A Proof Complexity View of Pseudo-Boolean Solving

Marc Vinyals

Tata Institute of Fundamental Research
Mumbai, India

Joint work with Jan Elffers, Jesús Giráldez-Cru, Stephan Gocht, and Jakob Nordström

Theory and Practice of Satisfiability Solving Workshop
August 28 2018, Casa Matemática Oaxaca, Mexico

The Power of CDCL Solvers

- Current SAT solvers use CDCL algorithm
- Replace heuristics by nondeterminism → CDCL proof system

The Power of CDCL Solvers

- Current SAT solvers use CDCL algorithm
- Replace heuristics by nondeterminism $\rightarrow$ CDCL proof system
- All CDCL proofs are resolution proofs
- Lower bound for resolution length $\Rightarrow$ lower bound for CDCL run time

*(Ignoring preprocessing)

The Power of CDCL Solvers

- Current SAT solvers use CDCL algorithm
- Replace heuristics by nondeterminism $\longrightarrow$ CDCL proof system
- All CDCL proofs are resolution proofs
- Lower bound for resolution length $\Rightarrow$ lower bound for CDCL run time

*(Ignoring preprocessing)

And the opposite direction?

Theorem [Pipatsrisawat, Darwiche '09; Atserias, Fichte, Thurley '09]

CDCL $\equiv_{\text{poly}}$ Resolution

- CDCL can simulate any resolution proof
- Not true for DPLL: limited to tree-like

More Powerful Solvers

Resolution is a weak proof system

- e.g. cannot count
- $x_1 + \cdots + x_n = n/2$ needs exponentially many clauses

More Powerful Solvers

Resolution is a weak proof system

- e.g. cannot count
- $x_1 + \cdots + x_n = n/2$ needs exponentially many clauses

Pseudo-Boolean constraints more expressive

$$x_1 + \cdots + x_n \geq n/2$$
$$\overline{x_1} + \cdots + \overline{x_n} \geq n/2$$

More Powerful Solvers

Resolution is a weak proof system

- e.g. cannot count
- $x_1 + \cdots + x_n = n/2$ needs exponentially many clauses

Pseudo-Boolean constraints more expressive

$$x_1 + \cdots + x_n \geq n/2$$
$$\overline{x_1} + \cdots + \overline{x_n} \geq n/2$$

Build solvers with native pseudo-Boolean constraints?

- Can generalize CDCL, even if tricky
- Not as successful as SAT solvers

What do we do

Question

What limits pseudo-Boolean solvers?

What do we do

Question

What limits pseudo-Boolean solvers?

Theoretical Barriers

- Study proof systems arising from pseudo-Boolean solvers

Implementation

- Evaluate solvers on theoretically easy formulas

What do we do

Question

What limits pseudo-Boolean solvers?

Theoretical Barriers

- Study proof systems arising from pseudo-Boolean solvers

Implementation

- Evaluate solvers on theoretically easy formulas

Cutting Planes

All pseudo-Boolean proofs are cutting planes proofs

Cutting Planes

All pseudo-Boolean proofs are cutting planes proofs

Work with linear pseudo-Boolean inequalities

$$x \vee \bar{y} \quad \rightarrow \quad x + \bar{y} \geq 1 \quad \equiv \quad x + (1 - y) \geq 1$$

$\bar{y} = 1 - y$ degree

Cutting Planes

All pseudo-Boolean proofs are cutting planes proofs

Work with linear pseudo-Boolean inequalities

$$x \vee \bar{y} \quad \rightarrow \quad x + \bar{y} \geq 1 \quad \equiv \quad x + (1 - y) \geq 1$$

$\bar{y} = 1 - y$ degree

Rules

Variable axioms

$$\frac{}{x \geq 0} \qquad \frac{}{-x \geq -1}$$

Addition

$$\frac{\sum a_i x_i \geq a \qquad \sum b_i x_i \geq b}{\sum (\alpha a_i + \beta b_i) x_i \geq \alpha a + \beta b}$$

Division

$$\frac{\sum a_i x_i \geq a}{\sum (a_i / k) x_i \geq \lceil a/k \rceil}$$

Goal: derive $0 \geq 1$

Addition in Practice

Addition

$$\frac{\sum a_i x_i \geq a \qquad \sum b_i x_i \geq b}{\sum (\alpha a_i + \beta b_i) x_i \geq \alpha a + \beta b}$$

- Unbounded choices
- Need a reason to add inequalities

Division in Practice

Division

$$\frac{\sum a_i x_i \geq a}{\sum (a_i/k) x_i \geq \lceil a/k \rceil}$$

- Too expensive

Weaker Rules

What is the bare minimum to simulate resolution?

$$\frac{x \vee y \vee \bar{z} \qquad \bar{x} \vee y}{y \vee \bar{z}}$$

Weaker Rules

What is the bare minimum to simulate resolution?

$$\frac{x \vee y \vee \bar{z} \qquad \bar{x} \vee y}{y \vee \bar{z}}$$

$$\frac{x + y + \bar{z} \geq 1 \qquad \bar{x} + y \geq 1}{x + \bar{x} + 2y + \bar{z} \geq 2}$$

Weaker Rules

What is the bare minimum to simulate resolution?

$$\frac{x \lor y \lor \bar{z} \qquad \bar{x} \lor y}{y \lor \bar{z}} \qquad\qquad \frac{x + y + \bar{z} \geq 1 \qquad \bar{x} + y \geq 1}{\ast + 2y + \bar{z} \geq 1}$$

- Addition only if some variable cancels

Weaker Rules

What is the bare minimum to simulate resolution?

$$\frac{x \vee y \vee \overline{z} \qquad \overline{x} \vee y}{y \vee \overline{z}}$$

$$\cfrac{\cfrac{x + y + \overline{z} \geq 1 \qquad \overline{x} + y \geq 1}{2y + \overline{z} \geq 1}}{y + \overline{z} \geq 1}$$

- Addition only if some variable cancels
- Division brings coefficients down to degree

Addition in Practice

Addition

$$\frac{\sum a_i x_i \geq a \qquad \sum b_i x_i \geq b}{\sum (\alpha a_i + \beta b_i) x_i \geq \alpha a + \beta b}$$

- Unbounded choices
- Need a reason to add inequalities

Cancelling Addition

- Some variable cancels: $\alpha a_i + \beta b_i = 0$
- aka. Generalized Resolution

Division in Practice

Division

$$\frac{\sum a_i x_i \geq a}{\sum (a_i/k) x_i \geq \lceil a/k \rceil}$$

- Too expensive

Saturation

$$\frac{\sum a_i x_i \geq a}{\sum \min(a, a_i) x_i \geq a}$$

Proof Systems

Power of subsystems of CP?

CP saturation general addition

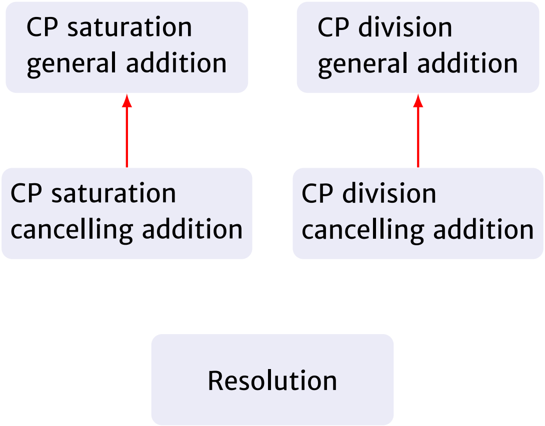
CP division general addition

CP saturation cancelling addition

CP division cancelling addition

Resolution

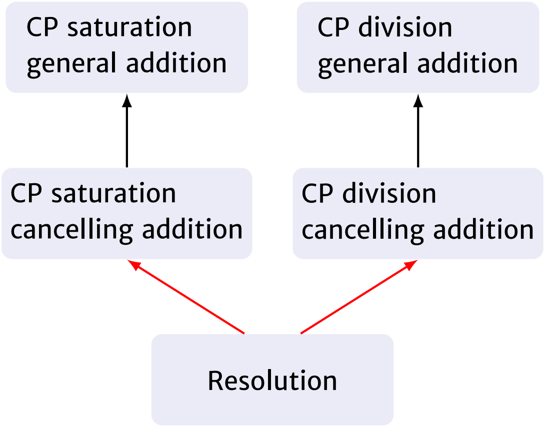
Proof Systems



Cancelling addition is a particular case of addition

$A \longrightarrow B$: B simulates A (with only polynomial loss)

Proof Systems

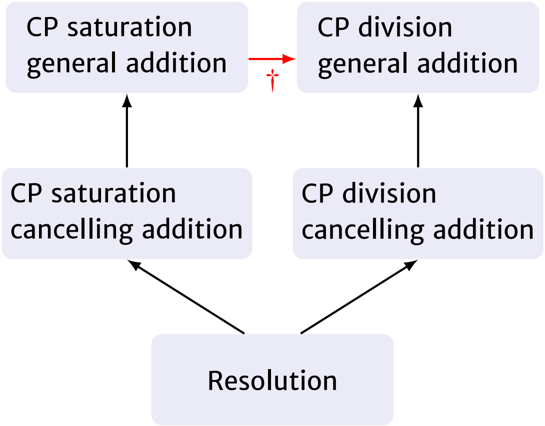


All subsystems simulate resolution

- Trivial over CNF inputs
- Also holds over linear pseudo-Boolean inputs

$A \longrightarrow B$: B simulates A (with only polynomial loss)

Proof Systems



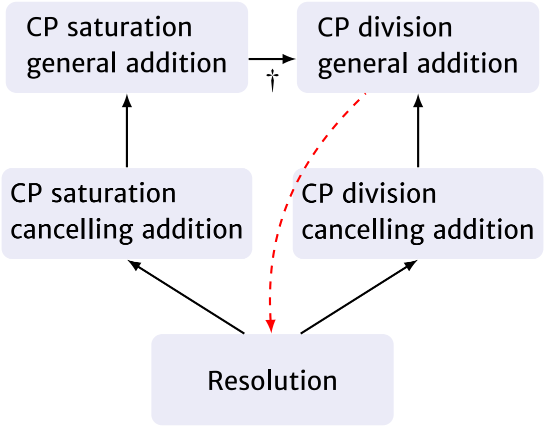
Repeated divisions simulate saturation

- Polynomial simulation only if polynomial coefficients

$A \longrightarrow B$: B simulates A (with only polynomial loss)

†: known only for polynomial-size coefficients

Proof Systems



CP stronger than resolution

- Pigeonhole principle
- Subset cardinality

have proofs of size

- polynomial in CP
- exponential in resolution

$A \longrightarrow B$: B simulates A (with only polynomial loss)
$A \dashrightarrow B$: B cannot simulate A (separation)

†: known only for polynomial-size coefficients

Bad News

Theorem

On CNF inputs all subsystems as weak as resolution

- No subsystem is implicationally complete
- Solvers very sensitive to input encoding

Cancelling Addition $\equiv$ Resolution

Observation [Hooker '88]

Over CNF inputs CP with cancelling addition $\equiv$ resolution.

Cancelling Addition ≡ Resolution

Observation [Hooker '88]

Over CNF inputs CP with cancelling addition ≡ resolution.

Proof Sketch

- Start with clauses (degree 1)
- Add two clauses → a clause

$$\frac{x + \sum y_i \geq 1 \qquad \bar{x} + \sum y_i \geq 1}{\ast + 1 + \sum y_i \geq 1 + 1}$$

Cancelling Addition ≡ Resolution

Observation [Hooker '88]

Over CNF inputs CP with cancelling addition ≡ resolution.

Proof Sketch

- Start with clauses (degree 1)
- Add two clauses → a clause

$$\frac{x + \sum y_i \geq 1 \qquad \bar{x} + \sum y_i \geq 1}{\sum y_i \geq 1}$$

Cancelling Addition ≡ Resolution

Observation [Hooker '88]

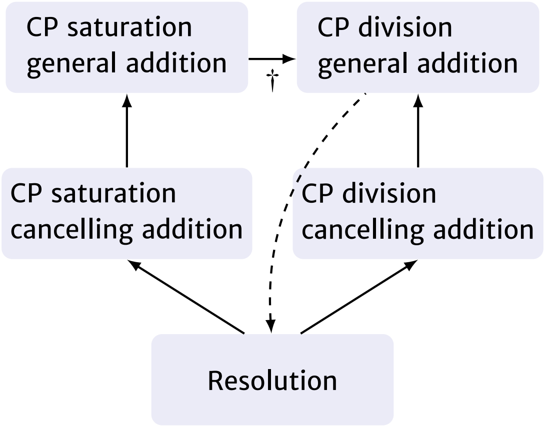
Over CNF inputs CP with cancelling addition ≡ resolution.

Proof Sketch

- Start with clauses (degree 1)
- Add two clauses → a clause

$$\frac{x + \sum y_i \geq 1 \qquad \bar{x} + \sum y_i \geq 1}{\sum y_i \geq 1} \equiv \frac{x \vee C \qquad \bar{x} \vee D}{C \vee D}$$

Proof Systems



Cancellation ≡ Resolution

- Over CNF inputs

[Hooker '88]

- Pigeonhole principle
- Subset cardinality

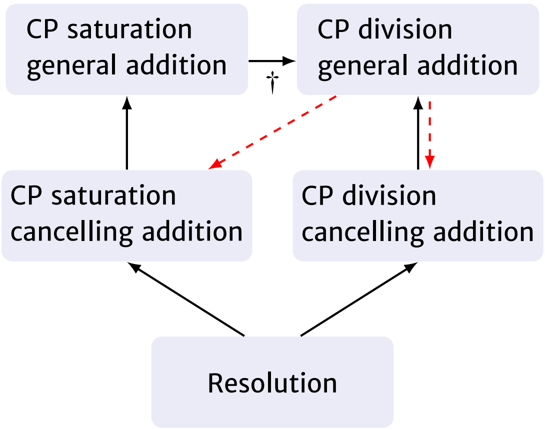
have proofs of size

- polynomial in CP
- exponential in resolution

$A \longrightarrow B$: B simulates A (with only polynomial loss)
$A \dashrightarrow B$: B cannot simulate A (separation)

†: known only for polynomial-size coefficients

Proof Systems



Cancellation ≡ Resolution

- Over CNF inputs

[Hooker '88]

- Pigeonhole principle
- Subset cardinality

have proofs of size

- polynomial in CP
- exponential in CP
  with cancelling addition
  and any rounding

$A \longrightarrow B$: B simulates A (with only polynomial loss)
$A \dashrightarrow B$: B cannot simulate A (separation)

†: known only for polynomial-size coefficients

Saturation: General Addition $\equiv$ Cancelling Addition

Theorem

Over CNF inputs CP with saturation $\equiv$ CP with cancelling addition.

Saturation: General Addition $\equiv$ Cancelling Addition

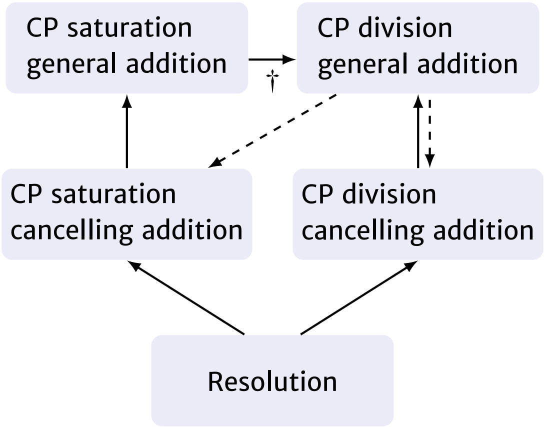
Theorem

Over CNF inputs CP with saturation $\equiv$ CP with cancelling addition.

Corollary

Over CNF inputs CP with saturation $\equiv$ resolution.

Proof Systems



Saturation ≡ Resolution

- Over CNF inputs

- Pigeonhole principle
- Subset cardinality

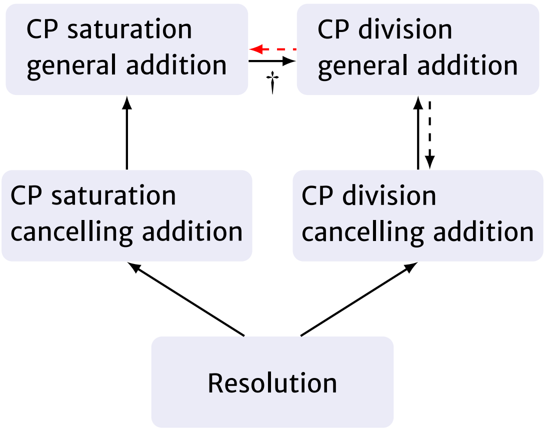
have proofs of size

- polynomial in CP
- exponential in CP
  with cancelling addition

$A \longrightarrow B$: B simulates A (with only polynomial loss)
$A \dashrightarrow B$: B cannot simulate A (separation)

†: known only for polynomial-size coefficients

Proof Systems



Saturation ≡ Resolution

- Over CNF inputs

- Pigeonhole principle
- Subset cardinality

have proofs of size

- polynomial in CP
- exponential in CP with cancelling addition or saturation

$A \longrightarrow B$: B simulates A (with only polynomial loss)
$A \dashrightarrow B$: B cannot simulate A (separation)

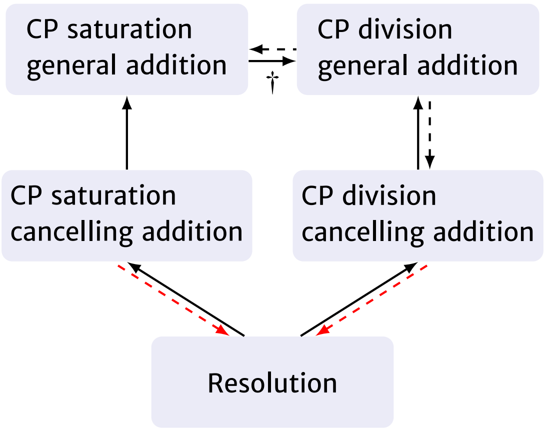
†: known only for polynomial-size coefficients

Linear Programming is Easy

Lemma

If a formula defines an empty polytope over $\mathbb{R}$
then have polynomial size proof in CP with cancelling addition.

Proof Systems



Pseudo-Boolean versions of

- Pigeonhole principle
- Subset cardinality

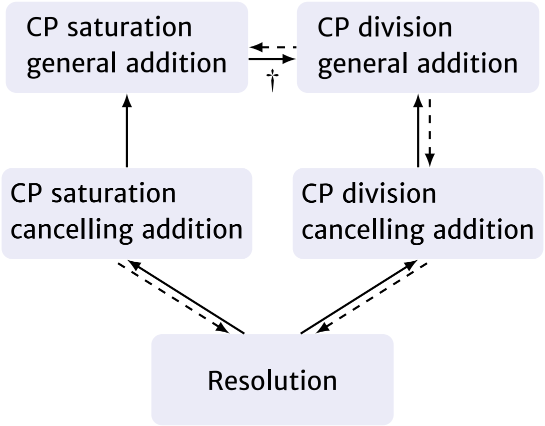
have proof of size

- polynomial in all CP subsystems
- exponential in resolution

$A \longrightarrow B$: B simulates A (with only polynomial loss)
$A \dashrightarrow B$: B cannot simulate A (separation)

†: known only for polynomial-size coefficients

Proof Systems



$A \longrightarrow B$: B simulates A (with only polynomial loss)
$A \dashrightarrow B$: B cannot simulate A (separation)

†: known only for polynomial-size coefficients

Pseudo-Boolean versions of

- Pigeonhole principle
- Subset cardinality

have proof of size

- polynomial in all CP subsystems
- exponential in resolution

CNF version exponential ⇒
Cannot recover encoding ⇒
Subsystems are incomplete

What do we do

Question

What limits pseudo-Boolean solvers?

Theoretical Barriers

- Study proof systems arising from pseudo-Boolean solvers
- Cancelling addition and saturation not enough

Implementation

- Evaluate solvers on theoretically easy formulas

What do we do

Question

What limits pseudo-Boolean solvers?

Theoretical Barriers

- Study proof systems arising from pseudo-Boolean solvers
- Cancelling addition and saturation not enough

Implementation

- Evaluate solvers on theoretically easy formulas

Easy Formulas

Craft combinatorial formulas easy for CP

- Build proofs in different subsystems
- Choice of parameters for different levels of hardness
  1. Easy for resolution
  2. Hard for resolution, infeasible LP
  3. Feasible LP, easy for saturation
  4. Require division?
- Easy for CP, even tree-like

Solvers

Solvers from PB evaluation 2016 with different techniques

- *Open-WBO*
  - Translate into CNF
  - $\simeq$ Resolution

Solvers

Solvers from PB evaluation 2016 with different techniques

- *Open-WBO*
  - Translate into CNF
  - $\simeq$ Resolution
- *Sat4j*
  - Linear inequalities
  - $\simeq$ CP saturation cancelling addition
- *RoundingSat*
  - Linear inequalities
  - $\lesssim$ CP division cancelling addition

Experimental Results

Experimental Observation

PB solvers not good at proof search.

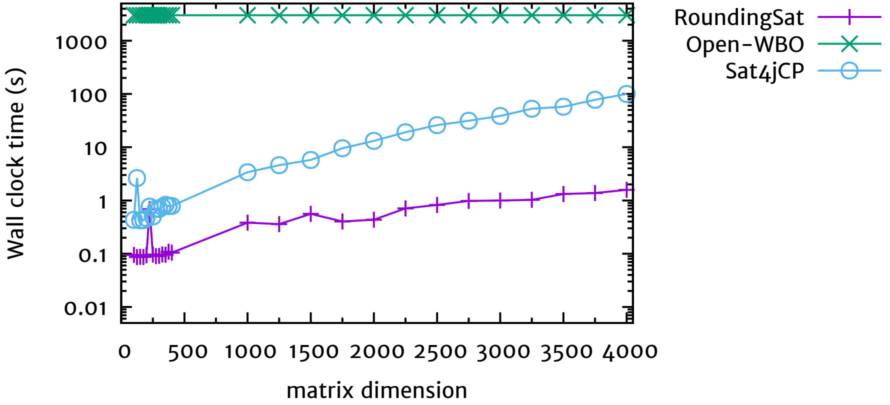
Experimental Results

Experimental Observation

PB solvers not good at proof search.

- Sometimes exponentially faster than CDCL
  - e.g. when LP close to infeasible

Experiments: SC (subset cardinality), random graphs



- No rational solutions
- Exponentially hard for resolution $\Rightarrow$ *Open-WBO* times out
- Cutting planes solvers run fast

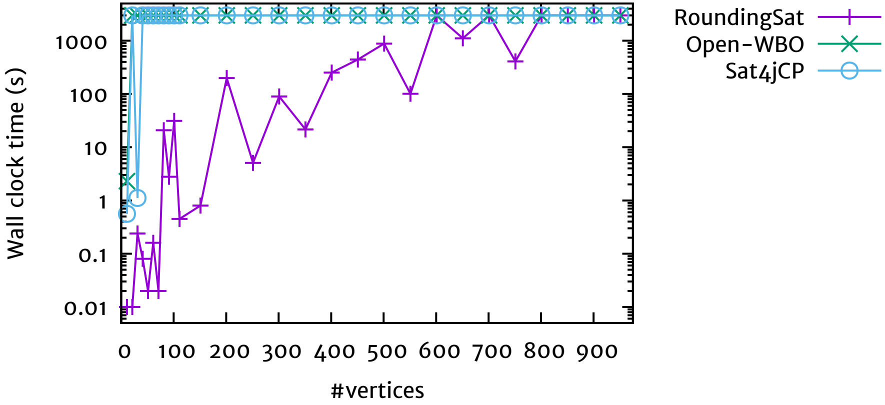
Experimental Results

Experimental Observation

PB solvers not good at proof search.

- Sometimes exponentially faster than CDCL
  - e.g. when LP close to infeasible
- Often not good at “truly Boolean” reasoning
  - in particular when division useful

Experiments: EC (even colouring), random graphs



- Provably hard for resolution ⇒ *Open-WBO* times out
- Conjecture hard for CP with saturation ⇒ *Sat4j* times out
- *RoundingSat* works best ⇒ division necessary?

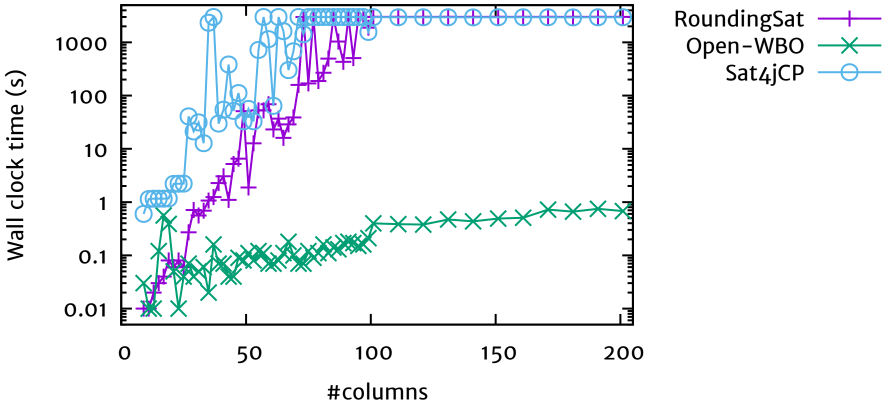
Experimental Results

Experimental Observation

PB solvers not good at proof search.

- Sometimes exponentially faster than CDCL
  - e.g. when LP close to infeasible
- Often not good at “truly Boolean” reasoning
  - in particular when division useful
- Sometimes even not good for infeasible LPs

Experiments: VC (vertex cover), grid, no-rational



- No rational solutions
- *Open-WBO* runs fast
- Cutting planes solvers fairly bad

What do we do

Question

What limits pseudo-Boolean solvers?

Theoretical Barriers

- Study proof systems arising from pseudo-Boolean solvers
- Cancelling addition and saturation not enough

Implementation

- Evaluate solvers on theoretically easy formulas
- Need to improve on proof search

Take Home

Remarks

- Classified subsystems of Cutting Planes
- On CNF Subsystems $\equiv$ Resolution $\rightarrow$ Sensitive to encoding
- Solvers not good at proof search

Take Home

Remarks

- Classified subsystems of Cutting Planes
- On CNF Subsystems $\equiv$ Resolution $\rightarrow$ Sensitive to encoding
- Solvers not good at proof search

Open problems

- Is division needed? Separation on PB inputs?
- Better search heuristics

Take Home

Remarks

- Classified subsystems of Cutting Planes
- On CNF Subsystems $\equiv$ Resolution $\rightarrow$ Sensitive to encoding
- Solvers not good at proof search

Open problems

- Is division needed? Separation on PB inputs?
- Better search heuristics

Thanks!