Long Proofs of (Seemingly) Simple Formulas

Mladen Mikša

KTH Royal Institute of Technology
Stockholm, Sweden

17th International Conference on Theory and Applications of Satisfiability Testing

Vienna, Austria
15 July 2014

Joint work with Jakob Nordström

SAT Solvers: Triumphs and Limits

State-of-the-art CDCL SAT solvers

Very successful in practice, can solve instances with millions of variables

Exist small instances with just a few hundred variables that are hard

Natural questions

1. When does CDCL work and why?
2. What is the smallest formula that is infeasible in practice?

This work: Focus on second question

(Some) Theoretical Hardness Results

Pigeonhole principle [Haken '85]

Claims existence of matching between $n+1$ pigeons and n holes

(Some) Theoretical Hardness Results

Pigeonhole principle [Haken '85]

Claims existence of matching between $n+1$ pigeons and n holes

Tseitin formulas [Urquhart '87]

Set of inconsistent parity constraints

(Some) Theoretical Hardness Results

Pigeonhole principle [Haken '85]

Claims existence of matching between $n+1$ pigeons and n holes

Tseitin formulas [Urquhart '87]

Set of inconsistent parity constraints

Random 3-CNF formulas [Chvátal, Szemerédi '88]

Randomly sampled 3-CNF formula

(Some) Theoretical Hardness Results

Pigeonhole principle [Haken '85]

Claims existence of matching between $n + 1$ pigeons and n holes

Tseitin formulas [Urquhart '87]

Set of inconsistent parity constraints

Random 3-CNF formulas [Chvátal, Szemerédi '88]

Randomly sampled 3-CNF formula

In practice become infeasible at around 200-300 variables

Small Infeasible Instances

Spence '10 proposed formulas based on cardinality constraints

Van Gelder and Spence '10 further developed the construction

Showed experimentally that these formulas harder than other benchmarks

Issued challenge formula for any solver to solve in less than day

Small Infeasible Instances

Spence '10 proposed formulas based on cardinality constraints

Van Gelder and Spence '10 further developed the construction

Showed experimentally that these formulas harder than other benchmarks

Issued challenge formula for any solver to solve in less than day

Our results

Theory: Prove exponential lower bounds

Experiments: Compare theory and practice

Resolution

Resolution — Basis for CDCL solvers

- **Input:** CNF formula F

  $(x \lor \overline{y} \lor z) \land (\overline{y} \lor \overline{z}) \land (x \lor y) \land (\overline{x} \lor \overline{z}) \land (\overline{x} \lor z)$

- **Resolution rule:**

$$\frac{C \lor x \qquad D \lor \overline{x}}{C \lor D}$$

- **Goal:** Proof of unsatisfiability (refutation)
  $=$ Derive empty clause $\bot$

This talk: All formulas unsatisfiable

Example refutation

1.	$x \lor \overline{y} \lor z$	Axiom
2.	$\overline{y} \lor \overline{z}$	Axiom
3.	$x \lor \overline{y}$	Res$(1, 2)$
4.	$x \lor y$	Axiom
5.	x	Res$(3, 4)$
6.	$\overline{x} \lor \overline{z}$	Axiom
7.	$\overline{x} \lor z$	Axiom
8.	$\overline{x}$	Res$(6, 7)$
9.	$\bot$	Res$(5, 8)$

Resolution

Resolution — Basis for CDCL solvers

- **Input:** CNF formula F

  $(x \vee \overline{y} \vee z) \wedge (\overline{y} \vee \overline{z}) \wedge (x \vee y) \wedge (\overline{x} \vee \overline{z}) \wedge (\overline{x} \vee z)$

- **Resolution rule:**

$$\frac{C \vee x \qquad D \vee \overline{x}}{C \vee D}$$

- **Goal:** Proof of unsatisfiability (refutation)
  $=$ Derive empty clause $\perp$

This talk: All formulas unsatisfiable

Refutation length: $\#$ clauses in refutation

Example refutation

1.	$x \vee \overline{y} \vee z$	Axiom
2.	$\overline{y} \vee \overline{z}$	Axiom
3.	$x \vee \overline{y}$	Res$(1,2)$
4.	$x \vee y$	Axiom
5.	x	Res$(3,4)$
6.	$\overline{x} \vee \overline{z}$	Axiom
7.	$\overline{x} \vee z$	Axiom
8.	$\overline{x}$	Res$(6,7)$
9.	$\perp$	Res$(5,8)$

Subset Cardinality Formulas

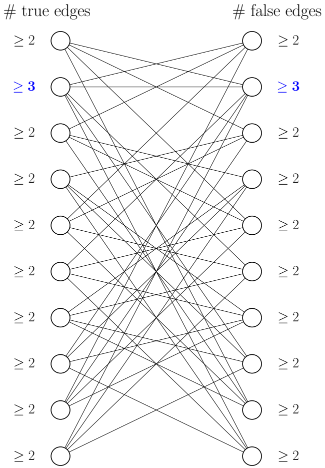
Cardinality constraints on 4-regular bipartite graph with one added edge

Variables: edges in graph

Clauses: vertex cardinality constraints

Left: majority of edges true

Right: majority of edges false



Subset cardinality formula

Subset Cardinality Formulas

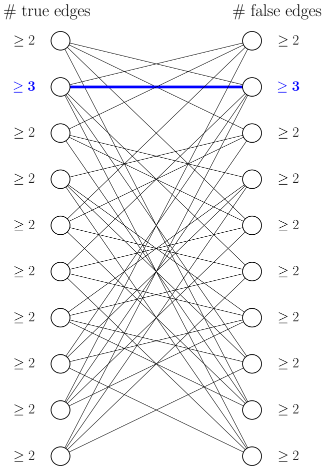
Cardinality constraints on 4-regular bipartite graph with one added edge

Variables: edges in graph

Clauses: vertex cardinality constraints

Left: majority of edges true

Right: majority of edges false



Subset cardinality formula

Subset Cardinality Formulas

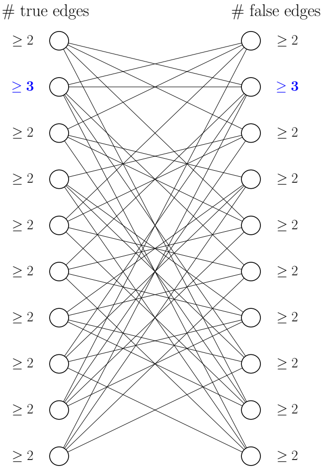
Cardinality constraints on 4-regular bipartite graph with one added edge

Variables: edges in graph

Clauses: vertex cardinality constraints

Left: majority of edges true

Right: majority of edges false



Subset cardinality formula

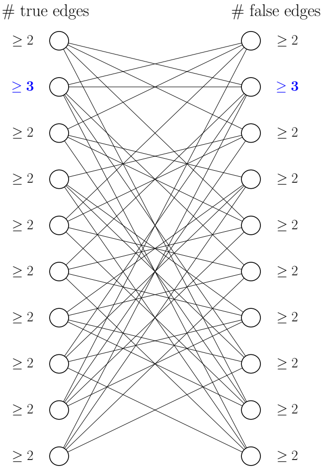
Subset Cardinality Formulas

Cardinality constraints on 4-regular bipartite graph with one added edge

Variables: edges in graph

Clauses: vertex cardinality constraints
  Left: majority of edges true
  Right: majority of edges false

Van Gelder and Spence '10: also require quadrangle-freeness
$\Rightarrow$ Loops of length 4 are forbidden



Subset cardinality formula

Subset Cardinality Formulas

Cardinality constraints on 4-regular bipartite graph with one added edge

Variables: edges in graph

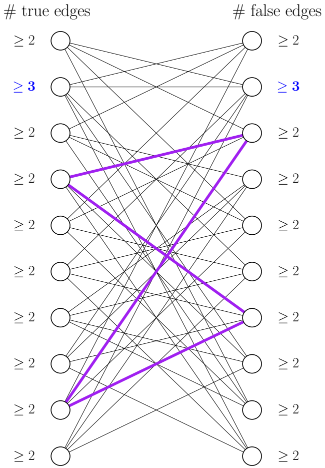
Clauses: vertex cardinality constraints

Left: majority of edges true

Right: majority of edges false

Van Gelder and Spence '10: also require quadrangle-freeness

$\Rightarrow$ Loops of length 4 are forbidden



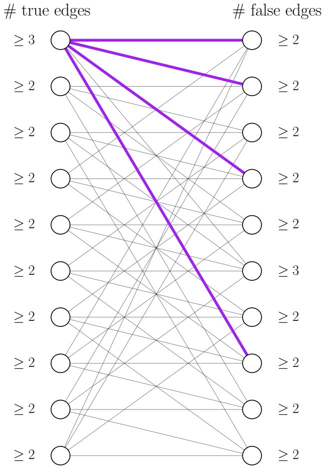
Subset cardinality formula

Fixed Bandwidth Formulas

Fixed bandwidth

Variant of subset cardinality formulas

Neighbors defined by cyclic shifts of original pattern



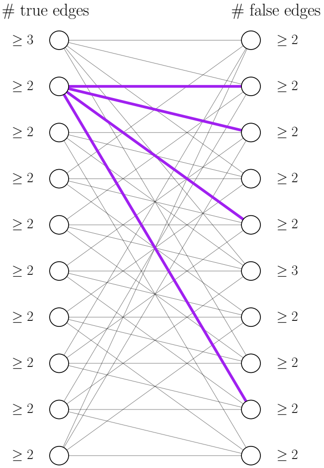
Fixed bandwidth

Fixed Bandwidth Formulas

Fixed bandwidth

Variant of subset cardinality formulas

Neighbors defined by cyclic shifts of original pattern



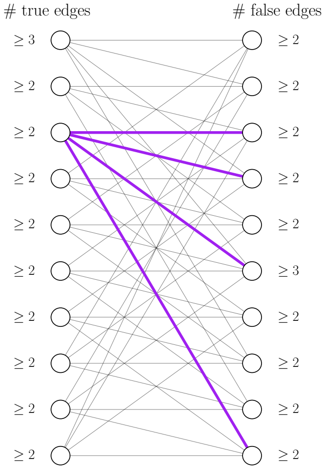
Fixed bandwidth

Fixed Bandwidth Formulas

Fixed bandwidth

Variant of subset cardinality formulas

Neighbors defined by cyclic shifts of original pattern



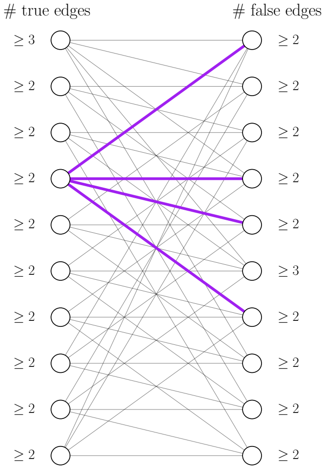
Fixed bandwidth

Fixed Bandwidth Formulas

Fixed bandwidth

Variant of subset cardinality formulas

Neighbors defined by cyclic shifts of original pattern



Fixed bandwidth

Fixed Bandwidth Formulas

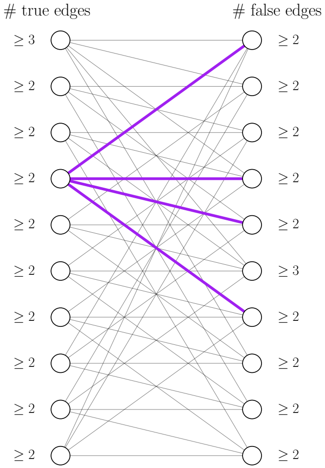
Fixed bandwidth

Variant of subset cardinality formulas

Neighbors defined by cyclic shifts of original pattern

Properties:

- Quadrangle-free
- Polynomial length refutations [Van Gelder and Spence '10]



Fixed bandwidth

Fixed Bandwidth Formulas

Fixed bandwidth

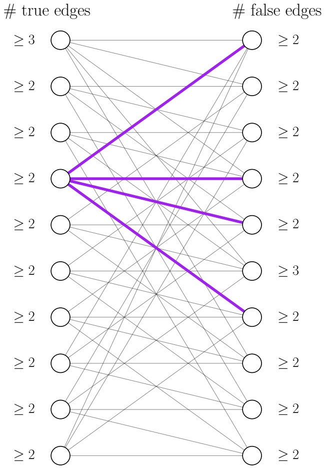
Variant of subset cardinality formulas

Neighbors defined by cyclic shifts of original pattern

Properties:

- Quadrangle-free
- Polynomial length refutations [Van Gelder and Spence '10]

⇒ Quadrangle-freeness not sufficient for lower bounds

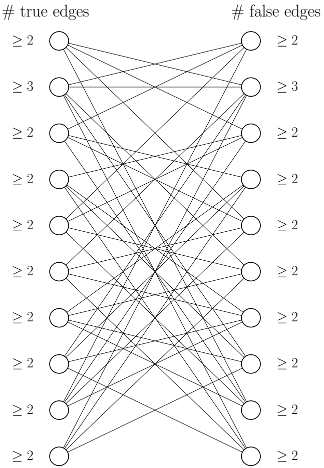


Fixed bandwidth

Expanders Graphs and Subset Cardinality Formulas

Expander graphs are ubiquitous in theoretical computer science

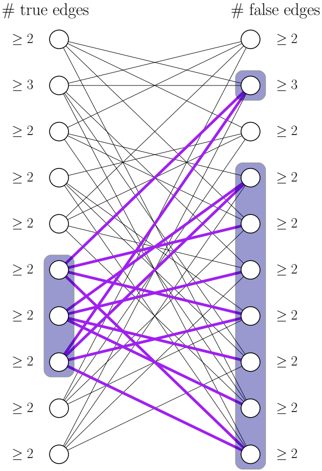
Bipartite expanders:
Subsets of left vertices have many neighbors on right



Expanders Graphs and Subset Cardinality Formulas

Expander graphs are ubiquitous in theoretical computer science

Bipartite expanders:
Subsets of left vertices have many neighbors on right

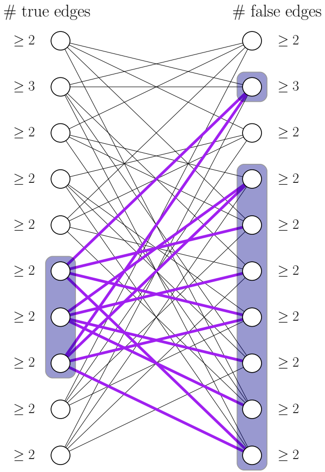


Expanders Graphs and Subset Cardinality Formulas

Expander graphs are ubiquitous in theoretical computer science

Bipartite expanders:
Subsets of left vertices have many neighbors on right

- Vertex neighbors spread out; no discernible pattern
- Underlying graph is expander
  $\Rightarrow$ Exponential lower bound



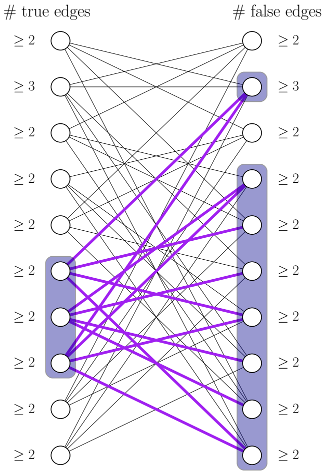
Expanders Graphs and Subset Cardinality Formulas

Expander graphs are ubiquitous in theoretical computer science

Bipartite expanders:
Subsets of left vertices have many neighbors on right

- Vertex neighbors spread out; no discernible pattern
- Underlying graph is expander
  $\Rightarrow$ Exponential lower bound

Fixed bandwidth graphs:
Decent neighbor spread, but deteriorates as size grows



Key Connection

Pigeonhole principle (PHP) formulas

Matching in complete bipartite graph with $n+1$ left and n right vertices

Graph PHP: hard even for non-complete graph if it's expanding

Subset cardinality formulas

Similar flavor to graph pigeonhole principle formulas

Key insight: reduce subset cardinality formulas to graph PHP formulas

Standard Tool: Restrictions

$$(x \vee \overline{y} \vee z) \wedge (\overline{y} \vee \overline{z}) \wedge (x \vee y) \wedge (\overline{x} \vee \overline{z}) \wedge (\overline{x} \vee z)$$

Restriction:

Assign truth value to some variables and simplify formula

True literal: remove clause

False literal: shrink clause

Example refutation

1.	$x \vee \overline{y} \vee z$	Axiom
2.	$\overline{y} \vee \overline{z}$	Axiom
3.	$x \vee \overline{y}$	Res$(1, 2)$
4.	$x \vee y$	Axiom
5.	x	Res$(3, 4)$
6.	$\overline{x} \vee \overline{z}$	Axiom
7.	$\overline{x} \vee z$	Axiom
8.	$\overline{x}$	Res$(6, 7)$
9.	$\perp$	Res$(5, 8)$

Standard Tool: Restrictions

$$(x \vee \overline{y} \vee \top) \wedge (\overline{y} \vee \overline{\top}) \wedge (x \vee y) \wedge (\overline{x} \vee \overline{\top}) \wedge (\overline{x} \vee \top)$$

Restriction:

Assign truth value to some variables and simplify formula

True literal: remove clause

False literal: shrink clause

Example refutation

1.	$x \vee \overline{y} \vee \top$	Axiom
2.	$\overline{y} \vee \overline{\top}$	Axiom
3.	$x \vee \overline{y}$	Res$(1, 2)$
4.	$x \vee y$	Axiom
5.	x	Res$(3, 4)$
6.	$\overline{x} \vee \overline{\top}$	Axiom
7.	$\overline{x} \vee \top$	Axiom
8.	$\overline{x}$	Res$(6, 7)$
9.	$\perp$	Res$(5, 8)$

Standard Tool: Restrictions

$$(\overline{y} \vee \overline{\top}) \wedge (x \vee y) \wedge (\overline{x} \vee \overline{\top})$$

Restriction:

Assign truth value to some variables and simplify formula

True literal: remove clause

False literal: shrink clause

Example refutation

2.	$\overline{y} \vee \overline{\top}$	Axiom
3.	$x \vee \overline{y}$	Res$(1,2)$
4.	$x \vee y$	Axiom
5.	x	Res$(3,4)$
6.	$\overline{x} \vee \overline{\top}$	Axiom
8.	$\overline{x}$	Res$(6,7)$
9.	$\perp$	Res$(5,8)$

Standard Tool: Restrictions

$$\overline{y} \qquad \wedge (x \vee y) \wedge \; \overline{x}$$

Example refutation

Restriction:

Assign truth value to some variables and simplify formula

True literal: remove clause

False literal: shrink clause

2.	$\overline{y}$	Axiom
3.	$x \vee \overline{y}$	Res$(1, 2)$
4.	$x \vee y$	Axiom
5.	x	Res$(3, 4)$
6.	$\overline{x}$	Axiom
8.	$\overline{x}$	Res$(6, 7)$
9.	$\perp$	Res$(5, 8)$

Standard Tool: Restrictions

$$\overline{y} \quad \wedge (x \vee y) \wedge \overline{x}$$

Restriction:

Assign truth value to some variables and simplify formula

True literal: remove clause

False literal: shrink clause

Standard fact:

Restrictions preserve refutations

Lower bound on restricted formula
$\Rightarrow$ Lower bound on original formula

Example refutation

2.	$\overline{y}$	Axiom
3.	$x \vee \overline{y}$	Res$(1,2)$
4.	$x \vee y$	Axiom
5.	x	Res$(3,4)$
6.	$\overline{x}$	Axiom
8.	$\overline{x}$	Res$(6,7)$
9.	$\perp$	Res$(5,8)$

Standard Tool: Restrictions

$$\overline{y} \wedge (x \vee y) \wedge \overline{x}$$

Restriction:

Assign truth value to some variables and simplify formula

True literal: remove clause

False literal: shrink clause

Standard fact:

Restrictions preserve refutations

Lower bound on restricted formula
$\Rightarrow$ Lower bound on original formula

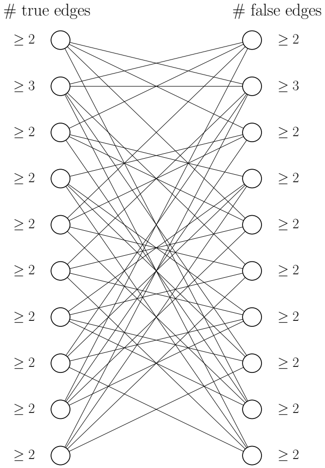
Example refutation

1.	$\overline{y}$	Axiom
2.	$x \vee y$	Axiom
3.	x	Res$(1,2)$
4.	$\overline{x}$	Axiom
5.	$\perp$	Res$(3,4)$

From Subset Cardinality to Pigeonhole Principle Formulas

Restriction:

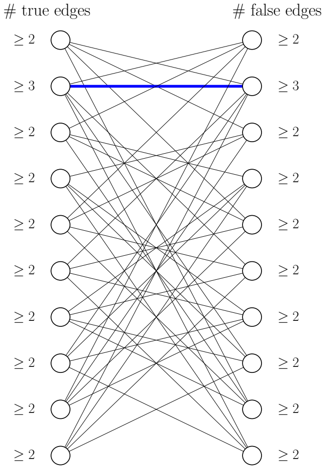
- Take added edge



From Subset Cardinality to Pigeonhole Principle Formulas

Restriction:

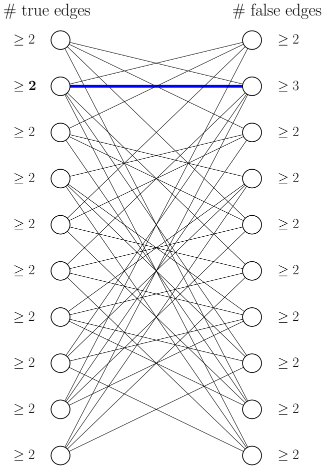
- Take added edge



From Subset Cardinality to Pigeonhole Principle Formulas

Restriction:

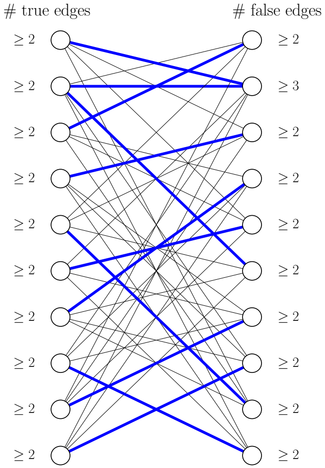
- Take added edge → set to true



From Subset Cardinality to Pigeonhole Principle Formulas

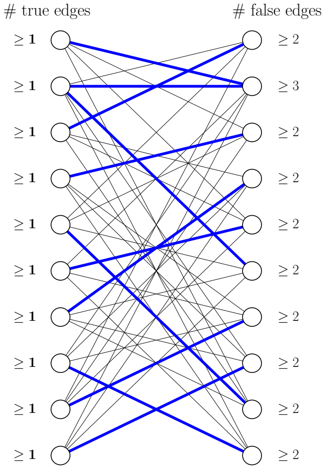
Restriction:

- Take added edge → set to true
- Find matching



Restriction:

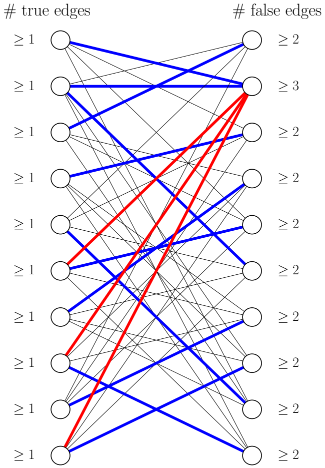
- Take added edge → set to true
- Find matching → set to true



From Subset Cardinality to Pigeonhole Principle Formulas

Restriction:

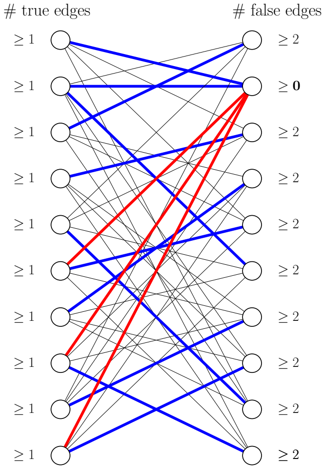
- Take added edge → set to true
- Find matching → set to true
- Edges incident to ≥ 3 vertex on right



From Subset Cardinality to Pigeonhole Principle Formulas

Restriction:

- Take added edge → set to true
- Find matching → set to true
- Edges incident to ≥ 3 vertex on right → set to false

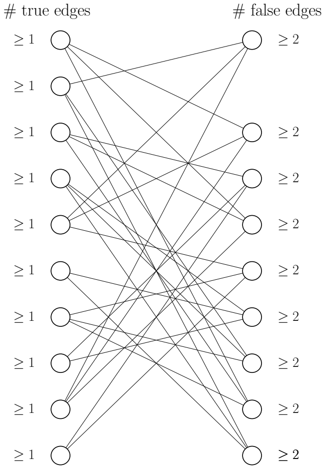


From Subset Cardinality to Pigeonhole Principle Formulas

Restriction:

- Take added edge $\rightarrow$ set to true
- Find matching $\rightarrow$ set to true
- Edges incident to ≥ 3 vertex on right $\rightarrow$ set to false

Restricted formula is graph pigeonhole principle



Graph Pigeonhole Principle Lower Bound

Prove lower bound for resolution and polynomial calculus (extending resolution with algebraic reasoning)

Use techniques in:

[Ben-Sasson and Wigderson '99] for resolution

[Alekhnovich and Razborov '01] for polynomial calculus

Cannot use quite as black box, but need minor technical tweaks

Putting the Pieces Together

- GPHP graph is expander ⇒ Exponential lower bound for GPHP
- Restriction ⇒ Exponential lower bound for subset cardinality formulas
- Random graphs expand almost surely (as size of graph increases)

Theorem

Random instances of **subset cardinality formulas** *are almost surely* **exponentially hard** *to refute in resolution and polynomial calculus.*

Experiments

Formulas:

- Subset cardinality formulas on random 4-regular graphs
- Fixed bandwidth formulas

Compared with:

- Random 3-CNF formulas
- Tseitin formulas on random 3-regular graphs

Equipment:

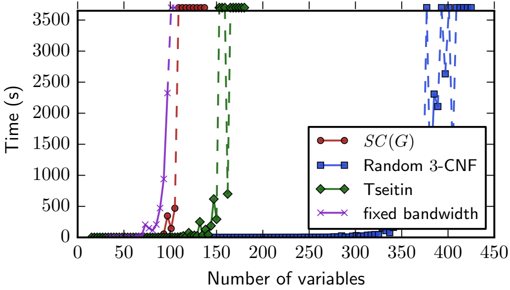
Computer with 2 quad-core AMD Opteron 2.2 Ghz CPUs (2374 HE) and 16 GB of memory

Only one solver running on computer at any given time

Timeout: 1 hour

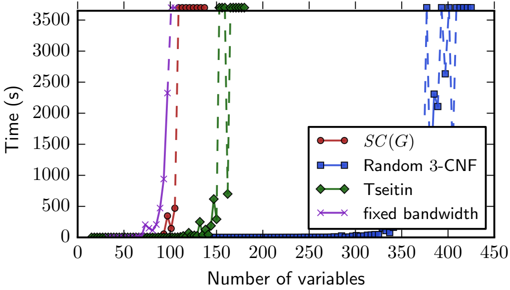
Solvers used: Glucose 2.2, March-rw, Lingeling-ala

Subset Cardinality Formulas Vs. Other Benchmarks



Subset cardinality hardest — Gaussian elimination doesn't help

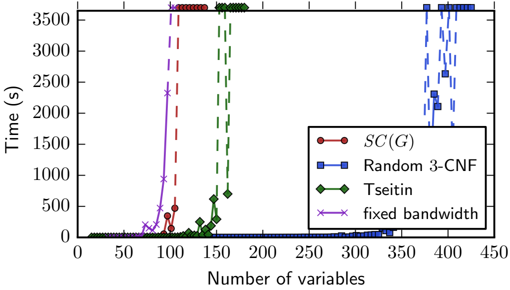
Subset Cardinality Formulas Vs. Other Benchmarks



Subset cardinality hardest — Gaussian elimination doesn't help

Can be easy for pseudo-Boolean solvers: Sat4j solves challenge formula in less than a second [Biere, Le Berre, Lonca, Manthey '14]

Subset Cardinality Formulas Vs. Other Benchmarks



Subset cardinality hardest — Gaussian elimination doesn't help

Can be easy for pseudo-Boolean solvers: Sat4j solves challenge formula in less than a second [Biere, Le Berre, Lonca, Manthey '14]

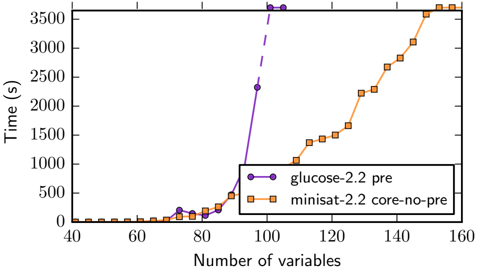
Intriguingly, fixed bandwidth formulas hardest, although easy in theory!

Fixed Bandwidth Formulas Harder than Random Instances

Question: Are hidden constants in polynomial upper bound too large?

Fixed Bandwidth Formulas Harder than Random Instances

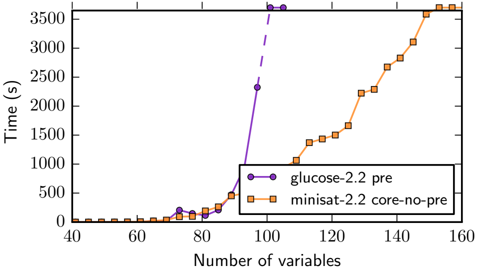
Question: Are hidden constants in polynomial upper bound too large?



Doesn't seem like it. Manually fixed variable ordering: Solver runs faster

Fixed Bandwidth Formulas Harder than Random Instances

Question: Are hidden constants in polynomial upper bound too large?



Doesn't seem like it. Manually fixed variable ordering: Solver runs faster

Maybe expansion is still the cause?
$\Rightarrow$ Higher for fixed bandwidth than random graphs in tested range

Open Problems

Why are fixed bandwidth instances hardest in practice?

1. Still explained by expansion — asymptotics just didn't kick in yet
2. CDCL with current heuristics doesn't fully explore power of resolution

Open Problems

Why are fixed bandwidth instances hardest in practice?

1. Still explained by expansion — asymptotics just didn't kick in yet
2. CDCL with current heuristics doesn't fully explore power of resolution

Find small infeasible formulas for current state-of-art solvers

Challenge formula from [VS '10] solved in 0.5 seconds by Sat4j

Hard formulas for CDCL+Gaussian elimination+Cardinality constraints?

Open Problems

Why are fixed bandwidth instances hardest in practice?

1. Still explained by expansion — asymptotics just didn't kick in yet
2. CDCL with current heuristics doesn't fully explore power of resolution

Find small infeasible formulas for current state-of-art solvers

Challenge formula from [VS '10] solved in 0.5 seconds by Sat4j

Hard formulas for CDCL+Gaussian elimination+Cardinality constraints?

Investigate formulas in [Markström '06]

Prove resolution (or even polynomial calculus) lower bounds

Compare practical hardness with that of subset cardinality formulas

Summary

Our contributions

- Theoretical hardness of subset cardinality formulas
- Experimental comparison with standard benchmarks

Conclusions

- Subset cardinality formulas remain hardest for CDCL solvers
- Tested range: fixed bandwidth formulas harder than random instances

Open problems

- Find small, hard instances even for solvers using algebraic and cardinality reasoning
- Understand power of CDCL with currently used heuristics compared to resolution

Summary

Our contributions

- Theoretical hardness of subset cardinality formulas
- Experimental comparison with standard benchmarks

Conclusions

- Subset cardinality formulas remain hardest for CDCL solvers
- Tested range: fixed bandwidth formulas harder than random instances

Open problems

- Find small, hard instances even for solvers using algebraic and cardinality reasoning
- Understand power of CDCL with currently used heuristics compared to resolution

Thank you for your attention!