On Division Versus Saturation in Cutting Planes

Jakob Nordström

KTH Royal Institute of Technology
Stockholm, Sweden

Bringing CP, SAT and SMT together:
Next Challenges in Constraint Solving
Schloss Dagstuhl – Leibniz Center for Informatics
February 5, 2019

Joint work with Stephan Gocht and Amir Yehudayoff

SAT in Theory and Practice

Computational complexity

- Satisfiability fundamental problem in theoretical computer science
- SAT canonical NP-complete problem [Coo71, Lev73]
- Hence totally intractable in worst case (probably)
- One of the million dollar "Millennium Problems"

SAT in Theory and Practice

Computational complexity

- Satisfiability fundamental problem in theoretical computer science
- SAT canonical NP-complete problem [Coo71, Lev73]
- Hence totally intractable in worst case (probably)
- One of the million dollar "Millennium Problems"

SAT solving

- Enormous progress in performance last 15–20 years
- State-of-the-art solvers can deal with real-world instances with millions of variables
- Used for hardware & software verification, OR, AI, ...
- But also exist tiny formulas that are totally beyond reach

SAT in Theory and Practice

Computational complexity

- Satisfiability fundamental problem in theoretical computer science
- SAT canonical NP-complete problem [Coo71, Lev73]
- Hence totally intractable in worst case (probably)
- One of the million dollar "Millennium Problems"

SAT solving

- Enormous progress in performance last 15–20 years
- State-of-the-art solvers can deal with real-world instances with millions of variables
- Used for hardware & software verification, OR, AI, ...
- But also exist tiny formulas that are totally beyond reach

Limitations of CDCL

1. Clauses weak formalism for encoding constraints
2. Also weak method of reasoning (resolution)

Pseudo-Boolean Reasoning (a.k.a. 0-1 Linear Programming)

- Pseudo-Boolean (PB) linear constraints are stronger than clauses

Compare

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 3$$

with

$$\begin{aligned} &(x_1 \lor x_2 \lor x_3 \lor x_4) \land (x_1 \lor x_2 \lor x_3 \lor x_5) \land (x_1 \lor x_2 \lor x_3 \lor x_6) \\ \land &(x_1 \lor x_2 \lor x_4 \lor x_5) \land (x_1 \lor x_2 \lor x_4 \lor x_6) \land (x_1 \lor x_2 \lor x_5 \lor x_6) \\ \land &(x_1 \lor x_3 \lor x_4 \lor x_5) \land (x_1 \lor x_3 \lor x_4 \lor x_6) \land (x_1 \lor x_3 \lor x_5 \lor x_6) \\ \land &(x_1 \lor x_4 \lor x_5 \lor x_6) \land (x_2 \lor x_3 \lor x_4 \lor x_5) \land (x_2 \lor x_3 \lor x_4 \lor x_6) \\ \land &(x_2 \lor x_3 \lor x_5 \lor x_6) \land (x_2 \lor x_4 \lor x_5 \lor x_6) \land (x_3 \lor x_4 \lor x_5 \lor x_6) \end{aligned}$$

Pseudo-Boolean Reasoning (a.k.a. 0-1 Linear Programming)

- Pseudo-Boolean (PB) linear constraints are stronger than clauses

Compare

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 3$$

with

$$\begin{aligned}
&(x_1 \lor x_2 \lor x_3 \lor x_4) \land (x_1 \lor x_2 \lor x_3 \lor x_5) \land (x_1 \lor x_2 \lor x_3 \lor x_6) \\
\land &(x_1 \lor x_2 \lor x_4 \lor x_5) \land (x_1 \lor x_2 \lor x_4 \lor x_6) \land (x_1 \lor x_2 \lor x_5 \lor x_6) \\
\land &(x_1 \lor x_3 \lor x_4 \lor x_5) \land (x_1 \lor x_3 \lor x_4 \lor x_6) \land (x_1 \lor x_3 \lor x_5 \lor x_6) \\
\land &(x_1 \lor x_4 \lor x_5 \lor x_6) \land (x_2 \lor x_3 \lor x_4 \lor x_5) \land (x_2 \lor x_3 \lor x_4 \lor x_6) \\
\land &(x_2 \lor x_3 \lor x_5 \lor x_6) \land (x_2 \lor x_4 \lor x_5 \lor x_6) \land (x_3 \lor x_4 \lor x_5 \lor x_6)
\end{aligned}$$

- And pseudo-Boolean reasoning exponentially more powerful in theory ("0-1 integer linear programming with learning")

Pseudo-Boolean Reasoning (a.k.a. 0-1 Linear Programming)

- Pseudo-Boolean (PB) linear constraints are stronger than clauses

Compare

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 3$$

with

$$\begin{aligned}
&(x_1 \lor x_2 \lor x_3 \lor x_4) \land (x_1 \lor x_2 \lor x_3 \lor x_5) \land (x_1 \lor x_2 \lor x_3 \lor x_6) \\
\land&(x_1 \lor x_2 \lor x_4 \lor x_5) \land (x_1 \lor x_2 \lor x_4 \lor x_6) \land (x_1 \lor x_2 \lor x_5 \lor x_6) \\
\land&(x_1 \lor x_3 \lor x_4 \lor x_5) \land (x_1 \lor x_3 \lor x_4 \lor x_6) \land (x_1 \lor x_3 \lor x_5 \lor x_6) \\
\land&(x_1 \lor x_4 \lor x_5 \lor x_6) \land (x_2 \lor x_3 \lor x_4 \lor x_5) \land (x_2 \lor x_3 \lor x_4 \lor x_6) \\
\land&(x_2 \lor x_3 \lor x_5 \lor x_6) \land (x_2 \lor x_4 \lor x_5 \lor x_6) \land (x_3 \lor x_4 \lor x_5 \lor x_6)
\end{aligned}$$

- And pseudo-Boolean reasoning exponentially more powerful in theory ("0-1 integer linear programming with learning")
- But PB solvers less efficient than CDCL in practice!?

Our Work

- Study pseudo-Boolean rules of reasoning used in practice
- How do they compare to cutting planes proof system?
- In particular, what is the power of division versus saturation?

Pseudo-Boolean Constraints and Normalized Form

In this talk, "pseudo-Boolean" (PB) refers to 0-1 integer linear constraints

Pseudo-Boolean Constraints and Normalized Form

In this talk, "pseudo-Boolean" (PB) refers to 0-1 integer linear constraints

Convenient to use non-negative linear combinations of literals, a.k.a. normalized form

$$\sum_i a_i \ell_i \geq A$$

- coefficients a_i: non-negative integers
- degree (of falsity) A: positive integer
- literals ℓ_i: x_i or $\overline{x}_i$ (where $x_i + \overline{x}_i = 1$)

(In what follows, all constraints assumed to be implicitly normalized)

Some Types of Pseudo-Boolean Constraints

1. Clauses are pseudo-Boolean constraints

$$x \vee \overline{y} \vee z \quad \Leftrightarrow \quad x + \overline{y} + z \geq 1$$

Refer to collection of such constraints as "CNF formula"

Some Types of Pseudo-Boolean Constraints

1. Clauses are pseudo-Boolean constraints

$$x \vee \overline{y} \vee z \quad \Leftrightarrow \quad x + \overline{y} + z \geq 1$$

Refer to collection of such constraints as "CNF formula"

2. Cardinality constraints

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 3$$

Some Types of Pseudo-Boolean Constraints

1. Clauses are pseudo-Boolean constraints

$$x \vee \overline{y} \vee z \quad \Leftrightarrow \quad x + \overline{y} + z \geq 1$$

Refer to collection of such constraints as "CNF formula"

2. Cardinality constraints

$$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 3$$

3. General constraints

$$x_1 + 2\overline{x}_2 + 3x_3 + 4\overline{x}_4 + 5x_5 \geq 7$$

Approaches to Pseudo-Boolean Solving

Conversion to disjunctive clauses

- Lazy approach: learn clauses from PB constraints
  - *Sat4j* [LP10] (one of versions in library)
- Eager approach: re-encode to clauses and run CDCL
  - *MiniSat+* [ES06]
  - *Open-WBO* [MML14]
  - *NaPS* [SN15]

Approaches to Pseudo-Boolean Solving

Conversion to disjunctive clauses

- Lazy approach: learn clauses from PB constraints
  - *Sat4j* [LP10] (one of versions in library)
- Eager approach: re-encode to clauses and run CDCL
  - *MiniSat+* [ES06]
  - *Open-WBO* [MML14]
  - *NaPS* [SN15]

Native reasoning with pseudo-Boolean constraints

- *PRS* [DG02]
- *Galena* [CK05]
- *Pueblo* [SS06]
- *Sat4j* [LP10]
- *RoundingSat* [EN18]

Approaches to Pseudo-Boolean Solving

Conversion to disjunctive clauses

- Lazy approach: learn clauses from PB constraints
  - *Sat4j* [LP10] (one of versions in library)
- Eager approach: re-encode to clauses and run CDCL
  - *MiniSat+* [ES06]
  - *Open-WBO* [MML14]
  - *NaPS* [SN15]

Native reasoning with pseudo-Boolean constraints

- *PRS* [DG02]
- *Galena* [CK05]
- *Pueblo* [SS06]
- *Sat4j* [LP10]
- *RoundingSat* [EN18]

The Cutting Planes Proof System [CCT87]

Literal axioms $$\frac{}{\ell_i \geq 0}$$

Linear combination $$\frac{\sum_i a_i \ell_i \geq A \qquad \sum_i b_i \ell_i \geq B}{\sum_i (c_A a_i + c_B b_i)\ell_i \geq c_A A + c_B B} \quad [c_A, c_B \geq 0]$$

Division $$\frac{\sum_i a_i \ell_i \geq A}{\sum_i \lceil a_i / c \rceil \ell_i \geq \lceil A/c \rceil} \quad [c > 0]$$

The Cutting Planes Proof System [CCT87]

Literal axioms $$\frac{}{\ell_i \geq 0}$$

Linear combination $$\frac{\sum_i a_i \ell_i \geq A \qquad \sum_i b_i \ell_i \geq B}{\sum_i (c_A a_i + c_B b_i) \ell_i \geq c_A A + c_B B} \quad [c_A, c_B \geq 0]$$

Division $$\frac{\sum_i a_i \ell_i \geq A}{\sum_i \lceil a_i / c \rceil \ell_i \geq \lceil A/c \rceil} \quad [c > 0]$$

Setting in this talk

Input: Set of pseudo-Boolean constraints without 0-1 solution

Goal: Prove unsatisfiability by deriving $0 \geq 1$ using cutting planes

The Cutting Planes Proof System [CCT87]

Literal axioms $$\frac{}{\ell_i \geq 0}$$

Linear combination $$\frac{\sum_i a_i\ell_i \geq A \qquad \sum_i b_i\ell_i \geq B}{\sum_i (c_A a_i + c_B b_i)\ell_i \geq c_A A + c_B B} \quad [c_A, c_B \geq 0]$$

Division $$\frac{\sum_i a_i\ell_i \geq A}{\sum_i \lceil a_i/c \rceil \ell_i \geq \lceil A/c \rceil} \quad [c > 0]$$

Setting in this talk

Input: Set of pseudo-Boolean constraints without 0-1 solution

Goal: Prove unsatisfiability by deriving $0 \geq 1$ using cutting planes

Ignore algorithmic aspects — heuristics beyond rigorous analysis — and assume optimal use of derivation rules

More About Cutting Planes

A toy example:

$$\frac{6x + 2y + 3z \geq 5 \qquad x + 2y + w \geq 1}{(6x + 2y + 3z) + 2(x + 2y + w) \geq 5 + 2 \cdot 1} \text{ Linear combination}$$

More About Cutting Planes

A toy example:

$$\frac{6x + 2y + 3z \geq 5 \qquad x + 2y + w \geq 1}{8x + 6y + 3z + 2w \geq 7} \text{ Linear combination}$$

More About Cutting Planes

A toy example:

$$\frac{\dfrac{6x + 2y + 3z \geq 5 \qquad x + 2y + w \geq 1}{8x + 6y + 3z + 2w \geq 7}\ \text{Linear combination}}{3x + 2y + z + w \geq 3}\ \text{Division}$$

More About Cutting Planes

A toy example:

$$\dfrac{\dfrac{6x + 2y + 3z \geq 5 \qquad x + 2y + w \geq 1}{8x + 6y + 3z + 2w \geq 7}\ \text{Linear combination}}{3x + 2y + z + w \geq 3}\ \text{Division}$$

- Literal axioms and linear combinations sound also over the reals
- Division is where the power of cutting planes lies
- Exponentially stronger than resolution/CDCL [Hak85, CCT87]

Generalized Resolution

In conflict-driven search, linear combination always made to cancel variable (on which constraints disagree)

Generalized resolution rule [Hoo88, Hoo92]

$$\frac{a_j x_j + \sum_{i \neq j} a_i \ell_i \geq A \qquad b_j \overline{x}_j + \sum_{i \neq j} b_i \ell_i \geq B}{\sum_{i \neq j} \left(\frac{c}{a_j} a_i + \frac{c}{b_j} b_i\right) \ell_i \geq \frac{c}{a_j} A + \frac{c}{b_j} B - c} \quad [c = \mathrm{lcm}(a_j, b_j)]$$

Generalized Resolution

In conflict-driven search, linear combination always made to cancel variable (on which constraints disagree)

Generalized resolution rule [Hoo88, Hoo92]

$$\frac{a_j x_j + \sum_{i \neq j} a_i \ell_i \geq A \qquad b_j \overline{x}_j + \sum_{i \neq j} b_i \ell_i \geq B}{\sum_{i \neq j} \left(\frac{c}{a_j} a_i + \frac{c}{b_j} b_i\right) \ell_i \geq \frac{c}{a_j} A + \frac{c}{b_j} B - c} \quad [c = \mathrm{lcm}(a_j, b_j)]$$

Another toy example:

$$\frac{2x + y + z \geq 2 \qquad 3\overline{x} + 2y + u + w \geq 3}{3(y+z) + 2(2y + u + w) \geq 3 \cdot 2 + 2 \cdot 3 - 6(x + \overline{x})} \quad \text{General resolution on } x$$

Generalized Resolution

In conflict-driven search, linear combination always made to cancel variable (on which constraints disagree)

Generalized resolution rule [Hoo88, Hoo92]

$$\frac{a_j x_j + \sum_{i \neq j} a_i \ell_i \geq A \qquad b_j \overline{x}_j + \sum_{i \neq j} b_i \ell_i \geq B}{\sum_{i \neq j} \left(\frac{c}{a_j} a_i + \frac{c}{b_j} b_i\right) \ell_i \geq \frac{c}{a_j} A + \frac{c}{b_j} B - c} \quad [c = \mathrm{lcm}(a_j, b_j)]$$

Another toy example:

$$\frac{2x + y + z \geq 2 \qquad 3\overline{x} + 2y + u + w \geq 3}{(3y + 3z) + (4y + 2u + 2w) \geq 12 - 6} \quad \text{General resolution on } x$$

Generalized Resolution

In conflict-driven search, linear combination always made to cancel variable (on which constraints disagree)

Generalized resolution rule [Hoo88, Hoo92]

$$\frac{a_j x_j + \sum_{i \neq j} a_i \ell_i \geq A \qquad b_j \overline{x}_j + \sum_{i \neq j} b_i \ell_i \geq B}{\sum_{i \neq j} \left(\frac{c}{a_j} a_i + \frac{c}{b_j} b_i\right) \ell_i \geq \frac{c}{a_j} A + \frac{c}{b_j} B - c} \quad [c = \mathrm{lcm}(a_j, b_j)]$$

Another toy example:

$$\frac{2x + y + z \geq 2 \qquad 3\overline{x} + 2y + u + w \geq 3}{7y + 3z + 2u + 2w \geq 6} \text{ General resolution on } x$$

Saturation

What's more, pseudo-Boolean solvers based on [CK05] do **not** do division

Instead use that no variable coefficient need be larger than maximum contribution required from that variable

Saturation rule

$$\frac{\sum_i a_i \ell_i \geq A}{\sum_i \min\{a_i, A\} \cdot \ell_i \geq A}$$

Saturation

What's more, pseudo-Boolean solvers based on [CK05] do **not** do division

Instead use that no variable coefficient need be larger than maximum contribution required from that variable

Saturation rule

$$\frac{\sum_i a_i \ell_i \geq A}{\sum_i \min\{a_i, A\} \cdot \ell_i \geq A}$$

Continuing our example:

$$\frac{7y + 3z + 2u + 2w \geq 6}{6y + 3z + 2u + 2w \geq 6}$$

Theoretical Understanding of Applied PB Reasoning?

Flavours of cutting planes in practice:

1. Boolean rule: (a) saturation or (b) division
2. Linear combinations: (a) generalized resolution or (b) no restrictions

Theoretical Understanding of Applied PB Reasoning?

Flavours of cutting planes in practice:

1. Boolean rule: (a) saturation or (b) division
2. Linear combinations: (a) generalized resolution or (b) no restrictions

Using generalized resolution seems inherent in conflict-driven setting
But what about Boolean rule?

- Saturation most popular (in [CK05, LP10], et cetera)
- Division used only recently in [EN18]
- What are the relative strengths of these rules? AFAIK no study!

Theoretical Understanding of Applied PB Reasoning?

Flavours of cutting planes in practice:

1. Boolean rule: (a) saturation or (b) division
2. Linear combinations: (a) generalized resolution or (b) no restrictions

Using generalized resolution seems inherent in conflict-driven setting
But what about Boolean rule?

- Saturation most popular (in [CK05, LP10], et cetera)
- Division used only recently in [EN18]
- What are the relative strengths of these rules? AFAIK no study!

Striking contrast to long line of work on resolution and CDCL ([BKS04, HBPV08, BHJ08, AFT11, PD11] . . .)

Our Results

1. For CNF, saturation no stronger than resolution proof system / CDCL (even for unrestricted linear combinations)

Our Results

1. For CNF, saturation no stronger than resolution proof system / CDCL (even for unrestricted linear combinations)
2. Division + generalized resolution can be exponentially stronger than saturation + unrestricted linear combinations

Our Results

1. For CNF, saturation no stronger than resolution proof system / CDCL (even for unrestricted linear combinations)
2. Division + generalized resolution can be exponentially stronger than saturation + unrestricted linear combinations
3. Single saturation step can require unbounded # divisions to simulate, even with unrestricted linear combinations

Our Results

1. For CNF, saturation no stronger than resolution proof system / CDCL (even for unrestricted linear combinations)
2. Division + generalized resolution can be exponentially stronger than saturation + unrestricted linear combinations
3. Single saturation step can require unbounded # divisions to simulate, even with unrestricted linear combinations

1st result strengthens [VEG+18]
Focus on 2nd and 3rd results — first of its kind (AFAIK)

Cutting Planes and Implicational Completeness

- All flavours of cutting planes except division + unrestricted linear combinations as in [CCT87] collapse to resolution for CNFs
- Full cutting planes implicationally complete — can recover, e.g., cardinality constraints from CNF

Cutting Planes and Implicational Completeness

- All flavours of cutting planes except division + unrestricted linear combinations as in [CCT87] collapse to resolution for CNFs
- Full cutting planes implicationally complete — can recover, e.g., cardinality constraints from CNF

$$
\begin{array}{rl}
x + y \phantom{{}+z} \geq 1 & \\
x + \phantom{y+{}} z \geq 1 & \\
y + z \geq 1 & \\
\hline
2x + 2y + 2z \geq 3 & \text{[2 non-cancelling additions]} \\
\hline
x + y + z \geq 2 & \text{[Divide by } 2\text{]}
\end{array}
$$

Cutting Planes and Implicational Completeness

- All flavours of cutting planes except division + unrestricted linear combinations as in [CCT87] collapse to resolution for CNFs
- Full cutting planes implicationally complete — can recover, e.g., cardinality constraints from CNF

$$\frac{\dfrac{\begin{array}{r} x + y \phantom{{}+z} \geq 1 \\ x + \phantom{y+{}} z \geq 1 \\ y + z \geq 1 \end{array}}{2x + 2y + 2z \geq 3} \text{ [2 non-cancelling additions]}}{x + y + z \geq 2} \text{ [Divide by } 2\text{]}$$

- Impossible with generalized resolution!
- So pigeonhole principle (PHP) in CNF hard for PB solvers
- CNFs make life hard for both saturation and division — but we want to show that division can be stronger!

Cutting Planes and Implicational Completeness

- All flavours of cutting planes except division + unrestricted linear combinations as in [CCT87] collapse to resolution for CNFs
- Full cutting planes implicationally complete — can recover, e.g., cardinality constraints from CNF

$$\frac{\dfrac{\begin{array}{r} x + y \geq 1 \\ x + z \geq 1 \\ y + z \geq 1 \end{array}}{2x + 2y + 2z \geq 3}\ \text{[2 non-cancelling additions]}}{x + y + z \geq 2}\ \text{[Divide by } 2\text{]}$$

- Impossible with generalized resolution!
- So pigeonhole principle (PHP) in CNF hard for PB solvers
- CNFs make life hard for both saturation and division — but we want to show that division can be stronger! *Can do so by cheating. . .*

Division + Resolution Can Be Stronger Than Saturation

Take CNF like PHP or subset cardinality [Spe10, VS10, MN14]
Exponentially hard for all flavours of cutting planes except [CCT87]

Division + Resolution Can Be Stronger Than Saturation

Take CNF like PHP or subset cardinality [Spe10, VS10, MN14]
Exponentially hard for all flavours of cutting planes except [CCT87]

$$\frac{\begin{array}{r} x + y \geq 1 \\ x + z \geq 1 \\ y + z \geq 1 \end{array}}{\dfrac{2x + 2y + 2z \geq 3}{x + y + z \geq 2}\ \text{[Divide by 2]}}\ \text{[2 non-cancelling additions]}$$

Division + Resolution Can Be Stronger Than Saturation

Take CNF like PHP or subset cardinality [Spe10, VS10, MN14]
Exponentially hard for all flavours of cutting planes except [CCT87]

$$\begin{array}{llll}
h_1 + h_2 + x + y & \geq 1 & & \\
\overline{h}_1 + x + z & \geq 2 & & \\
\overline{h}_2 + y + z & \geq 2 & & \\
\hline
2x + 2y + 2z & \geq 3 & & \text{[2 generalized resolution steps]} \\
\hline
x + y + z & \geq 2 & & \text{[Divide by 2]}
\end{array}$$

Add helper variables to make all linear combinations cancelling
⇒ Now easy for division, since easy for full cutting planes

Division + Resolution Can Be Stronger Than Saturation

Take CNF like PHP or subset cardinality [Spe10, VS10, MN14]
Exponentially hard for all flavours of cutting planes except [CCT87]

$$\frac{\dfrac{\begin{array}{rl} h_1 + h_2 + x + y & \geq 1 \\ \overline{h}_1 + x + z & \geq 2 \\ \overline{h}_2 + y + z & \geq 2 \end{array}}{2x + 2y + 2z \geq 3}\ \text{[2 generalized resolution steps]}}{x + y + z \geq 2}\ \text{[Divide by 2]}$$

Add helper variables to make all linear combinations cancelling
⇒ Now easy for division, since easy for full cutting planes

Assigning helper variables $= 0$ gives back CNF encoding
⇒ Cutting planes proofs preserved under partial assignments
⇒ Still hard for saturation, even with unrestricted linear combinations

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$200x + 51y + 50z + 49w \geq 100$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$200x + 51y + 50z + 49w \geq 100$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$\frac{200x + \quad 51y + \quad 50z + \quad 49w \geq \quad 100}{20000x + 5100y + 5000z + 4900w \geq 10000} \text{ Multiplication by } 100$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$\frac{\dfrac{200x + 51y + 50z + 49w \geq 100}{20000x + 5100y + 5000z + 4900w \geq 10000}\ \text{Multiplication by } 100}{199x + 51y + 50z + 49w \geq 100}\ \text{Division by } 101$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$\frac{\dfrac{\dfrac{200x + 51y + 50z + 49w \geq 100}{20000x + 5100y + 5000z + 4900w \geq 10000}\ \text{Multiplication by } 100}{199x + 51y + 50z + 49w \geq 100}\ \text{Division by } 101}{19900x + 5100y + 5000z + 4900w \geq 10000}\ \text{Multiplication by } 100$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$200x + 51y + 50z + 49w \geq 100$$

Multiplication by 100

$$20000x + 5100y + 5000z + 4900w \geq 10000$$

Division by 101

$$199x + 51y + 50z + 49w \geq 100$$

Multiplication by 100

$$19900x + 5100y + 5000z + 4900w \geq 10000$$

Division by 101

$$198x + 51y + 50z + 49w \geq 100$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$\dfrac{200x + 51y + 50z + 49w \geq 100}{20000x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\dfrac{20000x + 5100y + 5000z + 4900w \geq 10000}{199x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\dfrac{199x + 51y + 50z + 49w \geq 100}{19900x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\dfrac{19900x + 5100y + 5000z + 4900w \geq 10000}{198x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\dfrac{198x + 51y + 50z + 49w \geq 100}{19800x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$\frac{200x + 51y + 50z + 49w \geq 100}{20000x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\frac{20000x + 5100y + 5000z + 4900w \geq 10000}{199x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\frac{199x + 51y + 50z + 49w \geq 100}{19900x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\frac{19900x + 5100y + 5000z + 4900w \geq 10000}{198x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\frac{198x + 51y + 50z + 49w \geq 100}{19800x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\frac{19800x + 5100y + 5000z + 4900w \geq 10000}{197x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\vdots$$

Simulating Saturation by Division

Division can simulate saturation by completeness — but how efficiently?

$$\frac{200x + 51y + 50z + 49w \geq 100}{20000x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\frac{20000x + 5100y + 5000z + 4900w \geq 10000}{199x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\frac{199x + 51y + 50z + 49w \geq 100}{19900x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\frac{19900x + 5100y + 5000z + 4900w \geq 10000}{198x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\frac{198x + 51y + 50z + 49w \geq 100}{19800x + 5100y + 5000z + 4900w \geq 10000} \quad \text{Multiplication by } 100$$

$$\frac{19800x + 5100y + 5000z + 4900w \geq 10000}{197x + 51y + 50z + 49w \geq 100} \quad \text{Division by } 101$$

$$\vdots$$

Exponentially many steps measured in bitsize of coefficents

Impossible to get rid of exponential dependence in general!

Division Can't Simulate Saturation Efficiently

Consider saturation step

$$\frac{2Rx + \sum_{i=1}^{R} z_i \geq R}{Rx + \sum_{i=1}^{R} z_i \geq R}$$

Division Can't Simulate Saturation Efficiently

Consider saturation step

$$\frac{2Rx + \sum_{i=1}^{R} z_i \geq R}{Rx + \sum_{i=1}^{R} z_i \geq R}$$

Theorem

Deriving $Rx + \sum_{i=1}^{R} z_i \geq R$ from $2Rx + \sum_{i=1}^{R} z_i \geq R$ requires at least $\sqrt[4]{R}$ division steps for cutting planes with unrestricted linear combinations

Division Can't Simulate Saturation Efficiently

Consider saturation step

$$\frac{2Rx + \sum_{i=1}^{R} z_i \geq R}{Rx + \sum_{i=1}^{R} z_i \geq R}$$

Theorem

Deriving $Rx + \sum_{i=1}^{R} z_i \geq R$ from $2Rx + \sum_{i=1}^{R} z_i \geq R$ requires at least $\sqrt[4]{R}$ division steps for cutting planes with unrestricted linear combinations

Proof sketch:

- All derived lines are on form $L \doteq Ax + \sum_{i=1}^{R} b_i z_i \geq C$

Division Can't Simulate Saturation Efficiently

Consider saturation step

$$\frac{2Rx + \sum_{i=1}^{R} z_i \geq R}{Rx + \sum_{i=1}^{R} z_i \geq R}$$

Theorem

Deriving $Rx + \sum_{i=1}^{R} z_i \geq R$ from $2Rx + \sum_{i=1}^{R} z_i \geq R$ requires at least $\sqrt[4]{R}$ division steps for cutting planes with unrestricted linear combinations

Proof sketch:

- All derived lines are on form $L \doteq Ax + \sum_{i=1}^{R} b_i z_i \geq C$
- Define potential $\mathcal{P}(L) = C/\sqrt{A \sum_i b_i}$

Division Can't Simulate Saturation Efficiently

Consider saturation step

$$\frac{2Rx + \sum_{i=1}^{R} z_i \geq R}{Rx + \sum_{i=1}^{R} z_i \geq R}$$

Theorem

Deriving $Rx + \sum_{i=1}^{R} z_i \geq R$ from $2Rx + \sum_{i=1}^{R} z_i \geq R$ requires at least $\sqrt[4]{R}$ division steps for cutting planes with unrestricted linear combinations

Proof sketch:

- All derived lines are on form $L \doteq Ax + \sum_{i=1}^{R} b_i z_i \geq C$
- Define potential $\mathcal{P}(L) = C/\sqrt{A \sum_i b_i}$
- Start potential $1/\sqrt{2}$; end potential 1

Division Can't Simulate Saturation Efficiently

Consider saturation step

$$\frac{2Rx + \sum_{i=1}^{R} z_i \geq R}{Rx + \sum_{i=1}^{R} z_i \geq R}$$

Theorem

Deriving $Rx + \sum_{i=1}^{R} z_i \geq R$ *from* $2Rx + \sum_{i=1}^{R} z_i \geq R$ *requires at least* $\sqrt[4]{R}$ *division steps for cutting planes with unrestricted linear combinations*

Proof sketch:

- All derived lines are on form $L \doteq Ax + \sum_{i=1}^{R} b_i z_i \geq C$
- Define potential $\mathcal{P}(L) = C/\sqrt{A \sum_i b_i}$
- Start potential $1/\sqrt{2}$; end potential 1
- Linear combinations don't increase potential

Division Can't Simulate Saturation Efficiently

Consider saturation step

$$\frac{2Rx + \sum_{i=1}^{R} z_i \geq R}{Rx + \sum_{i=1}^{R} z_i \geq R}$$

Theorem

Deriving $Rx + \sum_{i=1}^{R} z_i \geq R$ from $2Rx + \sum_{i=1}^{R} z_i \geq R$ requires at least $\sqrt[4]{R}$ division steps for cutting planes with unrestricted linear combinations

Proof sketch:

- All derived lines are on form $L \doteq Ax + \sum_{i=1}^{R} b_i z_i \geq C$
- Define potential $\mathcal{P}(L) = C/\sqrt{A \sum_i b_i}$
- Start potential $1/\sqrt{2}$; end potential 1
- Linear combinations don't increase potential
- Potential increase from single division $< R^{-1/4}$

Saturation + Resolution Can Be Stronger Than Division?

- Does this show that saturation + generalized resolution can be exponentially stronger than division? **No!**

Saturation + Resolution Can Be Stronger Than Division?

- Does this show that saturation + generalized resolution can be exponentially stronger than division? **No!**
- Only shows that saturation step can't be simulated efficiently

Saturation + Resolution Can Be Stronger Than Division?

- Does this show that saturation + generalized resolution can be exponentially stronger than division? **No!**
- Only shows that saturation step can't be simulated efficiently
- Doesn't rule out that cutting planes with division could prove unsatisfiability of benchmarks in completely different way

Saturation + Resolution Can Be Stronger Than Division?

- Does this show that saturation + generalized resolution can be exponentially stronger than division? **No!**
- Only shows that saturation step can't be simulated efficiently
- Doesn't rule out that cutting planes with division could prove unsatisfiability of benchmarks in completely different way
- But if division is always as good as saturation, then it seems like proof of this can't be simple step-by-step simulation (as for most other such results)

Some Tentative Experimental Results

Caveat: Very much a work in progress...

Some Tentative Experimental Results

Caveat: Very much a work in progress...

Strength of Division

- When division better than saturation, *RoundingSat* [EN18] can run much faster than *Sat4j* [LP10]
- But sensitive to how helper variables encoded

Some Tentative Experimental Results

Caveat: Very much a work in progress...

Strength of Division

- When division better than saturation, *RoundingSat* [EN18] can run much faster than *Sat4j* [LP10]
- But sensitive to how helper variables encoded

Strength of Saturation

- Have easy benchmarks for saturation that look tricky for division — in theory
- In practice, the benchmarks we tried so far are hard for both division- and saturation-based solvers

Some Tentative Experimental Results

Caveat: Very much a work in progress. . .

Strength of Division

- When division better than saturation, *RoundingSat* [EN18] can run much faster than *Sat4j* [LP10]
- But sensitive to how helper variables encoded

Strength of Saturation

- Have easy benchmarks for saturation that look tricky for division — in theory
- In practice, the benchmarks we tried so far are hard for both division- and saturation-based solvers

Further caveat: obviously artificial benchmarks — we just want to see if separations can happen in actual solvers

Directions for Future Research

Division promising in practice

- Higher conflict speed when PB reasoning doesn't help [EN18]
- Seems to perform better when PB reasoning crucial [EGNV18]
- Keeps coefficients small — can do fixed-precision integer arithmetic

Directions for Future Research

Division promising in practice

- Higher conflict speed when PB reasoning doesn't help [EN18]
- Seems to perform better when PB reasoning crucial [EGNV18]
- Keeps coefficients small — can do fixed-precision integer arithmetic

Questions regarding saturation

- Theory: Can cutting planes with saturation be more powerful than division at proving unsatisfiability?
- Practice: Maybe best to combine division and saturation?

Directions for Future Research

Division promising in practice

- Higher conflict speed when PB reasoning doesn't help [EN18]
- Seems to perform better when PB reasoning crucial [EGNV18]
- Keeps coefficients small — can do fixed-precision integer arithmetic

Questions regarding saturation

- Theory: Can cutting planes with saturation be more powerful than division at proving unsatisfiability?
- Practice: Maybe best to combine division and saturation?

Fundamental challenges

- All PB solvers degenerate to resolution for CNF inputs
- Sometimes very poor performance even on rationally infeasible LPs! Combine with MIP techniques?

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm
- Possible to port to pseudo-Boolean reasoning
  (Didn't talk about how — see `tinyurl.com/ConflictDrivenPBS` for tutorial)

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm
- Possible to port to pseudo-Boolean reasoning
  (Didn't talk about how — see `tinyurl.com/ConflictDrivenPBS` for tutorial)
- Potential exponential gains mostly haven't materialized so far

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm
- Possible to port to pseudo-Boolean reasoning
  (Didn't talk about how — see `tinyurl.com/ConflictDrivenPBS` for tutorial)
- Potential exponential gains mostly haven't materialized so far
- Instead highly nontrivial challenges regarding
  - Efficient implementation
  - Theoretical understanding

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm
- Possible to port to pseudo-Boolean reasoning
  (Didn't talk about how — see `tinyurl.com/ConflictDrivenPBS` for tutorial)
- Potential exponential gains mostly haven't materialized so far
- Instead highly nontrivial challenges regarding
  - Efficient implementation
  - Theoretical understanding
- **This work:** Study of relative strengths of division and saturation rules — seem incomparable

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm
- Possible to port to pseudo-Boolean reasoning
  (Didn't talk about how — see `tinyurl.com/ConflictDrivenPBS` for tutorial)
- Potential exponential gains mostly haven't materialized so far
- Instead highly nontrivial challenges regarding
  - Efficient implementation
  - Theoretical understanding
- **This work:** Study of relative strengths of division and saturation rules — seem incomparable
- **Future research:** Build blisteringly fast pseudo-Boolean solvers (far from there yet, but remember CDCL took 50 years)

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm
- Possible to port to pseudo-Boolean reasoning
  (Didn't talk about how — see `tinyurl.com/ConflictDrivenPBS` for tutorial)
- Potential exponential gains mostly haven't materialized so far
- Instead highly nontrivial challenges regarding
  - Efficient implementation
  - Theoretical understanding
- **This work:** Study of relative strengths of division and saturation rules — seem incomparable
- **Future research:** Build blisteringly fast pseudo-Boolean solvers (far from there yet, but remember CDCL took 50 years)
- Along the way lots of fun questions to work on! ☺

Take-Home Message

- Conflict-driven search hugely successful SAT solving paradigm
- Possible to port to pseudo-Boolean reasoning
  (Didn't talk about how — see `tinyurl.com/ConflictDrivenPBS` for tutorial)
- Potential exponential gains mostly haven't materialized so far
- Instead highly nontrivial challenges regarding
  - Efficient implementation
  - Theoretical understanding
- **This work:** Study of relative strengths of division and saturation rules — seem incomparable
- **Future research:** Build blisteringly fast pseudo-Boolean solvers (far from there yet, but remember CDCL took 50 years)
- Along the way lots of fun questions to work on! ☺

Thank you for your attention!

References I

[AFT11] Albert Atserias, Johannes Klaus Fichte, and Marc Thurley. Clause-learning algorithms with many restarts and bounded-width resolution. *Journal of Artificial Intelligence Research*, 40:353–373, January 2011. Preliminary version in *SAT '09*.

[BHJ08] Samuel R. Buss, Jan Hoffmann, and Jan Johannsen. Resolution trees with lemmas: Resolution refinements that characterize DLL-algorithms with clause learning. *Logical Methods in Computer Science*, 4(4:13), December 2008.

[BKS04] Paul Beame, Henry Kautz, and Ashish Sabharwal. Towards understanding and harnessing the potential of clause learning. *Journal of Artificial Intelligence Research*, 22:319–351, December 2004. Preliminary version in *IJCAI '03*.

[CCT87] William Cook, Collette Rene Coullard, and György Turán. On the complexity of cutting-plane proofs. *Discrete Applied Mathematics*, 18(1):25–38, November 1987.

[CK05] Donald Chai and Andreas Kuehlmann. A fast pseudo-Boolean constraint solver. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 24(3):305–317, March 2005. Preliminary version in *DAC '03*.

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on Theory of Computing (STOC '71)*, pages 151–158, 1971.

References II

[DG02] Heidi E. Dixon and Matthew L. Ginsberg. Inference methods for a pseudo-Boolean satisfiability solver. In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI '02)*, pages 635–640, July 2002.

[EGNV18] Jan Elffers, Jesús Giráldez-Cru, Jakob Nordström, and Marc Vinyals. Using combinatorial benchmarks to probe the reasoning power of pseudo-Boolean solvers. In *Proceedings of the 21st International Conference on Theory and Applications of Satisfiability Testing (SAT '18)*, volume 10929 of *Lecture Notes in Computer Science*, pages 75–93. Springer, July 2018.

[EN18] Jan Elffers and Jakob Nordström. Divide and conquer: Towards faster pseudo-Boolean solving. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI '18)*, pages 1291–1299, July 2018.

[ES06] Niklas Eén and Niklas Sörensson. Translating pseudo-Boolean constraints into SAT. *Journal on Satisfiability, Boolean Modeling and Computation*, 2(1-4):1–26, 2006.

[Hak85] Armin Haken. The intractability of resolution. *Theoretical Computer Science*, 39(2-3):297–308, August 1985.

References III

[HBPV08] Philipp Hertel, Fahiem Bacchus, Toniann Pitassi, and Allen Van Gelder. Clause learning can effectively P-simulate general propositional resolution. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI '08)*, pages 283–290, July 2008.

[Hoo88] John N. Hooker. Generalized resolution and cutting planes. *Annals of Operations Research*, 12(1):217–239, 1988.

[Hoo92] John N. Hooker. Generalized resolution for 0-1 linear inequalities. *Annals of Mathematics and Artificial Intelligence*, 6(1):271–286, 1992.

[Lev73] Leonid A. Levin. Universal sequential search problems. *Problemy peredachi informatsii*, 9(3):115–116, 1973. In Russian. Available at http://mi.mathnet.ru/ppi914.

[LP10] Daniel Le Berre and Anne Parrain. The Sat4j library, release 2.2. *Journal on Satisfiability, Boolean Modeling and Computation*, 7:59–64, 2010.

[MML14] Ruben Martins, Vasco M. Manquinho, and Inês Lynce. Open-WBO: A modular MaxSAT solver. In *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *Lecture Notes in Computer Science*, pages 438–445. Springer, July 2014.

References IV

[MN14] Mladen Mikša and Jakob Nordström. Long proofs of (seemingly) simple formulas. In *Proceedings of the 17th International Conference on Theory and Applications of Satisfiability Testing (SAT '14)*, volume 8561 of *Lecture Notes in Computer Science*, pages 121–137. Springer, July 2014.

[PD11] Knot Pipatsrisawat and Adnan Darwiche. On the power of clause-learning SAT solvers as resolution engines. *Artificial Intelligence*, 175(2):512–525, February 2011. Preliminary version in *CP '09*.

[SN15] Masahiko Sakai and Hidetomo Nabeshima. Construction of an ROBDD for a PB-constraint in band form and related techniques for PB-solvers. *IEICE Transactions on Information and Systems*, 98-D(6):1121–1127, June 2015.

[Spe10] Ivor Spence. sgen1: A generator of small but difficult satisfiability benchmarks. *Journal of Experimental Algorithmics*, 15:1.2:1–1.2:15, March 2010.

[SS06] Hossein M. Sheini and Karem A. Sakallah. Pueblo: A hybrid pseudo-Boolean SAT solver. *Journal on Satisfiability, Boolean Modeling and Computation*, 2(1-4):165–189, March 2006. Preliminary version in *DATE '05*.

References V

[VEG+18] Marc Vinyals, Jan Elffers, Jesús Giráldez-Cru, Stephan Gocht, and Jakob Nordström. In between resolution and cutting planes: A study of proof systems for pseudo-Boolean SAT solving. In *Proceedings of the 21st International Conference on Theory and Applications of Satisfiability Testing (SAT '18)*, volume 10929 of *Lecture Notes in Computer Science*, pages 292–310. Springer, July 2018.

[VS10] Allen Van Gelder and Ivor Spence. Zero-one designs produce small hard SAT instances. In *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT '10)*, volume 6175 of *Lecture Notes in Computer Science*, pages 388–397. Springer, July 2010.