivations: groups



T_0 {a}, {b}, {c}, {d}

T_1 {a}, {b}, {c}, {d}

T_2 {a},{b},{c}, {d}

partition of V, two vertices are in the same equivalence class iff they are in the same component and have the same label at time T_i

T_3 {a,b},{c},{d}

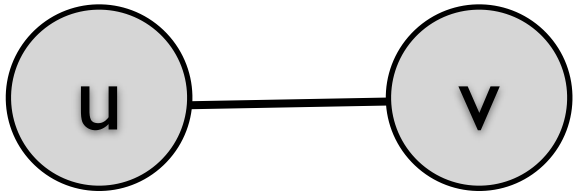
What about edges?

- To check whether a k-derivation represents a graph, it must satisfy certain properties
- These properties are “light weight” and easily expressed by clauses

Three Properties

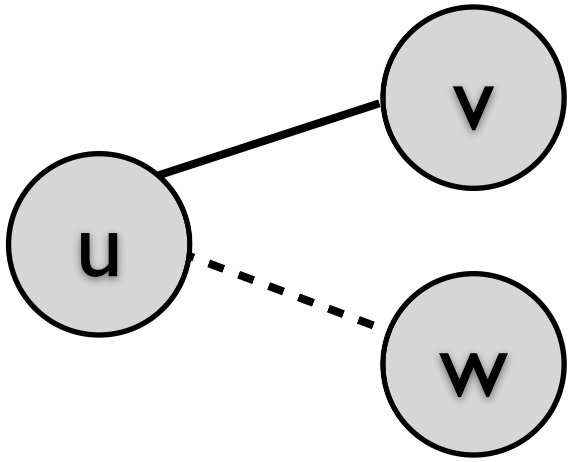
1. Edge Property:

$e(u,v) \wedge g(u,v,i) \;\Rightarrow\; c(u,v,i\text{-}1)$



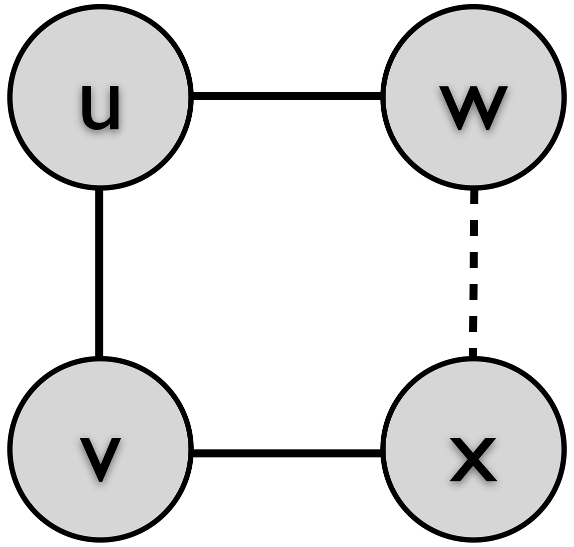
2. Neighborhood Property:

$e(u,v) \wedge \neg e(u,w) \wedge g(v,w,i) \Rightarrow c(u,v,i\text{-}1)$



3. Path Property:

$e(u,v) \wedge e(u,w) \wedge e(v,x) \wedge \neg e(w,x) \wedge$
$g(u,x,i) \wedge g(v,w,i) \Rightarrow c(u,v, i\text{-}1)$



Equivalence

- *A graph has clique-width at most k iff it has a k-derivation*
- *k-derivations and k-expressions can be translated into each other in polynomial time*

The formula F(G,k)

- variables:
  - c(u,v,t): u and v are in the same component at time t
  - g(u,v,t): u and v are in the same group at time t
- clauses:
  - basic clauses (transitivity, partitions get coarser,..)
  - clauses that represent the three properties
  - clauses that represent the Cardinality Condition

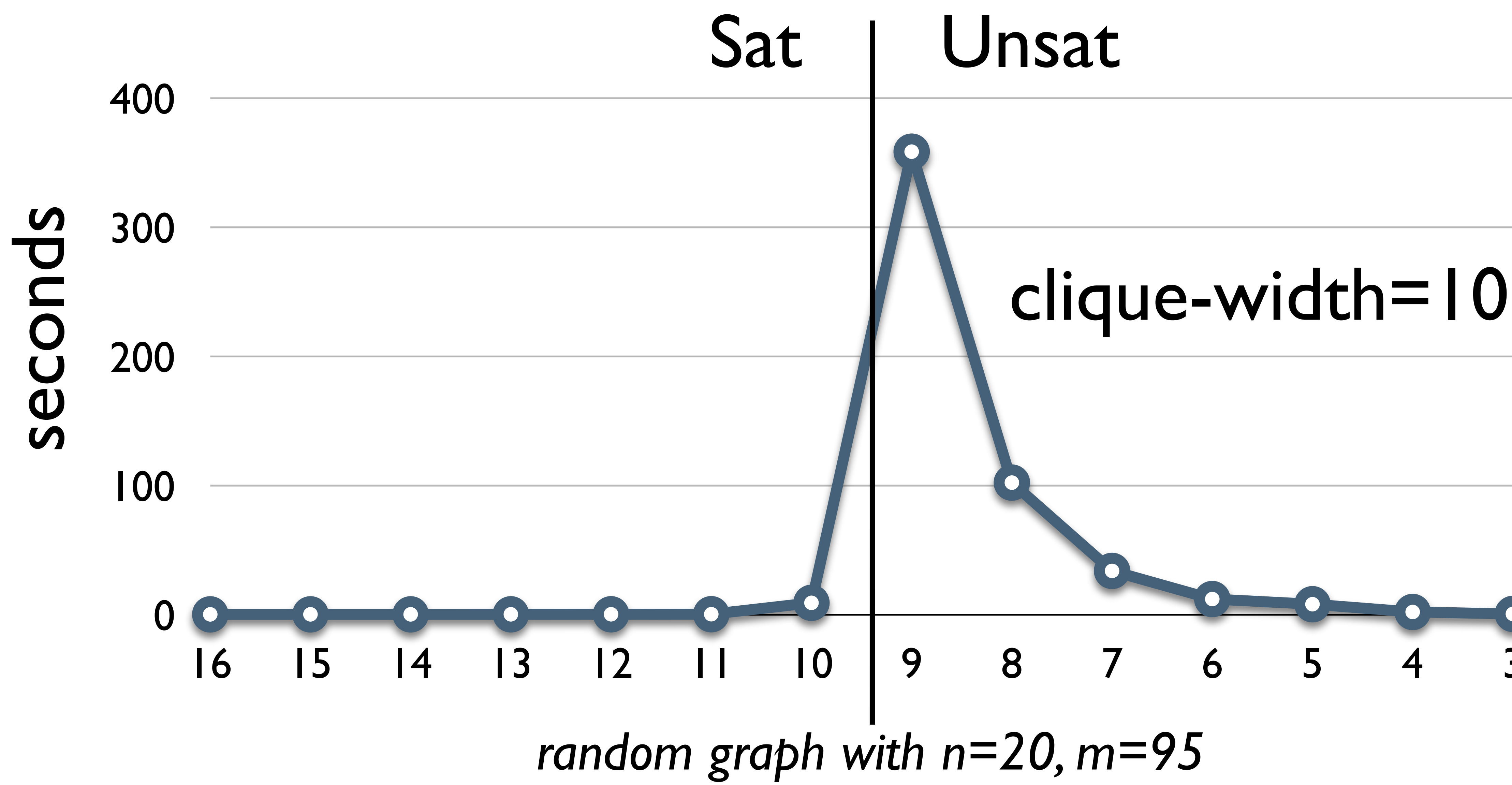
Encoding the Cardinality Condition

- We have developed an encoding based on "*order encoding*" [Tamura, Taga, Kitagawa, Banbara 2009]
- advantage: *conflicts are detected earlier*
- speeds up the process by an order of magnitude

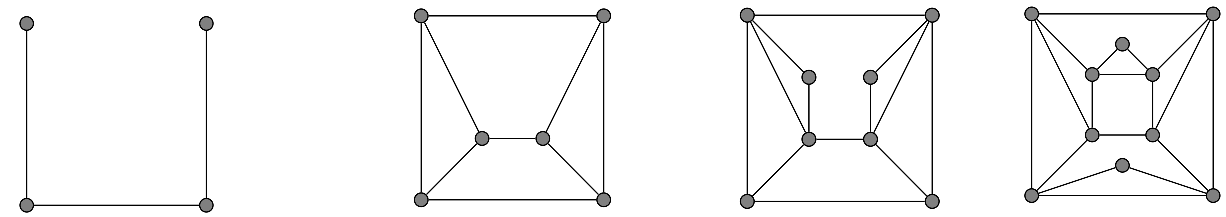
Results

Experimental setup

- Use the Glucose 2.2 solver [Audemard and Simon 2009]
- We initialize with k=n and decrease k until we get an unsatisfiable formula



All graphs up to 10 vertices

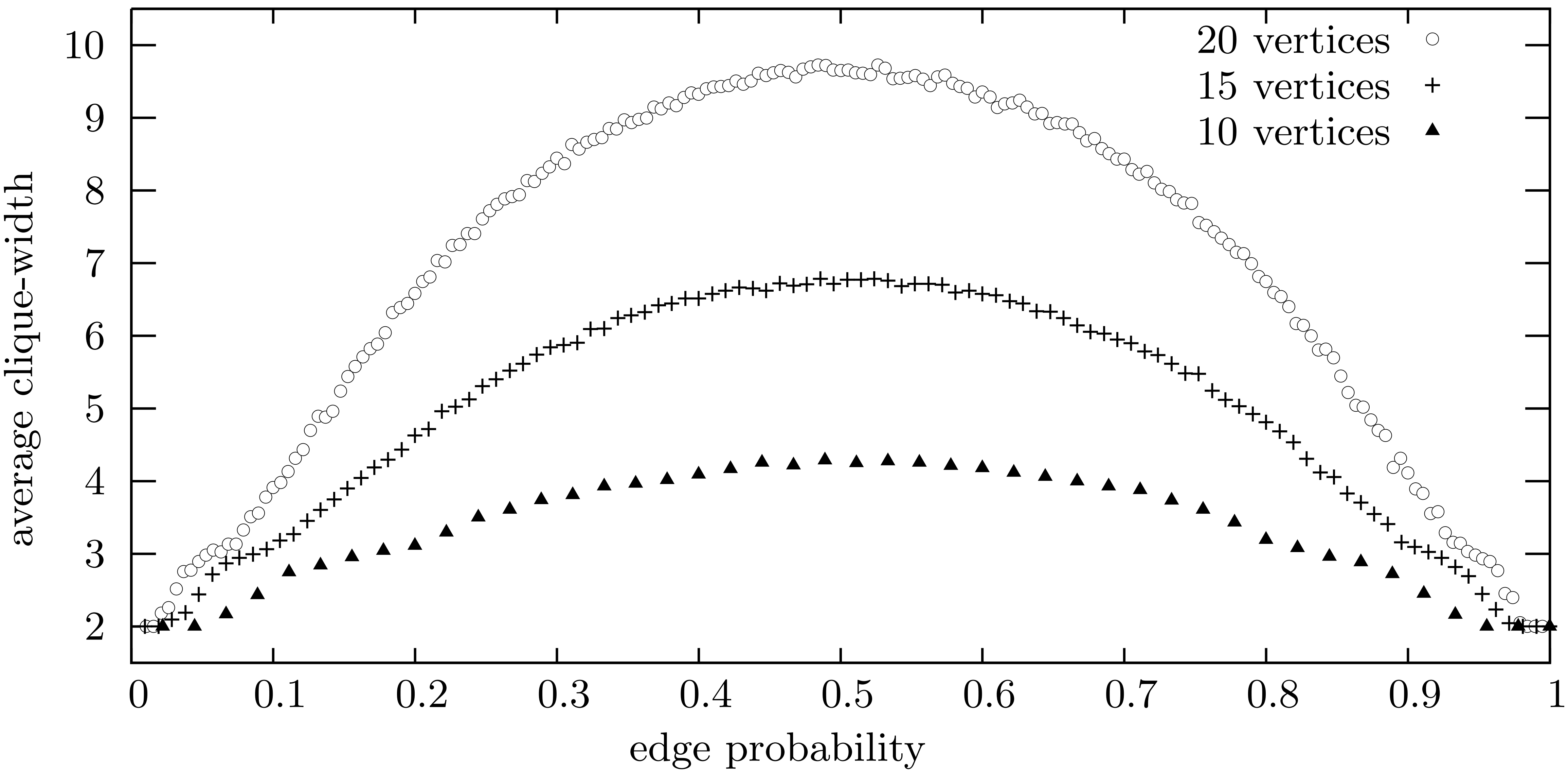


			clique-width				
$\|V\|$	connected	prime	2	3	4	5	6
4	6	1	0	**1**	0	0	0
5	21	4	0	4	0	0	0
6	112	26	0	25	**1**	0	0
7	853	260	0	210	50	0	0
8	11,117	4,670	0	1,873	2,790	**7**	0
9	261,080	145,870	0	16,348	125,364	4,158	0
10	11,716,571	8,110,354	0	142,745	5,520,350	2,447,190	**68**

clique-width sequence: 1,2,4,6,8,10,11,...

Random Graphs

an asymptotic analysis provided by [Lee, Lee, Oum 2012].
We analyzed small random graphs with n=10,15,20

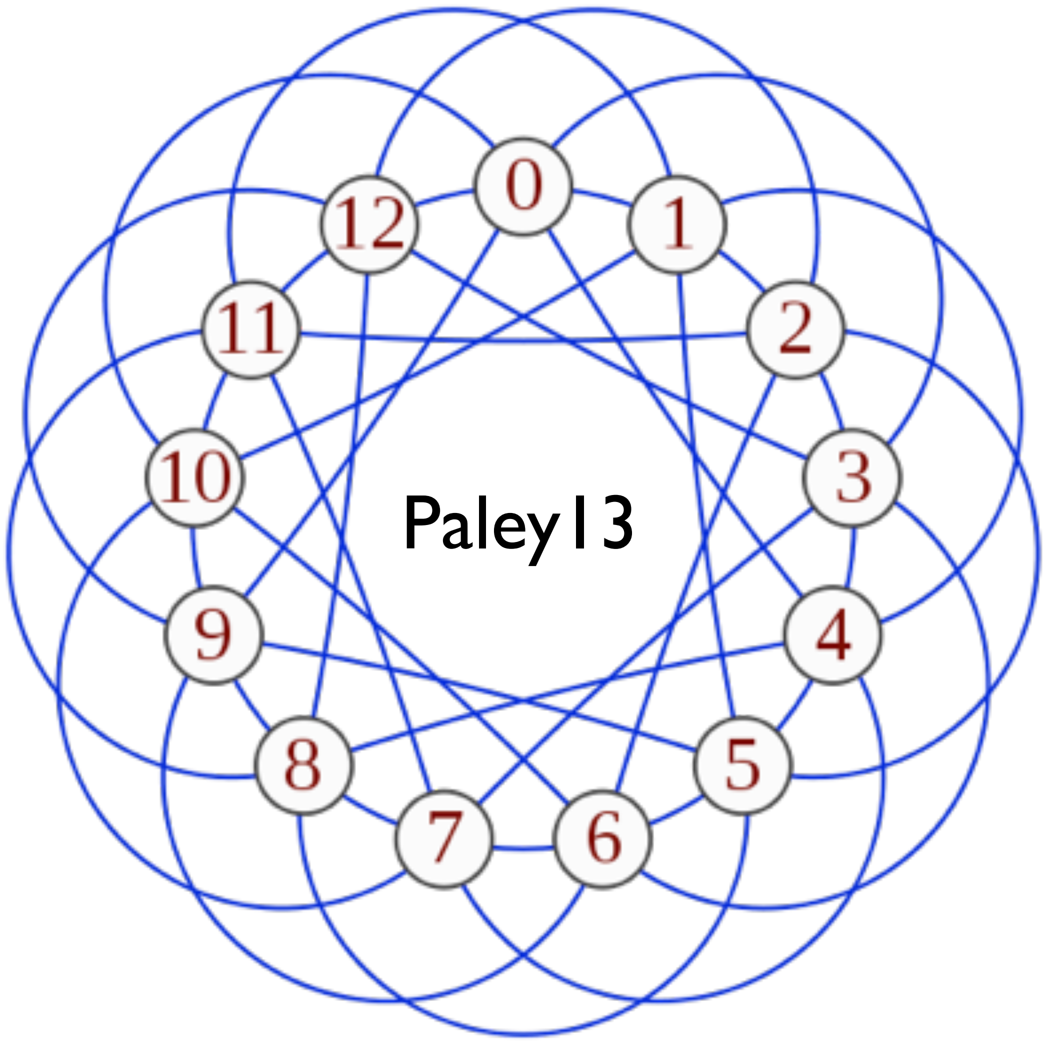


Famous Graphs (selection)

graph	$\|V\|$	$\|E\|$	cwd	variables	clauses	UNSAT	SAT
Brinkmann	21	42	10	8,526	163,065	3,933	1.79
Clebsch	16	40	8	3,872	60,520	191	0.09
Desargues	20	30	8	7,800	141,410	3,163	0.26
Dodecahedron	20	30	8	7,800	141,410	5,310	0.33
Errera	17	45	8	4,692	79,311	82	0.16
Flower snark	20	30	7	8,000	148,620	276	3.90
Folkman	20	40	5	8,280	168,190	12	0.36
Kittell	23	63	8	12,006	281,310	179	18.65
McGee	24	36	8	13,680	303,660	8,700	59.89
Paley-13	13	39	9	1,820	22,776	13	0.05
Paley-17	17	68	11	3,978	72,896	194	0.12
Pappus	18	27	8	5,616	90,315	983	0.14
Robertson	19	38	9	6,422	112,461	478	0.76

Paley Graphs

constructed from the members of a finite field by connecting pairs of elements that differ in a quadratic residue



Raymond Paley (1907–1933)
English mathematician
construction of Hadamard matrices

Conclusion

Summary

- first practical implementation for clique-width computation
- based on an efficient SAT encoding: (k-derivations, order encoding)
- compute exactly the clique-width of various graphs, identify critical graphs

Future work

- Extend the approach to variants of clique-width (such as: *signed cwd, linear cwd, m-cwd, NLC-with*)
- Compute cwd upper bounds for larger graphs
- We need at most n-k+1 time steps.
  Are better bounds possible?
  related work: "shrub-depth" [Ganian, Hlinený, Nesetril, Obdrzálek, Ossona de Mendez, Ramadurai 2012]
- Application: Use cwd-critical graphs as gadgets for reductions (is cwd=4 NPc?)