On Division Versus Saturation in Pseudo-Boolean Solving

Stephan Gocht, Jakob Nordström, Amir Yehudayoff

21.05.2019
The SAT Problem

- find assignment to Boolean variables that satisfies constraints
- SAT expressive formalism
  - captures many real-world problems
- theoretically hard, in practice often feasible
- state-of-the-art:
  - conflict driven clause learning (CDCL) SAT solvers
    - [MS96, BS97, MMZ + 01, ...]
The SAT Problem

- find assignment to Boolean variables that satisfies constraints
- SAT expressive formalism
  \( \Rightarrow \) captures many real-world problems
- theoretically hard, in practice often feasible
- state-of-the-art:
  conflict driven clause learning (CDCL) SAT solvers
  \([\text{MS96, BS97, MMZ}^+\text{01}, \ldots]\)

Potential for improvement?

- CDCL SAT solvers essentially based on resolution
- resolution very simple
  + simple data structures allow efficient implementation
  - weak method of reasoning
Pseudo-Boolean SAT Solving

- **exponential stronger reasoning** via cutting planes proof system
- constraints: pseudo-Boolean (PB) (0-1 linear inequalities)
Pseudo-Boolean SAT Solving

▶ exponential stronger reasoning via cutting planes proof system
▶ constraints: pseudo-Boolean (PB) (0-1 linear inequalities)

In practice:
▶ PB solvers often worse than resolution based solvers
▶ only implementation details?
Pseudo-Boolean SAT Solving

- exponential stronger reasoning via cutting planes proof system
- constraints: pseudo-Boolean (PB) (0-1 linear inequalities)

In practice:
- PB solvers often worse than resolution based solvers
- only implementation details? no
- different solver use different variants of cutting planes
- none as strong as full cutting planes

⇒ study cutting-planes subsystems used in PB solvers
Method of Reasoning Underlying CDCL: Resolution

**Literal** $a$: a variable $x$ or its negation $\bar{x}$

**Clause** $C = a_1 \lor \cdots \lor a_k$: disjunction ($\lor$) of variables

**CNF** $F = C_1 \land \ldots \land C_m$: conjunction ($\land$) of clauses
Method of Reasoning Underlying CDCL: Resolution

**Literal** $a$: a variable $x$ or its negation $\bar{x}$

**Clause** $C = a_1 \lor \cdots \lor a_k$: disjunction ($\lor$) of variables

**CNF** $F = C_1 \land \ldots \land C_m$: conjunction ($\land$) of clauses

Goal: Show $F$ unsatisfiable using:

Resolution rule: $\frac{C \lor x}{C \lor D}$

Example:

$$\frac{y \lor x \quad \bar{x} \lor y}{y \quad \bar{y}}$$
Method of Reasoning Underlying CDCL: Resolution

Literal $a$: a variable $x$ or its negation $\overline{x}$
Clause $C = a_1 \lor \cdots \lor a_k$: disjunction ($\lor$) of variables
CNF $F = C_1 \land \ldots \land C_m$: conjunction ($\land$) of clauses

Goal: Show $F$ unsatisfiable using:

Resolution rule

$$
\begin{array}{c}
C \lor x \\
\overline{x} \lor D \\
\hline
C \lor D
\end{array}
$$

Example:

$$
\begin{array}{c}
y \lor x \\
\overline{x} \lor y \\
\hline
y
\end{array}
\Rightarrow \overline{y}
$$

- implicationally complete, i.e. can drive all consequences
- in particular, can refute all unsatisfiable CNF formulas
Pseudo-Boolean Constraints

- integer linear inequalities over 0-1 variables
- $1 = \text{true}$, $0 = \text{false}$

Example: $3\bar{x} + 2\bar{y} + 2z \geq 5$

Normalized Form

- use literals, i.e. $x, \bar{x}$ with $\bar{x} = (1 - x) \Rightarrow x + \bar{x} = 1$
- only positive coefficients and “$\geq$”
- degree (of falsity): right hand side of normalized constraint
Pseudo-Boolean Constraints

- integer linear inequalities over 0-1 variables
- $1 = \text{true}$, $0 = \text{false}$

Example: $3\overline{x} + 2\overline{y} + 2z \geq 5$

Normalized Form

- use literals, i.e. $x, \overline{x}$ with $\overline{x} = (1 - x) \Rightarrow x + \overline{x} = 1$
- only positive coefficients and "$\geq$"
- degree (of falsity): right hand side of normalized constraint

Representing Clauses:

$$\overline{x} \lor y \lor z \Rightarrow x + y + z \geq 1$$

Cardinality constraints, e.g. at-least 2:

$$(x \lor y) \land (x \lor z) \land (y \lor z) \Rightarrow x + y + z \geq 2$$
Cutting Planes: Linear Combination

Literal Axioms

\[ x \geq 0 \quad \bar{x} \geq 0 \]
Cutting Planes: Linear Combination

Literal Axioms

\[
\begin{align*}
    x & \geq 0 \\
    \bar{x} & \geq 0
\end{align*}
\]

Positive Linear Combination — Remember: \( x + \bar{x} = 1 \)

\[
\begin{align*}
    3 \cdot (y + z + \bar{x} \geq 1) & \quad 1 \cdot (2x + z \geq 3) \\
    3y + 4z + 2x + 3\bar{x} & \geq 6
\end{align*}
\]

\[
3y + 4z + 2x + 3\bar{x} \geq 6
\]

= \(2 + \bar{x}\)
Cutting Planes: Linear Combination

Literal Axioms

\[ x \geq 0 \quad \bar{x} \geq 0 \]

Positive Linear Combination — Remember: \( x + \bar{x} = 1 \)

\[
\frac{3 \cdot (y + z + \bar{x} \geq 1)}{3y + 4z +} \quad \frac{1 \cdot (2x + z \geq 3)}{\bar{x} \geq 4}
\]
Cutting Planes: Linear Combination

Literal Axioms

\[ x \geq 0 \quad \overline{x} \geq 0 \]

Positive Linear Combination — Remember: \( x + \overline{x} = 1 \)

\[
\begin{align*}
3 \cdot (y + z + \overline{x} \geq 1) & \quad 1 \cdot (2x + z \geq 3) \\
\underline{3y + 4z +} & \quad \underline{\overline{x} \geq 4}
\end{align*}
\]

Generalized Resolution

\[
\begin{align*}
2\overline{x} + v + \overline{w} \geq 2 & \quad 2x + y + z \geq 2 \\
\underline{v + \overline{w} + y + z} & \geq 2
\end{align*}
\]
Division (divide and round up)

\[
\frac{x + 2y + 2z \geq 3}{x + y + z \geq 2}
\]

Divide by 2
Cutting Planes: Boolean Rule

**Division** (divide and round up)

\[
\frac{x + 2y + 2z \geq 3}{x + y + z \geq 2}
\]

Divide by 2

**Saturation** (set coefficient to value \(\leq\) degree of falsity)

\[
\frac{4x + 4\overline{y} + z \geq 2}{2x + 2\overline{y} + z \geq 2}
\]
Example

\[
\begin{align*}
\bar{x}_1 + \bar{x}_2 & \geq 1 \quad \bar{x}_1 + \bar{x}_3 \geq 1 \\
\bar{x}_2 + \bar{x}_3 & \geq 1 \\
x_1 + x_2 + x_3 & \geq 2
\end{align*}
\]
Example

\[
\begin{align*}
\bar{x}_1 + \bar{x}_2 & \geq 1 & \bar{x}_1 + \bar{x}_3 & \geq 1 \\
2\bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 & \bar{x}_2 + \bar{x}_3 & \geq 1 \\
\hline
\hline
\bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 & x_1 + x_2 + x_3 & \geq 2
\end{align*}
\]
Example

\[
\begin{align*}
\bar{x}_1 + \bar{x}_2 & \geq 1 & \bar{x}_1 + \bar{x}_3 & \geq 1 \\
2\bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 & \bar{x}_2 + \bar{x}_3 & \geq 1 \\
2\bar{x}_1 + 2\bar{x}_2 + 2\bar{x}_3 & \geq 3 \\
\hline
x_1 + x_2 + x_3 & \geq 2
\end{align*}
\]
Example

\[
\begin{align*}
\bar{x}_1 + \bar{x}_2 & \geq 1 & \bar{x}_1 + \bar{x}_3 & \geq 1 \\
2\bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 & \bar{x}_2 + \bar{x}_3 & \geq 1 \\
2\bar{x}_1 + 2\bar{x}_2 + 2\bar{x}_3 & \geq 3 \\
\bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 & x_1 + x_2 + x_3 & \geq 2
\end{align*}
\]
Example

\[
\begin{align*}
\bar{x}_1 + \bar{x}_2 & \geq 1 & \bar{x}_1 + \bar{x}_3 & \geq 1 \\
2\bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 & \bar{x}_2 + \bar{x}_3 & \geq 1 \\
2\bar{x}_1 + 2\bar{x}_2 + 2\bar{x}_3 & \geq 3 & \bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 \\
\bar{x}_1 + \bar{x}_2 + \bar{x}_3 & \geq 2 & 0 & \geq 1 \\
x_1 + x_2 + x_3 & \geq 2
\end{align*}
\]
Pseudo-Boolean Solvers and the Subsystem Used

not all rules used by implementations

generalized resolution and saturation:
  ▶ PRS [DG02]
  ▶ Galena [CK05]
  ▶ Pueblo [SS06]
  ▶ Sat4j [LP10]

generalized resolution and division:
  ▶ RoundingSat [EN18]

linear combination and division (full cutting planes):
  ▶ ∅
Systematic Overview [VEG⁺18]

\[ \mathcal{A} \rightarrow \mathcal{B} \quad \mathcal{B} \text{ can do everything } \mathcal{A} \text{ can} \]

\[ \mathcal{A} \rightarrow \quad \mathcal{B} \text{ can do everything } \mathcal{A} \text{ can and } \mathcal{B} \text{ can do things } \mathcal{A} \text{ can’t} \]

† polynomial-sized coefficients
Our Results

We want to study reasoning power depending on choice of:

- boolean rule: (a) division, (b) saturation
- linear combination: (a) generalized resolution, (b) unrestricted

- saturation as boolean rule $\Rightarrow$ generalized resolution as powerful as unrestricted linear combinations
- division + generalized resolution can be exponentially stronger than saturation + unrestricted linear combinations
- replacing single saturation, requires large # divisions
Our Result: Strength of Generalized Resolution

\[ \mathcal{A} \rightarrow \mathcal{B} \text{ } \mathcal{B} \text{ can do everything } \mathcal{A} \text{ can} \]

\[ \mathcal{A} \longrightarrow \mathcal{B} \text{ } \mathcal{B} \text{ can do everything } \mathcal{A} \text{ can and } \mathcal{B} \text{ can do things } \mathcal{A} \text{ can’t} \]

† polynomial-sized coefficients
Proof Sketch: “Rewriting” to Generalized Resolution

\[
\begin{array}{c}
\bar{x} + 2y \geq 2 \\
2x + 2y + z \geq 2 \\
x + 4y + z \geq 3
\end{array}
\]

- rewrite as generalized resolution:
Proof Sketch: “Rewriting” to Generalized Resolution

\[
\begin{align*}
\bar{x} + 2y & \geq 2 \\
2x + 2y + z & \geq 2 \\
\hline
x + 4y + z & \geq 3
\end{align*}
\]

▸ rewrite as generalized resolution:

\[
\begin{align*}
\text{generalized res.} & \quad 2 \cdot (\bar{x} + 2y \geq 2) & 2x + 2y + z \geq 2 \\
& \quad 6y + z \geq 4
\end{align*}
\]
Proof Sketch: “Rewriting” to Generalized Resolution

\[
\begin{align*}
\bar{x} + 2y &\geq 2 \\
2x + 2y + z &\geq 2 \\
x + 4y + z &\geq 3
\end{align*}
\]

- rewrite as generalized resolution:

\[
\begin{align*}
2 \cdot (\bar{x} + 2y \geq 2) &\quad \quad 2x + 2y + z \geq 2 \\
6y + z &\geq 4 &\quad \quad 2x + 2y + z \geq 2
\end{align*}
\]

postponed step

\[
\begin{align*}
2x + 8y + 2z &\geq 6
\end{align*}
\]

- postpone addition of constraints that can’t be rewritten
Proof Sketch: “Rewriting” to Generalized Resolution

\[
\begin{align*}
\bar{x} + 2y & \geq 2 \\
2x + 2y + z & \geq 2 \\
x + 3y + z & \geq 3
\end{align*}
\]

- rewrite as generalized resolution:

\[
\begin{align*}
\text{generalized res.} & \quad 2 \cdot (\bar{x} + 2y \geq 2) \\
& \quad 2x + 2y + z \geq 2 \\
\text{postponed step} & \quad 4y + z \geq 4 \\
& \quad 2x + 2y + z \geq 2 \\
& \quad 2x + 6y + 2z \geq 6
\end{align*}
\]

- postpone addition of constraints that can’t be rewritten
- saturation not affected by postponing
Our Result: Strength of Generalized Resolution

\[ \mathcal{A} \rightarrow \mathcal{B} \quad \mathcal{B} \text{ can do everything } \mathcal{A} \text{ can} \]

\[ \mathcal{A} \rightarrow \rightarrow \mathcal{B} \quad \mathcal{B} \text{ can do everything } \mathcal{A} \text{ can and } \mathcal{B} \text{ can do things } \mathcal{A} \text{ can't} \]

† polynomial-sized coefficients
Our Result: Strength of Division

\[ A \rightarrow B \quad B \text{ can do everything } A \text{ can} \]
\[ A \quad \quad B \text{ can do everything } A \text{ can and } B \text{ can do things } A \text{ can't} \]
\[ \uparrow \quad \text{polynomial-sized coefficients} \quad \neg \]

Stephan Gocht Division vs. Saturation 15/23
Strength of Division

- cutting planes stronger than resolution because it can “count”
- requires division + unrestricted linear combination
- for example, recovering cardinality constraints:

\[
\begin{align*}
\bar{x} + \bar{y} &\geq 1 \\
\bar{y} + \bar{z} &\geq 1 \\
\bar{x} + \bar{z} &\geq 1 \\
\hline
2\bar{x} + 2\bar{y} + 2\bar{z} &\geq 3 \\
\bar{x} + \bar{y} + \bar{z} &\geq 2
\end{align*}
\]
Strength of Division

- cutting planes stronger than resolution because it can “count”
- requires division + unrestricted linear combination
- for example, recovering cardinality constraints:

\[
\begin{align*}
    h_1 + h_2 &+ \bar{x} + \bar{y} \geq 1 \\
    \bar{h}_1 &+ \bar{y} + \bar{z} \geq 2 \\
    \bar{h}_2 &+ \bar{x} + \bar{z} \geq 2 \\
    \hline
    2\bar{x} + 2\bar{y} + 2\bar{z} &\geq 3 \\
    \hline
    \bar{x} + \bar{y} + \bar{z} &\geq 2
\end{align*}
\]

Achieve separation by modifying benchmark:

- introduce **helper variables** to allow generalized resolution
- still hard for saturation, easy for division + generalized res.
Practical Result for Division-Friendly Formulas

- apply “trick” to subset cardinality formulas

[Spe10, VS10, MN14]

Division–Friendly Formulas

![Graph showing CPU time in s (par1) vs. Scaling parameter with different markers for clausal, native, division, and saturation categories.]

Legend: clausal, native, division, saturation
Our Result: Strength of Division

\[ \mathcal{A} \rightarrow \mathcal{B} \quad \text{\(\mathcal{B}\) can do everything \(\mathcal{A}\) can} \]

\[ \mathcal{A} \rightarrow \mathcal{B} \quad \text{\(\mathcal{B}\) can do everything \(\mathcal{A}\) can and \(\mathcal{B}\) can do things \(\mathcal{A}\) can't} \]

† polynomial-sized coefficients  \[\rightarrow \neg \quad \text{negation} \]
Our Result: Strength of Saturation

\[ A \rightarrow B \quad \text{B can do everything A can} \]
\[ A \rightarrow B \quad \text{B can do everything A can and B can do things A can't} \]
\[ \uparrow \quad \text{polynomial-sized coefficients} \]
\[ \neg \quad \text{negation} \]
Simulating Saturation with Division, Lower Bound

To replace one saturation step

\[
2Rx + \sum_{i=1}^{2R} z_i \geq R \\
Rx + \sum_{i=1}^{2R} z_i \geq R
\]

by division…

\[
\frac{2Rx + \sum_{i=1}^{2R} z_i}{Rx + \sum_{i=1}^{2R} z_i} \geq R
\]
Simulating Saturation with Division, Lower Bound

To replace one saturation step

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

by division...

- it takes \( \Omega(\sqrt{R}) \) division steps
- still true if we add generalized resolution step to obtain unsaturated constraint, i.e., start from

\[
Rx + Ry + \sum_{i=1}^{R} z_i \geq R \quad \quad Rx + R\bar{y} + \sum_{i=R+1}^{2R} z_i \geq R
\]

- does not show that cutting planes with saturation can be exponentially stronger than cutting planes with division
define potential function $\mathcal{P}(C)$ such that:

- needs to change:
  $$\mathcal{P}(C_{\text{start}}) - \mathcal{P}(C_{\text{end}}) \geq 1/6$$

- doesn’t change with linear combination:
  $$\mathcal{P}(C_1 + C_2) \geq \min\{\mathcal{P}(C_1), \mathcal{P}(C_2)\}$$

- changes by a small amount by division:
  $$\mathcal{P}(C/k) \geq \mathcal{P}(C) - 1/\sqrt{R}$$
Proof Sketch

define potential function $\mathcal{P}(C)$ such that:

- needs to change:
  $\mathcal{P}(C_{\text{start}}) - \mathcal{P}(C_{\text{end}}) \geq 1/6$

- doesn’t change with linear combination:
  $\mathcal{P}(C_1 + C_2) \geq \min\{\mathcal{P}(C_1), \mathcal{P}(C_2)\}$

- changes by a small amount by division:
  $\mathcal{P}(C/k) \geq \mathcal{P}(C) - 1/\sqrt{R}$

$$\mathcal{P}(a x + \sum b_i z_i \geq A) := \ln \left(\frac{a}{A}\right)$$
Conclusion

\[ A \rightarrow^\ast B \quad B \text{ can do everything } A \text{ can} \]
\[ A \rightarrow B \quad B \text{ can do everything } A \text{ can and } B \text{ can do things } A \text{ can't} \]

\[ \dagger \text{ polynomial-sized coefficients} \quad \neg \quad \negation \]
Conclusion

A \rightarrow B \quad B \text{ can do everything } A \text{ can}

A \rightarrow B \quad B \text{ can do everything } A \text{ can and } B \text{ can do things } A \text{ can’t}

† \quad \text{polynomial-sized coefficients}

negation

Future Research Directions

▶ Can saturation be stronger than division in proving UNSAT?
▶ implement adaptive choice between division and saturation
▶ to supersede resolution on CNF we need “natural way” / heuristic for unrestricted linear combination + divisions
Future Research Directions

- Can saturation be stronger than division in proving UNSAT?
- Implement adaptive choice between division and saturation
- To supersede resolution on CNF we need “natural way” / heuristic for unrestricted linear combination + divisions

Thank you for your attention!
Simulating Saturation with Division

▶ saturation can be simulated by repeated division

\[
\begin{align*}
49x + 50y + 51z + 200y & \geq 100 \\
49x + 50y + 51z + 199y & \geq 100 \\
49x + 50y + 51z + 198y & \geq 100 \\
49x + 50y + 51z + 197y & \geq 100 \\
\ldots
\end{align*}
\]

▶ only guaranteed to reduce coefficient by 1 per iteration
⇒ only efficient if coefficients small
References I


References II


References III

[SS06] Hossein M. Sheini and Karem A. Sakallah.
Pueblo: A hybrid pseudo-Boolean SAT solver.
*Journal on Satisfiability, Boolean Modeling and Computation*,
Preliminary version in *DATE ’05*.

[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)*,

Zero-one designs produce small hard SAT instances.
In *Proceedings of the 13th International Conference on Theory and Applications of Satisfiability Testing (SAT ’10)*,